Université de Montréal

Identifying Latent Structures in Data

par

Sébastien Lachapelle

Département d'informatique et de recherche opérationnelle
Faculté des arts et des sciences

Thèse présentée en vue de l'obtention du grade de
Philosophiæ Doctor (Ph.D.)
en Informatique

April 12, 2024

Université de Montréal

Faculté des arts et des sciences

Cette thèse intitulée

Identifying Latent Structures in Data

présentée par

Sébastien Lachapelle

a été évaluée par un jury composé des personnes suivantes :

Yoshua Bengio
(président-rapporteur)

Simon Lacoste-Julien
(directeur de recherche)

Dhanya Sridhar
(membre du jury)

Aapo Hyvärinen
(examinateur externe)

Vincent Larivière
(représentant du doyen de la FESP)

À mes parents, Denis et Linda.

Résumé

Le triomphe de l'apprentissage profond dans divers domaines tels que la classification d'images, la reconnaissance vocale, la génération de langage naturel et la génération d'images a été rendu possible par l'augmentation de la taille des ensembles de données, l'augmentation de la capacité de calcul, une communauté open source dynamique et des innovations architecturales qui, ensemble, ont permis d'entrainer des réseaux neuronaux de plus en plus expressifs. Bien que cette nouvelle approche ait abouti à des percées impressionnantes, elle a été accompagnée d'un manque d'interprétabilité des modèles et de garanties théoriques. Cette thèse tente de construire des modèles suffisamment restreints pour être interprétables et/ou analysables théoriquement tout en restant suffisamment expressifs pour être utiles dans des modalités difficiles telles que les images. La plupart des contributions se concentrent sur l'identifiabilité, la propriété qu'un modèle statistique possède lorsque ses paramètres sont déterminés par la distribution qu'ils représentent, à une classe d'équivalence près. Bien que l'identifiabilité soit centrale en inférence causale, en apprentissage de graphe causal et en analyse de composantes indépendante, cette propriété n'est pas aussi bien comprise dans le contexte de l'apprentissage profond. Cette thèse soutient que l'étude de l'identifiabilité en apprentissage automatique est utile pour mieux comprendre les modèles existants ainsi que pour en construire de nouveaux qui soient interprétables et pourvus de garanties de généralisation. Ce qui en découle sont de nouvelles garanties d'identifiabilité pour des modèles expressifs, pour l'apprentissage de graphe causal et de représentations.

Les première et deuxième contributions (Chapitres 3 et 4) proposent de nouveaux algorithmes basés sur les gradients pour apprendre un graphe causal à partir de données observationnelles et interventionnelles, respectivement. Ces contributions ont étendu des approches contraintes continues des relations linéaires aux relations non linéaires et ont souligné l'avantage computationnel de ces approches lorsque l'ensemble de données est très grand.

Les troisième, quatrième et cinquième contributions (Chapitres 5, 6 et 7) fournissent de nouvelles garanties d'identifiabilité pour le désentrelacement (disentanglement) dans l'apprentissage de représentations. Le Chapitre 5 montre que, dans un modèle spécifique à variables latentes, les facteurs latents réels peuvent être identifiés à une permutation et une bijection par élément près lorsque des variables auxiliaires observées et/ou des facteurs latents passés les affectent de manière

parcimonieuse (sparse). Ces résultats ne font pas d'hypothèses paramétriques et caractérisent la structure du désentrelacement en fonction du graphe causal latent sous-jacent. Le Chapitre 6 introduit un problème d'optimisation bi-niveau pour l'apprentissage multi-tâches parcimonieux et prouve que, avec des tâches suffisamment parcimonieuses et diverses, la représentation apprise doit être désentrelacée. De plus, il fournit un argument formel montrant comment le désentrelacement est bénéfique dans un contexte d'apprentissage avec peu d'exemples (few-shot learning). Le Chapitre 7 étudie une classe simple de décodeurs que nous appelons "décodeurs additifs" pour lesquels nous pouvons prouver à la fois des garanties de désentrelacement et d'extrapolation. Les décodeurs additifs sont intéressants à étudier car ils ressemblent aux architectures utilisées dans l'apprentissage de représentations centrées sur les objets (object-centric representation learning) et constituent une étape vers la compréhension de la créativité et de l'extrapolation dans les modèles génératifs modernes.

Le Chapitre 8 discute de trois interprétations de l'identifiabilité et unifie les contributions de cette thèse à l'aide d'un cadre simple en trois étapes mettant en évidence le rôle de l'identifiabilité pour obtenir des garanties de généralisations. Spécifiquement, quatres types de problème sont couverts: l'apprentissage de graphes causals, les décodeurs additifs pour l'extrapolation, l'apprentissage multi-tâches parcimonieux et l'apprentissage semi-supervisé par regroupement (clustering). Les relations entre ces problèmes sont rendues transparentes grâce au cadre de la théorie de la décision statistique.

Mots clés: Identifiabilité, apprentissage de graphes causals, analyse de composantes indépendentes non linéaire, apprentissage de représentations causales, apprentissage de représentations identifiable, extrapolation, généralisation compositionelle, apprentissage représentations centrées sur les objets

Abstract

The triumph of deep learning in diverse settings such as image classification, speech recognition, natural language generation and image generation was driven mainly by increasingly large datasets, cheap compute, architectural innovations and a vibrant open-source community which together enabled training increasingly expressive neural networks. While this new approach yielded stunning breakthroughs, it came at the cost of model interpretability and theoretical guarantees. This thesis is an attempt at building models that are restricted enough to be interpretable and analyzed theoretically while remaining sufficiently expressive to be useful in high-dimensional data modalities. The focus of most contributions is on *identifiability*, the property a statistical model has when its parameters can be recovered from the distribution it entails, up to some equivalence class. While identifiability is central to causal inference, causal discovery and independent component analysis, its understanding in the context of deep learning is underdeveloped. This thesis argues that studying identifiability in deep learning and machine learning more broadly is useful to gain insights into existing models as well as to build new ones that are interpretable and amenable to generalization guarantees. What comes out are novel identifiability guarantees for expressive models, for both causal discovery and representation learning.

The first and second contributions (Chapters 3 & 4) propose novel gradient-based algorithms to learn a causal graph from observational and interventional data, respectively. These contributions extended continuous constrained approaches from linear to nonlinear relationships and highlighted the computational advantage of gradient-based approaches for large datasets.

The third, fourth and fifth contributions (Chapters 5, 6 & 7) provide novel identifiability guarantees for disentanglement in representation learning. Chapter 5 shows that, in a specific deep latent variable model, the ground-truth latent factors can be identified up to a permutation and an element-wise bijection when an observed auxiliary variable and/or past latent factors sparsely affect them. The result does not make parametric assumptions and characterizes the entanglement structure as a function of the ground-truth latent causal graph. Chapter 6 introduces a bilevel optimization problem to perform sparse multi-task learning and proves that, given sufficiently sparse and diverse tasks, the learned representation must be disentangled. Furthermore, it provides a formal argument for why disentanglement is beneficial in a few-shot learning setting. Chapter 7 studies a

simple class of decoders we call "additive decoders" for which we can prove both disentanglement and extrapolation guarantees. Additive decoders are interesting to study since they resemble architectures used in object-centric representation learning and form a step toward understanding creativity and extrapolation in modern generative models.

Chapter 8 discusses three interpretations of identifiability and unifies the contributions of this thesis under a simple three-steps framework highlighting the role of identifiability to obtain generalization guarantees. Specifically, four problem settings are covered: causal discovery, additive decoders for extrapolation, sparse multi-task learning and semi-supervised learning via clustering. The connections between all settings are made more transparent by framing them within statistical decision theory.

Keywords: Identifiability, causal discovery, nonlinear independent component analysis, causal representation learning, identifiable representation learning, extrapolation, compositional generalization, object-centric representation learning

Contents

List of Tables

List of Figures

List of acronyms and abbreviations

DAG	Directed acyclic graph
GraN-DAG	Gradient-based neural DAG learning
DCDI	Differentiable causal discovery from intervention
ICA	Independent component analysis
CGM	Causal graphical model
SEM	Structural equation model
NN	Neural network
ANM	Additive noise model
CPDAG	Completed partially directed acyclic graph
ReLU	Rectified Linear Unit
PNS	Preliminary neighbors selection
SHD	Structural Hamming distance
SID	Structural interventional distance
MEC	Markov equivalence class
MLE	Maximum likelihood estimation
SVM	Support vector machine
MCC	Mean correlation coefficient
LMS	Latent matching score
DCI	Disentanglement, completeness and informativeness scores
OCRL	Object-centric representation learning

Notation

The integers from 1 to n inclusively	$[n]$
Scalars (random or not, depending on context)	$x \in \mathbb{R}$
Vectors (random or not, depending on context)	$\boldsymbol{x} \in \mathbb{R}^n$
Distribution of the random vector $\boldsymbol{x}$	$\mathbb{P}_{\boldsymbol{x}}$
Expectation	$\mathbb{E}$
Support of the random vector $\boldsymbol{x}$	$\text{supp}(\boldsymbol{x}) \subseteq \mathbb{R}^n$
Matrices	$\boldsymbol{A} \in \mathbb{R}^{n \times m}$
Indicator function	$\mathbb{1}(\cdot)$
Euclidean/Frobenius norm on vectors/matrices	$\|\cdot\|$
The $L_{2,1}$ norm	$\|\boldsymbol{A}\|_{2,1} := \sum_{j=1}^{m} \|\boldsymbol{A}_{:,j}\|$
The $L_{2,0}$ norm	$\|\boldsymbol{A}\|_{2,0} := \sum_{j=1}^{m} \mathbb{1}(\|\boldsymbol{A}_{:,j}\| \neq 0)$
Sets are upper-case or calligraphic letters	$S, \mathcal{S}$
Vector formed with the coordinates $\boldsymbol{x}_i$, for all $i \in S \subseteq [n]$	$\boldsymbol{x}_S$
Matrix formed with the entries $\boldsymbol{A}_{i,j}$, for all $(i,j) \in S \times S'$	$\boldsymbol{A}_{S,S'}$
Scalar-valued functions are lower-case	$f : \mathbb{R}^n \to \mathbb{R}$
Vector-valued functions are lower-case bold	$\boldsymbol{f} : \mathbb{R}^n \to \mathbb{R}^m$
Jacobian of $\boldsymbol{f} : \mathbb{R}^n \to \mathbb{R}^m$	$D\boldsymbol{f} : \mathbb{R}^n \to \mathbb{R}^{m \times n}$
Gradient of $f : \mathbb{R}^n \to \mathbb{R}$	$\nabla f : \mathbb{R}^n \to \mathbb{R}^n$
Hessian of $f : \mathbb{R}^n \to \mathbb{R}$	$D^2 f : \mathbb{R}^n \to \mathbb{R}^{n \times n}$
Closure of the set $S \subseteq \mathbb{R}^n$ w.r.t. the $\mathbb{R}^n$ topology	$\overline{S}$
Interior of the set $S \subseteq \mathbb{R}^n$ w.r.t. the $\mathbb{R}^n$ topology	S°
Directed graph with node set V and edge set E	$\mathcal{G} = (V, E)$
(Conditional) independence in distribution $\mathbb{P}_{\boldsymbol{x}}$	$\perp\!\!\!\perp_{\mathbb{P}_{\boldsymbol{x}}}$
d-seperation in the graph $\mathcal{G}$	$\perp\!\!\!\perp_{\mathcal{G}}$

Acknowledgments

I have been very lucky in life so far, especially with the people I have encountered. Everyone mentioned below had a positive impact on the person I am today. As strange as it may sound, I really do feel like most of them are part of "me", to such an extent that I sometimes catch my mind emulating hypothetical discussions or heated debates with some of them, about machine learning, mathematics, science, philosophy or life more broadly. The credit for this thesis also goes to them.

Firstly, I would like to thank my PhD advisor Simon Lacoste-Julien for his commitment to scientific excellence, his skillful pedagogy and his unmatched enthusiasm for... well for everything! My experience with Simon has been a perfect balance between having the freedom to develop my own research interests and benefiting from a very hands-on mentorship on various aspects of academic life such as technical skills, communication skills and how to navigate the very social/human endeavor that is research. Simon has always been very understanding and encouraging, which definitely help me build confidence in my own research abilities. It has been an honor to work with someone as smart, competent and caring as Simon during my PhD.

I thank Sylvie Normandeau, who put me in touch with Yoshua Bengio when I was considering applying to Mila, Yoshua, who helped me preparing my application when I was still an undergraduate student, and Emma Frejinger, who was my supervisor during my first year as a graduate student.

I would also like to express gratitude towards the Mila community which was an ideal environment for me to grow as a researcher and a human being. The breadth of its expertise shaped my view of machine learning research while its cultural diversity enriched my perspective of the world. I thank its members and specifically the people involved in its creation.

The publications in this thesis would not have been possible without the dedicated work of my wonderful collaborators, namely Philippe Brouillard, Tristan Deleu, Divyat Mahajan, Quentin Bertrand, Alexandre Drouin, Alexandre Lacoste, Rémi Le Priol, Pau Rodríguez López, Yash Sharma, Katie Everett, Ioannis Mitliagkas and Yoshua Bengio. It has been a delight doing science with you.

I also thank colleagues who provided valuable revisions and discussions that improved the contributions of this thesis. First contribution: Rémi Le Priol, Tatjana Chavdarova, Charles Guille-Escuret, Nicolas Gagné, Yoshua Bengio, Alexandre Drouin and Florian Bordes. Second contribution: Nicolas Chapados, Rémi Lepriol, Damien Scieur, Assya Trofimov, Jose Gallego, Brady Neal and Grace Abuhamad. Third contribution: Aristide Baratin. Fourth contribution: David Berger.

Some of the friendships I made at Mila had a lasting effect on how I think. I started my PhD at the same time as Tristan Deleu, with whom I went through many classes and who helped me (more than he probably realizes) building my understanding of our field via countless discussions we had throughout the five years we were desk neighbors. Nicolas Gagné somehow always convinced me to learn more mathematics (which I do not regret) and had a lasting impact on how I think about this subject via our many "jam sessions" in front of his white board. I very much enjoyed the productive debates with Sékou-Oumar Kaba (like on that twelve-hour flight to Hawaii), with whom I shared a taste for philosophy. It was a pleasure to bug him with questions about physics too, which I always wished I knew more of.

The day-to-day of research can be exhausting and stressful at times. The friends I made at Mila helped me go through it and made life significantly more enjoyable. I thank Tristan Deleu, Samuel Lavoie, Sékou-Oumar Kaba, Julien Roy, Maude Lizaire, David Kanaa, Sacha Morin, Evgenii Nikishin, Nicolas Gagné, Padideh Nouri, Simon Verret, Simon Guiroy and David Berger for all these little moments.

I was also lucky to have a great social circle outside academic life, which helped me keep balance: I thank Vincent, Camila, Gabriel, Audrey, Quentin, Chloé, Tommy, Arianne, Martin, Stéphanie, Justin, Catherine, Jérémie, Mélina, Mathieu, Myriam, Olivier, Émilie, Bastien, Émilie, Marie-Pier and Joël for all the laughs.

Finally, the most important "thank you" goes to my parents, Denis Lachapelle and Linda Archambault, for their unconditional love and support. My parents always valued hard work and were immensely supportive throughout my life, whether I was pursing my PhD or a career as a professional musician. I am extremely thankful for the family they have built for my two sisters and I (Amélie and Stéphanie), which has always been a point of stability in my life (despite the chaos of the Sunday dinners with all five of my little nephews and nieces!). My mother has always been my biggest fan in all my pursuits. She is probably still adding views to some of my very old drumming videos on Youtube and bragging about me to her friends. My father taught me to be ambitious and shared his passion for science and engineering with me as a kid, whether by teaching me how to read electrical schematics and experiment with my own circuits, or by making us wonder at the stars and Einstein's relativity theory. I still carry his fascination with science in the work I do.

Funding was in part provided by the Canada CIFAR AI Chair Program, by an IVADO excellence PhD scholarship and by Samsung Electronics Co., Ldt. The experiments were in part enabled by computational resources provided by Calcul Québec (calculquebec.ca) and the Digital Research Alliance of Canada (alliancecan.ca).

Chapter 1

Introduction

> *[...] all inferences from experience suppose, as their foundation, that the future will resemble the past, and that similar powers will be conjoined with similar sensible qualities. If there be any suspicion that the course of nature may change, and that the past may be no rule for the future, all experience becomes useless, and can give rise to no inference or conclusion.* — Hume [1748, Section IV - Part II]

The process of making "inferences from experience", called *inductive reasoning*, is at the heart of machine learning. Induction is about using experience to infer a general rule, like when one observes that the sun has risen every day up to now to conclude that the sun will continue rising every day in the future. In the above quote, Hume describes the fundamental assumption underpinning all inductive reasoning: that "the future will resemble the past, and that similar powers will be conjoined with similar sensible qualities", a principle now known as the *uniformity of nature* [Salmon, 1953, Day, 1975]. Without it, experience would be of no use to predict the future. In contrast, *deductive reasoning* refers to the process of discovering statements that are logically entailed by others, like a mathematician deriving new theorems from known ones. While artificial intelligence as a field certainly aims at developing agents capable of both types of reasoning, the subfields of statistics and machine learning are fundamentally about formalizing the former: inductive reasoning.

While this is an important realization, the question of precisely *how* nature is uniform is left open. A large part of machine learning research is about exploring different inductive biases, i.e. assumptions about the data made by the learner. The assumption that observations are *independent and identically distributed* (i.i.d.), which is ubiquitous to both statistics and machine learning, can be thought of as one possible mathematization of "the uniformity of nature": the observations made in the past were generated from a random process that will remain the same in the future. Fundamental ideas such as the consistency of maximum likelihood estimation [Wasserman, 2010] and generalization in statistical machine learning [Mohri et al., 2018, Shalev-Shwartz and Ben-David, 2014] crucially rely on this assumption. Breiman [2001] argued that, historically, statisticians had a

tendency to make stronger parametric assumptions about "how the data came about" compared to machine learning researcher which kept the milder i.i.d. assumption. These allowed statisticians to provide significance tests for interpretable models, such linear regression, at the cost of lesser expressivity. In contrast, machine learning researchers have focused on developing more expressive models with the goal of tackling high-dimensional problems such as image classification and speech recognition where simple parametric models clearly did not apply, even if that meant interpretability was compromised [Breiman, 2001]. The advantage of this approach is exemplified by the triumph of deep learning, the subfield of machine learning focused on very expressive multilayered neural networks [Goodfellow et al., 2016], in applications such as computer vision [Krizhevsky et al., 2012, Radford et al., 2021], natural language processing [Brown et al., 2020] and image generation [Ramesh et al., 2022]. Although some successful architectures do exploit the structure present in the data-modality they were designed for, e.g. convolutional neural networks (CNN) which exploit the translation invariance of object classification, it seems progress in deep learning has been driven largely by growing datasets, computational capabilities and architectural innovations facilitating training; as opposed to exploiting structure present in the data. One can even argue that autoregressive language models such as GPT-3 [Brown et al., 2020] makes even weaker assumptions about the data by training on very long non-i.i.d. sequences of text. In a similar vein, the recent visual transformer (ViT) [Dosovitskiy et al., 2021a] demonstrates that adding further capacity and dropping the translation-invariance of CNNs can yield improved performance when coupled with more data. Despite the impressive progress coming out of this trend towards making models more and more expressive and training them on more and more data, machine learning models still appear to be less data-efficient than humans [Tenenbaum et al., 2011, Lake et al., 2017, Kühl et al., 2022], are hard to interpret, are sensitive to adversarial attacks [Szegedy et al., 2014], and lack robustness to environmental changes [Peters et al., 2016, Magliacane et al., 2018].

This thesis is an attempt at getting the best of both worlds by proposing models that are sufficiently expressive while being restricted enough to be interpretable and amenable to theoretical analyses. In most contributions, we postulate the existence of some "structure" present in the data, i.e. some specific way in which nature is uniform, and provide theoretical guarantees for when this structure can be discovered, allowing improved interpretablity and/or improved generalization. These theoretical results, often building on recent results in nonlinear independent component analysis (ICA) [Hyvärinen et al., 2023], take the form of *identifiability guarantees*, which state that the parameters of a statistical model can be inferred up to some equivalence class from the distribution it entails. Once the structure is identified, the model can be more easily interpreted and can be shown to have performance guarantees on specific downstream tasks (Chapter 8). This approach can also be used to explain the behavior of existing models that are already known to be successful by exposing the structure of the data they unknowingly exploit (Chapter 7). An

emphasis is placed on making assumptions that capture the essence of the problem at hands without compromising expressivity. However, progressively moving towards more realistic assumptions while keeping guarantees remains an important challenge. More future directions are discussed in Chapter 9.

Section 1.1 presents an overview of the structure of this thesis, Section 1.2 summarizes its contributions and Section 1.3 lists the contributions excluded from this thesis.

1.1. Overview of the thesis structure

This thesis is organized around five articles, each of which has its own Prologue contextualizing the work and briefly reviewing recent developments that followed it. In addition, this thesis includes a background summarizing central notions (Chapter 2), a chapter exploring different interpretations of identifiability and unifying most contributions under a three-steps framework (Chapter 8), and a conclusion discussing perspectives for future work (Chapter 9). The five contributions are listed below:

*Equal contributions.

- First Contribution (Chapter 3 & Prologue):
  Gradient-Based Neural DAG Learning by *Sébastien Lachapelle, Philippe Brouillard, Tristan Deleu & Simon Lacoste-Julien*. This work was presented at the 8th International Conference on Learning Representations (ICLR 2020).
- Second Contribution (Chapter 4 & Prologue):
  Differentiable Causal Discovery from Interventional Data by *Philippe Brouillard *, Sébastien Lachapelle *, Alexandre Lacoste, Simon Lacoste-Julien & Alexandre Drouin*. This work was published at the 34th Conference on Neural Information Processing Systems (NeurIPS 2020) with a **spotlight**.
- Third Contribution (Chapter 5 & Prologue):
  Nonparametric Partial Disentanglement via Mechanism Sparsity: Sparse Actions, Interventions and Sparse Temporal Dependencies by *Sébastien Lachapelle, Pau Rodríguez López, Yash Sharma, Katie Everett, Rémi Le Priol, Alexandre Lacoste and Simon Lacoste-Julien.* This work was submitted to the Journal of Machine Learning Research in 2024. This is a significantly extended version of two works: one published at the 1st Conference on Causal Learning and Reasoning (CLeaR 2022) and one presented at the 1st Workshop on Causal Representation Learning at UAI 2022, the latter of which received an **oral** and a **best paper award**.
- Fourth Contribution (Chapter 6 & Prologue):
  Synergies between Disentanglement and Sparsity: Generalization and Identifiability in Multi-Task Learning by *Sébastien Lachapelle*, Tristan Deleu*, Divyat Mahajan, Ioannis*

Mitliagkas, Yoshua Bengio, Simon Lacoste-Julien and Quentin Bertrand. This work was published at the 40th International Conference on Machine Learning (ICML 2023).

- Fifth Contribution (Chapter 7 & Prologue):
  Additive Decoders for Latent Variables Identification and Cartesian-Product Extrapolation by *Sébastien Lachapelle*, Divyat Mahajan*, Ioannis Mitliagkas and Simon Lacoste-Julien*. This work was published at the 37th Conference on Neural Information Processing Systems (NeurIPS 2023) with an **oral**.

1.2. Research contributions

The first and second contributions are concerned with the problem of causal discovery (Chapters 3 & 4) while the third, fourth and fifth contributions are about identifiable representation learning (Chapters 5, 6 & 7). These contributions are then unified under one simple framework in Chapter 8 which highlights how identifiability can be seen as an intermediate step when proving generalization guarantees.

1.2.1. Gradient-based causal discovery (Chapters 3 & 4)

The weaknesses of deep learning systems motivated a recent surge of interest in *causality* [Pearl, 2019, Schölkopf, 2019, Schölkopf et al., 2021, Goyal and Bengio, 2021]. In a causal model, each variable is determined by a *causal mechanism* which takes as input other variables: its causal parents. Importantly, these mechanisms are assumed to remain unchanged unless they are targeted by an *intervention*, i.e. a change to the causal system affecting only a few mechanisms. This can be seen as another formalization of the principle of uniformity of nature which relaxes the "identically distributed" in "i.i.d." by allowing the distribution to change, although in some limited way (only a few mechanisms can change). The various causal relationships can be summarized by a directed acyclic graph (DAG) called a *causal graph*. When this graph is known, it can be used to predict the effect of interventions in the system, such as what will be the effect of taking some treatment on the health status of a patient, without actually having to perform the intervention in the real world. Measuring these effects is the concern of *causal inference*. However, in many applications, the causal graph is unknown, which means it must be discovered from data. This is the problem of *causal discovery*, which is the subject of the first two contributions.

Chapters 3 & 4 tackle the problem of learning a causal graph from data.[1] Both contributions build on the work of Zheng et al. [2018] which proposed to reformulate the inherently discrete problem of searching over the space of DAGs into a continuous constrained optimization problem. This formulation allows the exploration of drastically different optimization algorithms such as the augmented Lagrangian procedure (Section 2.6.1). The first contribution extended this work

[1] Strictly speaking, Chapter 3 does not require any causal interpretation.

to allow for nonlinear dependencies, thanks to neural networks, while the second contribution showed how this approach can be adapted to leverage interventional data. Benefits of this approach include a favorable computational complexity as a function of sample size, thanks to stochastic gradient descent, (Chapter 4) and an ease of integration with deep learning models. The Prologue of Chapter 3 and the Prologue of Chapter 4 provide further context for how these projects came about, discuss limitations and review recent works addressing these challenges.

1.2.2. Identifiable representation learning (Chapters 5, 6 & 7)

Chapters 5, 6 & 7 provide novel identifiability guarantees in representation learning. These give theoretical grounding for how to extract *disentangled factors of variations* from high-dimensional observations such as images [Bengio et al., 2013]. The term "disentangled" is used to describe representations in which "natural factors of variations" such as object positions, colors or sizes are represented individually as single coordinates. Disentanglement is difficult largely due to the problem of unidentifiability: many representations which are "not natural" yield as good a fit to the data as the "natural one". This issue was already present in simple linear models [Hyvärinen et al., 2001] and got much worse with more expressive neural networks [Hyvärinen and Pajunen, 1999, Locatello et al., 2020b]. The results introduced in the following contributions always restrain the expressivity of the model to get rid of the "unnatural representations" and assume the data is generated from a ground-truth model, often building on the seminal work in nonlinear ICA which first showed that the latent factors can be identified even in the nonlinear mixing case [Hyvarinen and Morioka, 2016, 2017, Hyvärinen et al., 2019, Khemakhem et al., 2020a]. One of the main motivations for learning disentangled representations is to make deep learning models easier to interpret, but also to easily obtain representations that are invariant to certain factors of variations. The following contributions also uncover novel ways in which disentanglement can be beneficial for downstream performance (also see Chapter 8). See Section 5.7 for an exhaustive literature review on identifiable representation learning.

Chapter 5 studies the identifiability of a deep latent variable model (Section 2.5) in which sequences of high-dimensional observations $\{\boldsymbol{x}^t\}$ such as images are explained by a sequence of lower-dimensional latent factors of variations $\{\boldsymbol{z}^t\}$ via $\boldsymbol{x}^t = \boldsymbol{f}(\boldsymbol{z}^t)$ where $\boldsymbol{f}$ is a deep neural network. Identifiability of the latent factors is obtained by assuming that they are related together via a sparse causal graphical model, which might include auxiliary variables such as actions and/or an environment index. We provide conditions such that fitting this model while regularizing the latent causal graph to be sparse entails disentanglement. While other works have leveraged independence of latent variables in a temporal setting [Tong et al., 1993, Hyvarinen and Morioka, 2017, Klindt et al., 2021], this contribution was the first to show that more permissive forms of sparse temporal dependencies are sometimes enough to disentangle. This work was also among the first, concurrently

with Lippe et al. [2022], to show that interventions on latent variables can be enough to disentangle them, a principle previously hypothesized by Schölkopf et al. [2021] without formal proofs. The Prologue of Chapter 5 provides further context and describes recent developments which build on this contribution.

Chapter 6 explores a multi-task learning setting in which every prediction task has the form $\boldsymbol{y} = \boldsymbol{w}^\top \boldsymbol{f}(\boldsymbol{x})$ where $\boldsymbol{y}$ is a label, $\boldsymbol{x}$ is an image, $\boldsymbol{f}$ is a representation fixed across tasks and $\boldsymbol{w}$ is sparse weight vector that changes from one task to another. We propose solving a bilevel optimization problem in which $\boldsymbol{f}$ is learned in the outer-problem while the task-specific weight vector $\boldsymbol{w}$ is learned in the inner-problem, with sparsity regularization. Importantly, we show that solving this bilevel optimization problem yields a disentangled representation, under some conditions on both the data- and task-generating processes. We also provide a simple but rigorous argument for why a disentangled representation is advantageous in a few-shot learning setting where the future unknown task is sparse. In the Prologue of Chapter 6, I explain how this can be seen as a formalization of an idea formulated by Bengio et al. [2013] and mention a few recent works which leveraged the proof techniques we introduced.

Chapter 7 is about leveraging the additive structure of simple images consisting of multiple objects for both disentanglement and extrapolation. In Chapter 5, disentanglement was enabled by a restriction on the distribution of the latent factors (sparse dependencies), while here we instead restrict the decoder $\boldsymbol{f}$ to be additive and show this makes the latent factors identifiable. Although additive decoders are very simple and restrictive, they bear similarities with the more expressive decoders used in object-centric learning [Locatello et al., 2020c]. Studying the identifiability of additive decoders might help us gain some theoretical understanding as to why object-centric decoders can perform segmentations without any supervision. In addition, we show that additivity allows generation of images that were not part of the training distribution, but that are still on the manifold of reasonable images. We speculate that this kind of identifiability analysis leading to extrapolation guarantees might be applied to understand creativity in modern text-to-image models [Ramesh et al., 2022].

1.2.3. Interpretations of identifiability and motivations for downstream performance (Chapter 8)

In addition to the five articles described above, Chapter 8 explores three interpretations of identifiability and motivates the study of identifiability as an intermediate step when proving downstream performance guarantees. I propose a simple three-step framework highlighting the role of identifiability for proving generalization guarantees and illustrate it with four seemingly unrelated problem settings, three of which are based on contributions of this work. The connections are made more rigorous by framing all four problem settings within statistical decision theory.

1.3. Excluded publications

The above is a list of publications I have contributed to during my PhD that I decided to exclude.

- **A Meta-Transfer Objective for Learning to Disentangle Causal Mechanisms** by *Yoshua Bengio, Tristan Deleu, Nasim Rahaman, Rosemary Ke, Sébastien Lachapelle, Olexa Bilaniuk, Anirudh Goyal and Christopher Pal.* This work was presented at the 8th International Conference on Learning Representations (ICLR 2020).
- **On the Convergence of Continuous Constrained Optimization for Structure Learning** by *Ignavier Ng, Sébastien Lachapelle, Nan Rosemary Ke and Simon Lacoste-Julien*. This work was presented at the 25th International Conference on Artificial Intelligence and Statistics (AISTATS 2022).
- **Disentanglement via Mechanism Sparsity Regularization: A New Principle for Nonlinear ICA** by *Sébastien Lachapelle, Pau Rodríguez López, Yash Sharma, Katie Everett, Rémi Le Priol, Alexandre Lacoste and Simon Lacoste-Julien.* This work was published at the 1st Conference on Causal Learning and Reasoning (CLeaR 2022).
- **Typing assumptions improve identification in causal discovery** by *Philippe Brouillard, Perouz Taslakian, Alexandre Lacoste, Sébastien Lachapelle and Alexandre Drouin.* This work was published at the 1st Conference on Causal Learning and Reasoning (CLeaR 2022) with an **oral**.

Chapter 2

Background

In this chapter, we cover some important notions necessary to understand this thesis.

- Section 2.1 provides a brief introduction to basic notions of probability such as probability measures, random variables and (conditional) independence.
- Section 2.2 gives an introduction to the framework of statistical decision theory, maximum likelihood estimation, the bias-variance trade-off and identifiability.
- Section 2.3 introduces causal graphical models, how they support interventional queries and the important Markov property.
- Section 2.4 covers briefly both constraint-based and score-based approach to the problem of causal discovery, which consist of learning a causal graph from data.
- Section 2.5 gives a quick overview of existing approaches to representation learning with a focus on identifiability in latent variable models and independent component analysis (ICA)
- Section 2.6 covers the basics of constrained optimization, leading up to the augmented Lagrangian method.
- Section 2.7 gives brief descriptions of two popular gradient estimators, namely REINFORCE and the reparametrization trick.

2.1. Elementary probability theory

Probability theory is the branch of mathematics which deals with uncertainty. It provides a coherent framework to describe quantitatively how much is known about a specific system and provides tools to answer various queries about its precise state. Since this tool is fundamental to the contribution of this report and to machine learning in general, we present a quick summary of important notions. For a more in-depth presentation which avoids references to measure theory, we refer the reader to Ross [2010].

The set of possible states a system can be in is called the *sample space* and is typically denoted by Ω. Elements of Ω, denoted by ω, are called *outcomes* while subsets of Ω, denoted by E, are

called *events*. A *probability measure* $\mathbb{P}$ assigns to each event $E \subseteq \Omega$ a number $\mathbb{P}(E) \in [0,1]$ which described how likely it is that the outcome is in E. By definition, a probability measure must also satisfy the following three axioms: (i) for all $E \subseteq \Omega$, $\mathbb{P}(E) \in [0,1]$, (ii) $\mathbb{P}(\Omega) = 1$, and (iii) for a countable sequence of mutually disjoint events $E_1, E_2, ...$, we have that $\mathbb{P}(\bigcup_{i=1}^{\infty} E_i) = \sum_{i=1}^{\infty} \mathbb{P}(E_i)$. Strictly speaking, if $\Omega = \mathbb{R}^n$, it is impossible to define a probability measure $\mathbb{P}$ (which by definition satisfies all three axioms) *that is defined over all subsets of* $\mathbb{R}^n$. The standard solution to circumvent this problem is to restrict the domain of $\mathbb{P}$ to *sufficiently nice* subsets of $\mathbb{R}^n$ so that the axioms can be satisfied. These nice sets are called *Lebesgue measurable*, but we will not present the definition as most sets encountered in practice are Lebesgue measurable. See Durrett [2011] for details.

A *random vector* is a function $\boldsymbol{x} : \Omega \to \mathbb{R}^d$ and the *distribution* of $\boldsymbol{x}$ is a probability measure defined as $\mathbb{P}_{\boldsymbol{x}}(E) = \mathbb{P}(\boldsymbol{x}^{-1}(E))$ for all events $E \subseteq \mathbb{R}^d$, where $\boldsymbol{x}^{-1}(E)$ denotes the preimage of E under $\boldsymbol{x}$. The output of a random vector $\boldsymbol{x}(\omega) \in \mathbb{R}^d$ can be thought of as a numerical measurement of the state of the system $\omega \in \Omega$.

Throughout this chapter, we will assume that $\mathbb{P}_{\boldsymbol{x}}$ can be written as $\mathbb{P}_{\boldsymbol{x}}(E) = \int_E p(\boldsymbol{x}) d\boldsymbol{x}$ or $\mathbb{P}_{\boldsymbol{x}}(E) = \sum_{\boldsymbol{x} \in E} p(\boldsymbol{x})$ where $p : \mathbb{R}^d \to [0, \infty)$.[1] In the first case, we say $\boldsymbol{x}$ is *continuous* and p is called a *probability density function*, while in the second case, we say $\boldsymbol{x}$ is *discrete* and p is called a *probability mass function*. The second axiom of probability measures implies that $\int p(\boldsymbol{x}) d\boldsymbol{x} = 1$ or $\sum_{\boldsymbol{x}} p(\boldsymbol{x}) = 1$. In the following definitions, we use integrals everywhere, but one can replace them by sums to obtain equivalent definitions for discrete random vectors. Note that, in many circumstances, we define a probability measure by first specifying its density/mass function. However, one should keep in mind that not all probability measures can be expressed with density/mass functions as we defined here.

Notation. Given an integer n, we use the shorthand $[n]$ to denote the set $\{1, ..., n\}$. Given a set $S \subseteq [d]$, we write $\boldsymbol{x}_S$ to refer to the random vector containing random variables $\boldsymbol{x}_j$ for $j \in S$ and write $\boldsymbol{x}_{-S}$ to refer to the vector containing random variables $\boldsymbol{x}_j$ for $j \in [d] \setminus S$. We use $\boldsymbol{x}_S$ to denote both the random vector $\boldsymbol{x}_S : \Omega \to \mathbb{R}^{|S|}$ and a realization $\boldsymbol{x}_S \in \mathbb{R}^{|S|}$ since both meanings can always be disambiguated from context in this thesis. When writing $\int_A f(\boldsymbol{x}) d\boldsymbol{x}_S$ for some $A \subseteq \mathbb{R}^{|S|}$ and some function $f : \mathbb{R}^d \to \mathbb{R}$, we assume integration with respect to the Lebesgue measure.

We now present the notion of marginal distributions which allows us to answer probabilistic queries concerning only a subset of variables in the system.

Definition 2.1. *(Marginal density) Let $\boldsymbol{x} : \Omega \to \mathbb{R}^d$ be a random vector and let $\mathbb{P}_{\boldsymbol{x}}$ be its distribution with density p. Given a proper subset $S \subseteq \{1, ..., d\}$, the marginal density of $\boldsymbol{x}_S$ is*

$$p(\boldsymbol{x}_S) := \int_{\mathbb{R}^{d-|S|}} p(\boldsymbol{x}) d\boldsymbol{x}_{-S} \,. \tag{2.1}$$

[1] Note that we use $\boldsymbol{x}$ to denote both the random vector $\boldsymbol{x} : \Omega \to \mathbb{R}^d$ and a realization of the random vector $\boldsymbol{x} \in \mathbb{R}^d$, since, in this thesis, context is always sufficient to disambiguate the two possible meanings.

For instance, given a subset $E \subseteq \mathbb{R}^{|S|}$, the integral $\int_E p(\boldsymbol{x}_S) d\boldsymbol{x}_S$ gives the probability that $\boldsymbol{x}_S \in E$.

Conditional probabilities describe the probability of an event occurring given another event occurred. This notion is important to define the notion of conditional independence.

Definition 2.2. *(Conditional density) Let $\boldsymbol{x} : \Omega \to \mathbb{R}^d$ be a random vector and let $\mathbb{P}_{\boldsymbol{x}}$ be its distribution with density p. Given two disjoint subsets $A, B \subseteq \{1, ..., d\}$, the conditional density of $\boldsymbol{x}_A$ given $\boldsymbol{x}_B = \boldsymbol{x}_B^0$ is*

$$p(\boldsymbol{x}_A | \boldsymbol{x}_B^0) := \frac{p(\boldsymbol{x}_A, \boldsymbol{x}_B^0)}{p(\boldsymbol{x}_B^0)}, \tag{2.2}$$

where we assumed $p(\boldsymbol{x}_B^0) > 0$.

For instance, given a subset $E \subseteq \mathbb{R}^{|A|}$, the integral $\int_E p(\boldsymbol{x}_A \mid \boldsymbol{x}_B) d\boldsymbol{x}_A$ gives the probability that $\boldsymbol{x}_A \in E$ given that $\boldsymbol{x}_B = \boldsymbol{x}_B^0$.

We now introduce the notion of (conditional) independence, a central notion in probabilistic graphical models.

Definition 2.3. *((Conditional) independence) Let $\boldsymbol{x} : \Omega \to \mathbb{R}^d$ be a random vector and let $\mathbb{P}_{\boldsymbol{x}}$ be its distribution with density p. Given two disjoint sets $A, B \subseteq [d]$, we say $\boldsymbol{x}_A$ is independent of $\boldsymbol{x}_B$ when*

$$p(\boldsymbol{x}_A, \boldsymbol{x}_B) = p(\boldsymbol{x}_A) p(\boldsymbol{x}_B), \ \forall \boldsymbol{x}_A, \boldsymbol{x}_B \,. \tag{2.3}$$

When this is the case, we write $\boldsymbol{x}_A \perp\!\!\!\perp_{\mathbb{P}_{\boldsymbol{x}}} \boldsymbol{x}_B$. Given three disjoint sets $A, B, C \subseteq [d]$, we say that $\boldsymbol{x}_A$ is conditionally independent of $\boldsymbol{x}_B$ given $\boldsymbol{x}_C$ whenever

$$p(\boldsymbol{x}_A, \boldsymbol{x}_B \mid \boldsymbol{x}_C) = p(\boldsymbol{x}_A \mid \boldsymbol{x}_C) p(\boldsymbol{x}_B \mid \boldsymbol{x}_C), \ \forall \boldsymbol{x}_A, \boldsymbol{x}_B, \boldsymbol{x}_C \text{ s.t. } p(\boldsymbol{x}_C) > 0 \,. \tag{2.4}$$

When this is the case, we write $\boldsymbol{x}_A \perp\!\!\!\perp_{\mathbb{P}_{\boldsymbol{x}}} \boldsymbol{x}_B \mid \boldsymbol{x}_C$.

To get an intuitive understanding of the notion of independence, we can rewrite $p(\boldsymbol{x}_A, \boldsymbol{x}_B) = p(\boldsymbol{x}_A) p(\boldsymbol{x}_B)$ as $p(\boldsymbol{x}_A \mid \boldsymbol{x}_B) = p(\boldsymbol{x}_A)$ (assuming $p(\boldsymbol{x}_B) > 0$) which tells us that knowing the value of $\boldsymbol{x}_B$ does not modify our belief about the value of $\boldsymbol{x}_A$. Analogously for *conditional* independence, we can rewrite $p(\boldsymbol{x}_A, \boldsymbol{x}_B \mid \boldsymbol{x}_C) = p(\boldsymbol{x}_A \mid \boldsymbol{x}_C) p(\boldsymbol{x}_B \mid \boldsymbol{x}_C)$ as $p(\boldsymbol{x}_A \mid \boldsymbol{x}_B, \boldsymbol{x}_C) = p(\boldsymbol{x}_A \mid \boldsymbol{x}_C)$ (assuming $p(\boldsymbol{x}_B \mid \boldsymbol{x}_C) > 0$) which tells us that when knowing the value of $\boldsymbol{x}_C$, additionally knowing the value of $\boldsymbol{x}_B$ does not change our belief about $\boldsymbol{x}_A$.

2.2. Statistical decision theory

At its core, *statistical decision theory* is a framework to analyze and compare different decision-making strategies under uncertainty. This section is inspired by the exposition of Berger [1985] and Lacoste–Julien et al. [2011].

Central to this framework is the idea that the *state of nature* is captured by an unknown parameter $\boldsymbol{\theta}$ which is assumed to belong to a set of *possible states* Θ. The decision maker must take an *action* $\boldsymbol{a}$ among a set of *possible actions* A. If the state of the world happens to be $\boldsymbol{\theta}_0 \in \Theta$ and the action $\boldsymbol{a}_0 \in A$ is taken, then a cost $\ell(\boldsymbol{\theta}_0, \boldsymbol{a}_0)$ is incurred. Thus, the *loss function* $\ell : \Theta \times A \to \mathbb{R}$ gives the cost incurred for all combinations $(\boldsymbol{\theta}, \boldsymbol{a})$. It is further assumed that the decision maker can base its decision on an observation D (for example a dataset of multiple observations) which reveals information about the state of the world $\boldsymbol{\theta}$. This observation is assumed to be a realization of some distribution $\mathbb{D}_{\boldsymbol{\theta}}$. The observation D is assumed to belong to a *sample space* denoted by $\mathcal{D}$. The decision process is modelled by a decision rule $\boldsymbol{\delta} : \mathcal{D} \to A$ which associates an action $\boldsymbol{a} \in A$ to each observation $D \in \mathcal{D}$. With this notation in mind, the loss incurred by rule $\boldsymbol{\delta}$ when the state of the world is $\boldsymbol{\theta}$ and D is observed is given by $\ell(\boldsymbol{\theta}, \boldsymbol{\delta}(D))$, which is random since D is random. It is customary to analyze the expectation of this cost, which is called the *risk*:

$$r(\boldsymbol{\theta}, \boldsymbol{\delta}) := \mathbb{E}_{D \sim \mathbb{D}_{\boldsymbol{\theta}}} \ell(\boldsymbol{\theta}, \boldsymbol{\delta}(D)) \,.$$

Note that, in principle, other summarizations of the random cost $\ell(\boldsymbol{\theta}, \boldsymbol{\delta}(D))$ could be analyzed, like the probability that it is smaller that some threshold value ϵ, as is the subject of *probably approximately correct* learning (PAC) [Mohri et al., 2018].

The goal of statistical decision theory is to compare various decision rules. Since the risk $r(\boldsymbol{\theta}, \boldsymbol{\delta})$ depends not just on the rule $\boldsymbol{\delta}$, but also on the state of the world $\boldsymbol{\theta}$, we must "aggregate" further. For instance, one could consider the *worst-case risk* $\max_{\boldsymbol{\theta} \in \Theta} r(\boldsymbol{\theta}, \boldsymbol{\delta})$ or a *weighted risk* of the form $\int r(\boldsymbol{\theta}, \boldsymbol{\delta}) \pi(\boldsymbol{\theta}) d\boldsymbol{\theta}$ where $\pi(\boldsymbol{\theta})$ is a probability density that can be interpreted as the *belief* the decision maker holds before taking action. Interestingly, a decision rule can be optimal for one criterion and not another, indicating that no decision rule is universally optimal.

Most problems in statistics and machine learning can be formulated in the language of decision theory. For instance, the problem of **parameter estimation** corresponds to accurately guessing $\boldsymbol{\theta}$, so that $A := \Theta$. A natural loss function here would be $\ell(\boldsymbol{\theta}, \boldsymbol{a}) := \|\boldsymbol{\theta} - \boldsymbol{a}\|_2^2$. In its most standard form, D corresponds to a dataset of n independent observations $(\boldsymbol{x}^{(1)}, \ldots, \boldsymbol{x}^{(n)})$ that are identically distributed according to some distribution $\mathbb{P}_{\boldsymbol{\theta}}$ parameterized by $\boldsymbol{\theta} \in \Theta$. In other words, $D := (\boldsymbol{x}^{(1)}, \ldots, \boldsymbol{x}^{(n)})$ is distributed according to $\mathbb{D}_{\boldsymbol{\theta}} := \mathbb{P}_{\boldsymbol{\theta}}^n$, where $\mathbb{P}_{\boldsymbol{\theta}}^n$ denotes the product measure $\mathbb{P}_{\boldsymbol{\theta}} \times \cdots \times \mathbb{P}_{\boldsymbol{\theta}}$ (n times).

A closely related setting would be **density estimation**, where one cares only about finding a probability distribution which is close to ground-truth data generating distribution. Formally, we observe a dataset $D := (\boldsymbol{x}^{(1)}, \ldots, \boldsymbol{x}^{(n)})$ sampled from $\mathbb{D}_{\boldsymbol{\theta}} := \mathbb{P}^n$ and would like to find a distribution $\hat{\mathbb{P}}$ that is close to $\mathbb{P}$. One natural loss function for this setting would be $\ell(\mathbb{P}, \hat{\mathbb{P}}) := D_{KL}(\mathbb{P} || \hat{\mathbb{P}})$, where D_{KL} denotes the *Kullback-Leibler divergence* which is defined by $D_{KL}(\mathbb{P} || \hat{\mathbb{P}}) := \int p(\boldsymbol{x}) \log \frac{p(\boldsymbol{x})}{\hat{p}(\boldsymbol{x})} d\boldsymbol{x}$

where p and $\hat{p}$ are the densities of $\mathbb{P}$ and $\hat{\mathbb{P}}$ w.r.t. to the Lebesgue measure (which we assume exist).[2] Note that, in this setting, the unknown state of the world $\boldsymbol{\theta}$ would be $\mathbb{P}$ itself, i.e. $\boldsymbol{\theta} := \mathbb{P}$. Finally, we have that the decision rule $\boldsymbol{\delta}$ maps datasets D to distributions $\hat{\mathbb{P}}$.

Another example would be **hypothesis testing**, where $A := \{0, 1\}$ corresponds to either accepting or rejecting the null hypothesis and where ℓ could capture the cost of committing *false negative* and *false positive* errors.

In **supervised machine learning**, the observation would be a dataset $D := ((\boldsymbol{x}^{(1)}, \boldsymbol{y}^{(1)}), \dots, (\boldsymbol{x}^{(n)}, \boldsymbol{y}^{(n)}))$ consisting of independent input-output pairs identically distributed according to some unknown distribution $\boldsymbol{\theta} := \mathbb{P}$ so that $D \sim \mathbb{D}_{\boldsymbol{\theta}} := \mathbb{P}^n$. An action corresponds to a predictor function $\boldsymbol{f}$ mapping inputs to outputs and the decision rule $\boldsymbol{\delta}$ would correspond to a learning procedure taking as input the dataset D, and outputting a predictor $\boldsymbol{f}$, i.e. $\boldsymbol{\delta}(D) = \boldsymbol{f}$. In the case of regression, a typical choice of loss function would be $\ell(\mathbb{P}, \boldsymbol{f}) := \mathbb{E}_{(\boldsymbol{x},\boldsymbol{y})\sim\mathbb{P}}\|\boldsymbol{y} - \boldsymbol{f}(\boldsymbol{x})\|_2^2$. In machine learning, this is typically called the *generalization error*.

2.2.1. Maximum likelihood estimation (MLE) & identifiability

Let us focus on the problem of parameter estimation where we observe a dataset $D := (\boldsymbol{x}^{(1)}, \dots, \boldsymbol{x}^{(n)})$ sampled from $\mathbb{D}_{\boldsymbol{\theta}} := \mathbb{P}_{\boldsymbol{\theta}}^n$ and must produce an estimate $\hat{\boldsymbol{\theta}}$ that is close to the "ground-truth" parameter $\boldsymbol{\theta}$, which we know to be in some parameter space Θ. Let us assume further that, for every $\boldsymbol{\theta} \in \Theta$, $\mathbb{P}_{\boldsymbol{\theta}}$ has a density w.r.t. the Lebesgue measure given by $p(\boldsymbol{x}; \boldsymbol{\theta})$. A standard strategy for this setting is *maximum likelihood estimation* (MLE), which corresponds to choosing a distribution that maximizes the so-called *likelihood function*:

$$\hat{\boldsymbol{\theta}}_{\text{MLE}}^{(n)} \in \arg\max_{\boldsymbol{\theta}' \in \Theta} \sum_{i=1}^{n} \log p(\boldsymbol{x}^{(i)}; \boldsymbol{\theta}') =: L^{(n)}(\boldsymbol{\theta}') \,.$$

In the language of decision theory, the corresponding decision rule is given by

$$\boldsymbol{\delta}_{\text{MLE}}((\boldsymbol{x}^{(1)}, \dots, \boldsymbol{x}^{(n)})) := \hat{\boldsymbol{\theta}}_{\text{MLE}}^{(n)} \,.$$

It is well-known that, under regularity conditions on $p(\boldsymbol{x}; \boldsymbol{\theta})$, the estimator $\hat{\boldsymbol{\theta}}_{\text{MLE}}^{(n)}$ is *consistent* in the sense that $\ell(\boldsymbol{\theta}, \hat{\boldsymbol{\theta}}_{\text{MLE}}^{(n)}) := \|\boldsymbol{\theta} - \hat{\boldsymbol{\theta}}_{\text{MLE}}^{(n)}\|_2^2 \to 0$ (in probability) as $n \to \infty$ [Wasserman, 2010]. One of these regularity conditions requires $\boldsymbol{\theta}$ to be *identifiable from* $\mathbb{P}_{\boldsymbol{\theta}}$:

$$\forall \boldsymbol{\theta}, \boldsymbol{\theta}' \in \Theta,\ \mathbb{P}_{\boldsymbol{\theta}} = \mathbb{P}_{\boldsymbol{\theta}'} \implies \boldsymbol{\theta} = \boldsymbol{\theta}' \,. \tag{2.5}$$

[2] The KL-divergence can be defined more generally for arbitrary (σ-finite) measures as long as $\hat{\mathbb{P}}(E) = 0 \implies \mathbb{P}(E) = 0$ for all events E (in which case we say that $\hat{\mathbb{P}}$ is *absolutely continuous* w.r.t. $\mathbb{P}$) and is given by $D_{KL}(\mathbb{P}||\hat{\mathbb{P}}) := \int \log \frac{d\mathbb{P}}{d\hat{\mathbb{P}}} d\mathbb{P}$ where $\frac{d\mathbb{P}}{d\hat{\mathbb{P}}}$ is the *Radon-Nikodym derivative* of $\mathbb{P}$ w.r.t. $\hat{\mathbb{P}}$ and the integral refers to the *Lebesgue integral* with base measure $\mathbb{P}$. This thesis avoids most of these technicalities by considering only continuous or discrete random variables.

This means that a parameter $\boldsymbol{\theta}$ can always be uniquely determined from the distribution $\mathbb{P}_{\boldsymbol{\theta}}$ it entails. Put differently, identifiability means that the map $\boldsymbol{\theta} \mapsto \mathbb{P}_{\boldsymbol{\theta}}$ is injective. For example, in the usual family of Gaussian distributions, the parameter (μ, σ^2) is identifiable from the distribution. However, if instead we parameterize with $\mu = \alpha + \beta$, the parameter $(\alpha, \beta, \sigma^2)$ is not identifiable since multiple choices of parameter yield the same distribution.

In order to understand the role of identifiability, we present an informal argument for why the maximum likelihood estimator is consistent.

Proof sketch for the consistency of $\hat{\boldsymbol{\theta}}^{(n)}_{\text{MLE}}$. We follow the presentation of Wasserman [2010, p.127]. Before starting, we recall a crucial property of the KL-divergence. In general, $D_{KL}(\mathbb{P}||\hat{\mathbb{P}}) \geq 0$ with equality if and only if $\mathbb{P} = \hat{\mathbb{P}}$. Define

$$D(\hat{\boldsymbol{\theta}}) := D_{KL}(\mathbb{P}_{\boldsymbol{\theta}}||\mathbb{P}_{\hat{\boldsymbol{\theta}}}) \text{ and } D^{(n)}(\hat{\boldsymbol{\theta}}) := \frac{1}{n}\sum_{i=1}^{n} \log \frac{p(\boldsymbol{x}^{(i)}; \boldsymbol{\theta})}{p(\boldsymbol{x}^{(i)}; \hat{\boldsymbol{\theta}})} d\boldsymbol{x}\,.$$

Since $D^{(n)}(\hat{\boldsymbol{\theta}}) = n^{-1}(L^{(n)}(\boldsymbol{\theta}) - L^{(n)}(\hat{\boldsymbol{\theta}}))$, it is clear that maximizing the log-likelihood $L^{(n)}(\hat{\boldsymbol{\theta}})$ is the same as minimizing $D^{(n)}(\hat{\boldsymbol{\theta}})$. In other words, the maximum likelihood estimator is a minimizer of $D^{(n)}(\hat{\boldsymbol{\theta}})$. Furthermore, the law of large numbers guarantees that $D^{(n)}(\hat{\boldsymbol{\theta}}) \to D(\hat{\boldsymbol{\theta}})$ as $n \to \infty$ (in probability). This observation suggests that the minimizer of $D^{(n)}(\hat{\boldsymbol{\theta}})$ should converge in probability to a minimizer of $D(\hat{\boldsymbol{\theta}})$. Because the parameter $\boldsymbol{\theta}$ is identifiable, $\boldsymbol{\theta}$ is the unique minimizer of $D(\hat{\boldsymbol{\theta}})$. Hence, intuitively, we should have that $\hat{\boldsymbol{\theta}}^{(n)}_{\text{MLE}}$ converges to $\boldsymbol{\theta}$ in probability. While this argument is informal, it can be made rigorous by adding further regularity assumptions [Wasserman, 2010, p.127], but this is outside the scope of this thesis. ■

One can question the relevance of recovering the "correct" parameter $\boldsymbol{\theta}$. What if we only care about modelling the data faithfully? This goal is better captured by the problem of density estimation where the loss function is given by $\ell(\mathbb{P}, \hat{\mathbb{P}}) := D_{KL}(\mathbb{P}, \hat{\mathbb{P}})$. In fact, to formulate this setting, one does not even have to specify a ground-truth parameter in the first place. The "state of nature" $\boldsymbol{\theta}$, to use the terminology of decision theory, is the data-generating distribution itself, i.e. $\boldsymbol{\theta} := \mathbb{P}$. That being said, one can still apply MLE to estimate $\mathbb{P}$. If one choose a parametric family $\{\mathbb{P}_{\boldsymbol{\eta}} \mid \boldsymbol{\eta} \in H \subseteq \mathbb{R}^k\}$ expressive enough to contain $\mathbb{P}$, an argument exactly analogous to the one presented above can be used to show that MLE is consistent in the sense that $D_{KL}(\mathbb{P}||\mathbb{P}_{\hat{\boldsymbol{\eta}}^{(n)}_{\text{MLE}}}) \to 0$ (in probability) as $n \to \infty$. Do not conflate $\boldsymbol{\eta}$, which is the parameter of the model, and $\boldsymbol{\theta}$, which is the "state of nature". Again, regularity conditions are still required to make the argument formal, but the key point is that we can get away without identifiability this time. In light of this observation, is there any reason to care about identifiability? We will answer this question in Section 2.2.3 briefly and in more depth in Chapter 8.

2.2.2. Bias-variance trade-off

Although consistency is a good indication that a decision rule will be good for very large datasets, it does not say anything about how fast the estimator approaches its target, which is typically referred to as the *sample complexity* of the estimator. Such analyzes are important to understand the behavior of a decision rule when the number of samples is limited, as they uncover the famous *bias-variance trade-off* which is absolutely central to statistics [Wasserman, 2010] and machine learning [Hastie et al., 2009, Mohri et al., 2018]. In the context of supervised regression with loss $\ell(\mathbb{P}, \hat{f}) := \mathbb{E}_{(\boldsymbol{x},y)\sim\mathbb{P}}(y - \hat{f}(\boldsymbol{x}))^2$, one can define the bias and variance of an estimator $\hat{f}_D$ at $\boldsymbol{x}$ as

$$\text{bias}(\hat{f}_D, \boldsymbol{x}) := \mathbb{E}_D(\hat{f}_D(\boldsymbol{x}) - \mathbb{E}(y \mid \boldsymbol{x})) \tag{2.6}$$

$$\text{var}(\hat{f}_D, \boldsymbol{x}) := \mathbb{E}_D(\hat{f}_D(\boldsymbol{x}) - \mathbb{E}_D\hat{f}_D(\boldsymbol{x}))^2\,, \tag{2.7}$$

where $D := ((\boldsymbol{x}^{(1)}, y^{(1)}), ..., (\boldsymbol{x}^{(n)}, y^{(n)})) \sim \mathbb{P}^n$. One can show that $f^*(\boldsymbol{x}) := \mathbb{E}(y \mid \boldsymbol{x})$ is the minimizer of $\mathbb{E}_{(\boldsymbol{x},y)\sim\mathbb{P}}(y - \hat{f}(\boldsymbol{x}))^2$, which implies that the bias measures how close the expectation of $\hat{f}_D(\boldsymbol{x})$ is to the optimal prediction $\mathbb{E}(y \mid \boldsymbol{x})$. The variance measures how uncertain $\hat{f}_D(\boldsymbol{x})$ is due to the randomness of the dataset D, for a given input $\boldsymbol{x}$. With simple manipulations, we arrive at the following decomposition of the risk of the estimator:

$$\mathbb{E}_{D,\boldsymbol{x},y}(y - \hat{f}_D(\boldsymbol{x}))^2 = \mathbb{E}_{\boldsymbol{x}}[\text{bias}^2(\hat{f}_D, \boldsymbol{x}) + \text{var}(\hat{f}_D, \boldsymbol{x})] + \mathbb{E}_{\boldsymbol{x},y}(y - \mathbb{E}(y \mid \boldsymbol{x}))^2\,, \tag{2.8}$$

where the rightmost term corresponds to the error committed by the best predictor $\mathbb{E}(y \mid \boldsymbol{x})$, and is thus *irreducible* and independent of the choice of estimator. The above decomposition suggests that a good learner is one which strikes both a low bias and low variance. A standard approach is to pick the empirical risk minimizer:

$$\hat{f}_D \in \arg\min_{\hat{f}\in\mathcal{F}} \frac{1}{n}\sum_{i=1}^{n}(y^{(i)} - \hat{f}(\boldsymbol{x}^{(i)}))^2\,, \tag{2.9}$$

where $\mathcal{F}$ is some hypothesis class of potential predictors. Richer hypothesis classes typically reduce the bias while increasing the variance. This suggests that one should select a model class $\mathcal{F}$ that strikes a good balance between both competing objectives. This is the bias-variance trade-off.

Not all losses allows for a bias-variance decomposition as the one shown above, but the terms "bias" and "variance" are often used informally in more diverse contexts to refer to the tension between the complexity of the model class and how difficult it is to estimate with finitely many samples. Another approach which applies to more general loss ℓ is to decompose the suboptimality gap in *estimation error* and *approximation error*. The *Bayes optimal error* is defined as $\ell^* := \inf_f \ell(\mathbb{P}, f)$ where the infimum is taken over all (measurable) functions f. We can then

arrive at the following decomposition

$$\ell(\mathbb{P}, \hat{f}_D) - \ell^* = \underbrace{\ell(\mathbb{P}, \hat{f}_D) - \inf_{f \in \mathcal{F}} \ell(\mathbb{P}, f)}_{\text{estimation error}} + \underbrace{\inf_{f \in \mathcal{F}} \ell(\mathbb{P}, f) - \ell^*}_{\text{approximation error}} . \tag{2.10}$$

The estimation error quantifies the error due to using a finite dataset to choose $\hat{f}$ as opposed to the actual distribution $\mathbb{P}$ (analogous to the variance term) while the approximation error quantifies the error due to restricting the search to predictors in $\mathcal{F}$ as opposed to all measurable predictors f (analogous to the bias term). The approximation error term suggests we should pick $\mathcal{F}$ as large as possible, but this generally leads to an increase in estimation error. The field of *statistical machine learning* is about providing upper bounds on the estimation error that hold with high probability (randomness comes from the dependence on D) for various function classes [Shalev-Shwartz and Ben-David, 2014, Mohri et al., 2018]. What comes out of these analyzes is that "smaller" function classes $\mathcal{F}$ lead to tighter bounds on the estimation error.

Although these types of analysis provide tight bounds for "underparameterized" methods such as *support vector machines* and *LASSO regression* [Mohri et al., 2018], they fail to explain the success of deep learning, which leverages large overparameterized neural networks capable of achieving zero training loss while still striking low test loss despite having very loose upper bounds on their estimation error [Zhang et al., 2017]. These observations have motivated more research which have yielded insights into this apparent mystery (see e.g. Belkin [2021]).

This thesis is not concerned with these important questions and, instead, focuses on identifiability which is somewhat orthogonal to the question of sample complexity. Of course, in practice, the finite-sample aspect of learning will impose itself, but identifiability at least provides some confidence that it is not completely impossible to approximately recover the ground-truth parameter $\boldsymbol{\theta}$.

2.2.3. Why study identifiability?

Finding interpretable structure in the data. Among all parameters $\boldsymbol{\eta}$ that fit the data-generating distribution $\mathbb{P}$ perfectly, some of them might be easier to interpret than others. In that case, studying the identifiability of a model class becomes crucial, since it provides a necessary condition for MLE to converge to an interpretable model. More precisely, if $\boldsymbol{\eta}^* \in H$ is considered interpretable with $\mathbb{P}_{\boldsymbol{\eta}^*} = \mathbb{P}$ and if the model $\{\mathbb{P}_{\boldsymbol{\eta}} \mid \boldsymbol{\eta} \in H\}$ is identifiable, then the only parameter that fits the ground-truth distribution exactly is the interpretable one, and the MLE estimator is going to converge to it in probability (under some regularity conditions).[3] We now provide a few examples

[3]Sometimes, we get identifiability of $\boldsymbol{\eta}$ only up to some equivalence class (see for example Section 2.5.1). In that case, defining what we mean by consistency is more subtle. For example, Datta and Chakrabarty [2023] shows that MLE for probabilistic principal component analysis, which is identifiable only up to rotations of its latent space, is consistent in a Euclidean quotient space.

of settings where the goal is to uncover interpretable structure from the data. Note that we expand on these examples further in Chapter 8 and provide three ways to interpret identifiability results.

- **Causal discovery:** The goal of causal discovery is to understand what are the causal relationships between various features given some data. These relationships are captured by a *causal graph* which consists of directed edges indicating direct causal relationships between features. In this setting, the causal graph is itself a parameter and understanding in which context it can be identified is of crucial importance. Indeed, if the graph is not identifiable, then there is no hope of estimating it from data. Causality and causal discovery are discussed in more details Sections 2.3 & 2.4.
- **Disentanglement:** In some context such as biology [Lopez et al., 2023], one is hoping to find an interpretable representation of some high-dimensional observation such as cell images or gene expression data. A common strategy is to fit a probabilistic model with low dimensional latent variables to the observations in hope that they will correspond to interpretable aspect of the data at hand. If the representation is not sufficiently identifiable, interpretability can be compromised. We discuss representation learning, disentanglement and the closely related *independent component analysis* problem in Section 2.5.
- **Clustering:** In some scientific settings, one might desire to find a reasonable partition of the data into different clusters corresponding to meaningful categories. Similarly to disentanglement, a standard approach to achieve this is to fit a probabilistic latent variable model where the latent variable corresponds to the identity of the cluster. If the clusters are not identifiable from the distribution, there is very little hope that the model is going to find meaningful clusters, as multiple reruns of the algorithm is likely to find different clusters each time.

Out-of-distribution generalization. Could it be that, among all models $\mathbb{P}_\eta$ that fit the data-generating distribution $\mathbb{P}$ exactly, some generalize better out of distribution? The term "out-of-distribution" is of course extremely vague as it refers to all distributions that were not seen during training. To make some progress, one has to be specific about which distributions one would like to generalize to. The decision theory framework allows us to do exactly that. To achieve this, we let the "state of nature" $\boldsymbol{\theta}$ capture all the distributions or tasks one should care about in a given context, and let the loss function $\ell(\boldsymbol{\theta}, \boldsymbol{\eta})$ measure how well the parameter $\boldsymbol{\eta}$ performs on each these tasks. In Chapter 8, we provide examples in causal discovery, disentanglement and clustering where identifiability is a key ingredient in obtaining out-of-distribution performance guarantees. Intuitively, all these examples consist in (i) noticing some structure in the data and the tasks one wishes to solve, (ii) show this structure can be recovered from data via an identifiability result, and (iii) leverage the learned structure to guarantee improved performance on a downstream task. This is idea is developed in much more depth in Chapter 8.

2.3. Causal graphical models

The language of probability allows us to describe the uncertainty state of a system. Here is a simple example: "What is the probability that a patient recovers from some illness *given that she received the drug* A?". Mathematically, this question can be formalized using conditional probabilities: $p(r = 1 \mid d = A)$ where r and d stand for "recovery" and "drug". If $p(r = 1 \mid d = A) > p(r = 1 \mid d = B)$, can we automatically conclude that drug A is more effective than drug B? Put differently, if you are forced to give the same drug to everyone, should you choose drug A to maximize the proportion of recovery? Although it might be tempting to say "yes", the correct answer is "not necessarily". We now provide an informal argument for why that is.

There is a third variable that we must consider: the health status of the patient prior to taking the drug, which we denote by h. Consider the following factorization of the joint distribution over all three variables

$$p(r, h, d) = p(h)p(d \mid h)p(r \mid h, d)\,. \tag{2.11}$$

This factorization implies no conditional independences, which means we could have chosen a different and equally valid factorization, like $p(d)p(r \mid d)p(h \mid r, d)$. However, the factorization of (2.11) is "nice" in the following sense: the conditional distributions appearing in it are such that the variables on the right of "|" have a direct causal effect on the variable on the left of "|". Indeed, as suggested by $p(r \mid h, d)$, the health status of the patient, h, and the drug she took, d, have a causal effect on r. Also, as suggested by $p(d \mid h)$, h has an effect on d since the drug is typically prescribed by a physician based on the health of the patient. For instance, drug B might be given only to patients that are seriously ill because of its greater cost.

Using the definition of conditional probability, we can show that

$$p(r = 1 \mid d = A) = \sum_h p(r = 1 \mid h, d = A)p(h \mid d = A)\,. \tag{2.12}$$

Now imagine a different world in which everyone must take drug $d = A$, regardless of their health status h. Would the model (2.11) still be a good description of this situation? No, because, in this model, d depends on h. A better model would be the following:

$$p^{do(d=A)}(r, h, d) = p(h)\mathbb{1}(d = A)p(r \mid h, d)\,, \tag{2.13}$$

where we replaced $p(d \mid h)$ by an indicator function $\mathbb{1}(d = A)$, which models the fact that the drug is chosen deterministically to be A. This new distribution can be marginalized over h and d to obtain

$$p^{do(d=A)}(r = 1) = \sum_h p(r = 1 \mid h, d = A)p(h)\,. \tag{2.14}$$

We now have two different quantities that are also similar: $p(r = 1 \mid d = A)$ is the probability of recovery given that the patient received drug A, while $p^{do(d=A)}(r = 1)$ is the probability of recovery in the world where everyone receives A.

When deciding which drug is more efficient in the sense that it would maximize recovery rate among the population, we should use model (2.13) since it properly captures the fact that everyone receives the same drug. It turns out that it is possible to have simultaneously

$$p(r = 1 \mid d = A) > p(r = 1 \mid d = B) \text{ and } p^{do(d=B)}(r = 1) > p^{do(d=A)}(r = 1)\,.$$

This is an instance of the *Simpson's paradox*, and it could occur for instance when drug B is the most effective drug but also more expensive, so that physicians prescribe it only for seriously ill patients. However, in a world where money was not an issue, everyone should receive this treatment to maximize the proportion of recovery.

The distribution $p^{do(d)}$ we just constructed is so important that it has a name: it is an *interventional distribution*. In contrast, p is called the *observational distribution*. We will see that the framework of causality generalizes these ideas. Roughly speaking, causality can be separated in two categories: *causal inference* and *causal structure learning*.

Causal inference deals with the problem of expressing queries that are *interventional* in nature, like $p^{do(d=A)}(r = 1)$, in terms of purely *observational* quantities, like $p(r = 1 \mid d = A)$. We already saw an example where this is possible. Indeed, (2.14) shows that $p^{do(d=A)}(r = 1)$ can be written in terms of factors that can be computed from the observational distribution p. The practical relevance of this is clear: it allows us to estimate the effects of interventions without actually performing them in the real world.

However, causal inference typically requires the knowledge of "what causes what". Indeed, in the example we just saw, we said that the factorization of (2.11) corresponded to the causal structure of the problem. Importantly, we made use of this causal factorization to compute the interventional distribution $p^{do(d)}$. A different causal factorization would have led to a different $p^{do(d)}$. This structure is captured by what is called a *causal graph*. In our example, the causal graph could be determined simply by common sense. In some situations, the causal graph can be uncovered only by an expert in the field of interest. There are also situations where the causal graph is simply unknown and must be discovered. *Causal structure learning*, a.k.a. *causal discovery*, is the problem of discovering a causal graph from observational and, potentially, interventional data. The first and second contributions of this thesis (Chapters 3 and 4) are mainly concerned with the causal structure learning problem, rather than causal inference.

The rest of this section is strongly inspired by the presentation of Peters et al. [2017].

2.3.1. Graph terminology

The causal framework is articulated via directed graphs, which make for a compact and visual tool to describe and reason about conditional independences and causal relationships. This section presents important graph terminology necessary to understand central notions presented later on.

A **directed graph** $\mathcal{G} = (V, E)$ consists of a **node set** $V = \{1, ..., d\}$ and an **edge set** $E \subseteq V^2$ containing the directed edges, i.e. $(i, j) \in E$ when there is a directed edge from i to j. When $(i, j) \in E$, we sometimes write $i \to j \in \mathcal{G}$ instead. A node i is a **parent** of j if $i \to j \in \mathcal{G}$ and a **child** of j if $j \to i \in \mathcal{G}$. We note the set of parents of j by $\pi_j^{\mathcal{G}}$. Three nodes i, j and k form an **immorality** in $\mathcal{G}$ when $i \to j \in \mathcal{G}$ and $j \leftarrow k \in \mathcal{G}$, but $i \to k \notin \mathcal{G}$ nor $i \leftarrow k \notin \mathcal{G}$. We say two graphs $\mathcal{G}_1 = (V, E_1)$ and $\mathcal{G}_2 = (V, E_2)$ **share skeleton** if for all $(i, j) \in V$, $(i, j) \in E_1$ or $(j, i) \in E_1$ if and only if $(i, j) \in E_2$ or $(j, i) \in E_2$. A **path** is a sequence of *distinct* nodes $i_1, ..., i_m$ such that $i_k \to i_{k+1}$ or $i_k \leftarrow i_{k+1}$ for all $k = 1, ..., m-1$. If $i_{k-1} \to i_k$ and $i_k \leftarrow i_{k+1}$ in a path, the node i_k is a **collider relative to the path**. If $i_k \to i_{k+1}$ for all k, we say there is a **directed path** for i_1 to i_m and i_1 is called an **ancestor** of i_m and i_m is called a **descendant** of i_1. A directed graph $\mathcal{G}$ is **acyclic** if there is no directed cycles, i.e. there is no pair (i, j) such that there is a directed path from i to j and a directed path from j to i. We then call $\mathcal{G}$ a **directed acyclic graph** (DAG).

2.3.2. Causal graphical models (CGM) and Interventions

We are now ready to present the formal definition of a causal graphical model (CGM).

Definition 2.4. *(Causal graphical model) A causal graphical model over random variables $\boldsymbol{x} : \Omega \to \mathbb{R}^d$ is a DAG $\mathcal{G}$ together with a collection of functions $f_j(\boldsymbol{x}_j|\boldsymbol{x}_{\pi_j^{\mathcal{G}}})$ such that*

$$\int f_j(\boldsymbol{x}_j \mid \boldsymbol{x}_{\pi_j^{\mathcal{G}}}) d\boldsymbol{x}_j = 1 \; \forall \boldsymbol{x}_{\pi_j^{\mathcal{G}}} \, . \tag{2.15}$$

These functions induce a distribution $\mathbb{P}_{\boldsymbol{x}}$ over $\boldsymbol{x}$ via the density function

$$p(\boldsymbol{x}) := \prod_{j=1}^{d} f_j(\boldsymbol{x}_j \mid \boldsymbol{x}_{\pi_j^{\mathcal{G}}}) \, . \tag{2.16}$$

This is referred to as the observational distribution. Given an interventional target $I \subseteq V$ and functions $\tilde{f}_j(\boldsymbol{x}_j \mid \boldsymbol{x}_{\pi_j^{\mathcal{G}}})$ for all $j \in I$ (also integrating to 1), a CGM induces an interventional distribution via the following expression:

$$p^{(I)}(\boldsymbol{x}) = \prod_{j \notin I} f_j(\boldsymbol{x}_j \mid \boldsymbol{x}_{\pi_j^{\mathcal{G}}}) \prod_{j \in I} \tilde{f}_j(\boldsymbol{x}_j \mid \boldsymbol{x}_{\pi_j^{\mathcal{G}}}) \, . \tag{2.17}$$

This is referred to as the interventional distribution entailed by I. When multiple interventions are observed, we regroup all the interventional targets into an interventional family $\mathcal{I} := (I_1, ..., I_K)$ and use the shorthand $p^{(k)} = p^{(I_k)}$ to refer to the kth interventional distribution.

The definition of intervention presented above captures the idea of sparse or localized change in a distribution. Each conditional can be thought of as a mechanism which can be manipulated or intervened upon. This interpretation is sometimes referred to as the "principle of independent mechanisms" [Peters et al., 2017, Section 2.1]. This definition generalizes the example we saw in Section 2.3. Indeed, it can be verified that $p^{do(D=A)}$ from (2.13) is an interventional distribution in the sense of Definition 2.4.

This definition already shows how the causal graph $\mathcal{G}$ imposes constraints or *invariances* on the distributions $p, p^{(1)}, p^{(2)}, ...p^{(K)}$. It should be clear that different graphs will result in different invariances.

2.3.3. Markov property and Markov equivalence

We first present the notion of d-separation in directed graphs. This notion is important because, in a causal graphical model, d-separations in the graph $\mathcal{G}$ imply analogous conditional independences in the observational distribution $\mathbb{P}_{\boldsymbol{x}}$. This provides a useful tool to read off conditional independence statements from the graph $\mathcal{G}$. See Definition 2.6 for more on this.

Definition 2.5. *(Pearl's d-separation; Pearl [1985, 1988]) In a DAG $\mathcal{G}$, a path $i_1, ..., i_m$ is **blocked by a set** S (with neither i_1 nor i_m in it) whenever there is a node i_k such that one of the following possibility holds:*

(1) $i_k \in S$ and

$$i_{k-1} \to i_k \to i_{k+1} \tag{2.18}$$

$$\text{or } i_{k-1} \leftarrow i_k \leftarrow i_{k+1} \tag{2.19}$$

$$\text{or } i_{k-1} \leftarrow i_k \to i_{k+1} \tag{2.20}$$

(2) neither i_k nor any of its descendant is in S and

$$i_{k-1} \to i_k \leftarrow i_{k+1}\,. \tag{2.21}$$

Furthermore, in a DAG $\mathcal{G}$, we say that two disjoints subsets of vertices A and B are d-separated by a third (also disjoint) subset S if every path between nodes in A and B is blocked by S. We then write $A \perp\!\!\!\perp_{\mathcal{G}} B \mid S$.

The Markov property relates the notion of d-separation to conditional independence statements in a distribution. This property is important since it is satisfied in causal graphical models.

Definition 2.6. *(Markov Property) Given a DAG $\mathcal{G}$, a distribution $\mathbb{P}_{\boldsymbol{x}}$ is said to satisfy*

*(1) the **global Markov property** with respect to the DAG $\mathcal{G}$ if*

$$A \perp\!\!\!\perp_{\mathcal{G}} B \mid C \implies \boldsymbol{x}_A \perp\!\!\!\perp_{\mathbb{P}_{\boldsymbol{x}}} \boldsymbol{x}_B \mid \boldsymbol{x}_C \tag{2.22}$$

for all disjoint node sets A, B and C (where $\perp\!\!\!\perp_{\mathbb{P}_{\boldsymbol{x}}}$ denotes conditional independence in $\mathbb{P}_{\boldsymbol{x}}$),

(2) *the **local Markov property** with respect to the DAG $\mathcal{G}$ if each variable is independent of its non-descendants given its parents,*

(3) *the **Markov factorization property** with respect to the DAG $\mathcal{G}$ if $\mathbb{P}_{\boldsymbol{x}}$ has a density p and*

$$p(\boldsymbol{x}) = \prod_{j=1}^{d} p_j(\boldsymbol{x}_j \mid \boldsymbol{x}_{\pi_j^{\mathcal{G}}}) \,, \tag{2.23}$$

where $\int p_j(\boldsymbol{x}_j \mid \boldsymbol{x}_{\pi_j^{\mathcal{G}}}) d\boldsymbol{x}_j = 1$.

It turns out that these three Markov properties are equivalent if $\mathbb{P}_{\boldsymbol{x}}$ has a density [Lauritzen, 1996, Theorem 3.27]. In that case, we say that a *distribution $\mathbb{P}_{\boldsymbol{x}}$ is Markov to $\mathcal{G}$* when it satisfies the equivalent Markov properties of Definition 2.6. It should be clear from the definitions that, in a CGM, the observational distribution $\mathbb{P}_{\boldsymbol{x}}$ is Markov to $\mathcal{G}$. These conditional independences can be thought of as constraints or invariances in a distribution, alluding again at the idea that graph imposes constraints on a distribution.

An important question for causal discovery and structure learning more generally is whether multiple graphs can entail the same set of conditional independences. The answer is "yes" and this fact is captured by the notion of Markov equivalence. This fact should be depressing to anyone willing to learn the causal graph from observations (i.e. people interested in structure learning, like us), since it suggests that recovering the graph from the distribution is impossible (at least without further assumptions and/or interventions).

Definition 2.7. *(Markov equivalence) We denote $\mathcal{M}(\mathcal{G})$ to be the set of all distributions that are Markov to $\mathcal{G}$. Two DAGs $\mathcal{G}_1$ and $\mathcal{G}_1$ are Markov equivalent if $\mathcal{M}(\mathcal{G}_1) = \mathcal{M}(\mathcal{G}_2)$. This is the case if and only if $\mathcal{G}_1$ and $\mathcal{G}_2$ contains the same d-separations. The set of all DAGs which are Markov equivalent to $\mathcal{G}$ is called the Markov equivalence class of $\mathcal{G}$, denoted by MEC$(\mathcal{G})$.*

Verma and Pearl [1990] showed a simple graphical characterization of equivalence which simplifies the verification of whether two DAGs are Markov equivalent.

Lemma 2.1. *(Graphical criteria for Markov equivalence; Verma and Pearl [1990]) Two DAGs $\mathcal{G}_1$ and $\mathcal{G}_2$ are Markov equivalent if and only if they share skeletons and have the same immoralities.*

The Markov property and the notion of Markov equivalence can be extended to interventional distributions. Given an *interventional family* $\mathcal{I} := (I_1, ..., I_K)$, where each I_k is an interventional target (Definition 2.4), the $\mathcal{I}$-Markov property can be defined as well as the notion of $\mathcal{I}$-Markov equivalence of graphs. Appendix A.1 of Chapter 4 contains a condensed presentation of these notions, as introduced by Yang et al. [2018].

2.4. Causal structure learning

We already saw that, when the causal graph is known, causal inference allows one to answer a wide variety of interventional queries *without having to actually perform these interventions in*

the real world. Examples of such queries include: “What will be the effect of passing a law that requires wearing a mask in public on the daily new cases of COVID-19?” or "What will be the impact of giving drug X on symptom Y?". In many cases, expertise in an area (or sometimes simple common sense) can tell us a lot about what the causal graph actually is [Pearl, 2009a, Chapter 5]. But in many other situations such as genomics [Dixit et al., 2016], the causal graph is unknown and must be inferred. This is where causal structure learning comes in.

Causal structure learning (or causal discovery) is the problem of learning a graph from observations. The observations are assumed to come from a causal graphical model with DAG $\mathcal{G}$. The observations can come from either the purely observational distribution or interventional distributions. In this section, we concentrate on the case where all observations come from the observational distribution, i.e. without any interventions. Causal structure learning from interventions will be covered in Chapter 4.

2.4.1. Structure identifiability

The learner is given n samples from the observational distribution $\mathbb{P}_{\boldsymbol{x}}$ of a CGM and wants to infer its corresponding causal graph $\mathcal{G}$. Given infinite data[4], is it possible to recover the ground truth graph $\mathcal{G}$? In general, it is impossible without further assumptions, as we already briefly mentioned in Section 2.3.3. Given infinite data, is it possible to recover even just the Markov equivalence class of $\mathcal{G}$? Again, no, unless we make further assumptions. To render the Markov equivalence class identifiable, it is sufficient to assume faithfulness.

Definition 2.8. *(Faithfulness) A distribution $\mathbb{P}_{\boldsymbol{x}}$ is faithful to $\mathcal{G}$ if for all disjoint sets $A, B, C \subseteq V$,*

$$\boldsymbol{x}_A \perp\!\!\!\perp_{\mathbb{P}_{\boldsymbol{x}}} \boldsymbol{x}_B | \boldsymbol{x}_C \implies A \perp\!\!\!\perp_{\mathcal{G}} B | C \,. \tag{2.24}$$

It should be noted that faithfulness is the converse of the global Markov property (Definition 2.6). See Peters et al. [2017, p.107] for an example of a causal model which violates faithfulness. This assumption is considered reasonable since constructing an unfaithful distribution requires careful “tuning” of the parameters. For instance for linear models, the set of unfaithful distributions has a Lebesgue measure of zero [Spirtes et al., 2000, Theorem 3.2].

Intuitively, this assumption ensures that conditional independences in the distribution are actually represented in the graph. This is useful for causal discovery since it allows one to extract information about $\mathcal{G}$ by looking at conditional independences in $\mathbb{P}_{\boldsymbol{x}}$. Faithfulness and the Markov property establishes a one-to-one correspondence between conditional independences in the distribution and d-separations in graph. This means we can infer the set of d-separations present in $\mathcal{G}$, but since many graphs have the same set of d-separations, we can only recover the Markov equivalence class of $\mathcal{G}$.

[4]We are referring to the hypothetical situations where the actual distribution $\mathbb{P}_{\boldsymbol{x}}$ is fully known.

One can improve graph identifiability further by making stronger assumptions about the nature of the data-generating CGM. The idea is to restrict the class of functions in which the ground truth $p_j(\boldsymbol{x}_j \mid \boldsymbol{x}_{\pi_j^{\mathcal{G}}})$ belongs. In Chapter 3, we leverage such a result which was initially introduced by Peters et al. [2014].

Having access to interventional data also improves identifiability substantially. Given some assumptions analogous to faithfulness (Definition 2.8), it can be shown that interventions allow to identify what is called the *$\mathcal{I}$-Markov equivalence class of $\mathcal{G}$* which is typically much smaller than the standard Markov equivalence class. Appendix A.1 of Chapter 4 define these notions as originally introduced by Yang et al. [2018]. Appendix A.2 goes further and provides an original identifiability results based on the maximization of a regularized maximum likelihood score.

2.4.2. Algorithms

Knowing that a graph (or an equivalence class) can be identified from the distribution suggests that we should be able to come up with algorithms to estimate it from observations sampled from $\mathbb{P}_{\boldsymbol{x}}$. Most algorithms fall into one of these categories: *independence-based* and *score-based* methods. We briefly present a few instances of these types of methods.

Independence-based methods run a sequence of (conditional) independence tests to discover which d-separations hold in the underlying ground truth DAG $\mathcal{G}$. The faithfulness assumption allows us to make the jump from (conditional) independence statements to d-separations in the graph. Any conditional independence tests can be used as long as it is flexible enough to capture the potentially nonlinear dependencies present in the distribution. For instance, a popular option is the Hilbert-Schmidt independence criterion (HSIC) [Gretton et al., 2007]. Algorithms for selecting which independence tests to run differ only by the order in which they perform the tests, which can sometimes have a drastic effect on the running time of the algorithm. Some algorithms like the IC algorithm [Pearl, 2009a] and the SGS algorithm [Spirtes et al., 2000] test for all possible conditional independences of pairs given a subset while the PC algorithm [Spirtes et al., 2000] does not have to run all tests to be exhaustive [Peters et al., 2017].

Score-based methods cast the problem of learning a DAG as an optimization problem of the form

$$\max_{\hat{\mathcal{G}} \in \text{DAG}} \mathcal{S}(\mathcal{D}, \hat{\mathcal{G}}) \,, \tag{2.25}$$

where $\mathcal{S}(\mathcal{D}, \hat{\mathcal{G}})$ is a score function to maximize and $\mathcal{D}$ is a dataset. A typical choice of score is the Bayesian Information Criterion (BIC) which is given by

$$\mathcal{S}(\mathcal{D}, \hat{\mathcal{G}}) = \max_{\hat{\boldsymbol{\theta}}} \log p(\mathcal{D} \mid \hat{\boldsymbol{\theta}}, \hat{\mathcal{G}}) - \frac{\log n}{2} |\hat{\mathcal{G}}| \,, \tag{2.26}$$

where $\log p(\mathcal{D} \mid \hat{\boldsymbol{\theta}}, \hat{\mathcal{G}})$ is the log-likelihood function given some model (e.g. Gaussian linear model) and $|\hat{\mathcal{G}}|$ is the number of parameters to estimate with graph $\hat{\mathcal{G}}$. The number of parameters tends to grow with the size of the graph, for instance if we are considering a Gaussian linear model, each edge in the graph requires its own scalar parameter, i.e. $|\hat{\mathcal{G}}|$ is the number of edges. This term is encouraging sparsity.

The discrete optimization problem in (2.25) presents a serious computational challenge since the space of DAGs grows super-exponentially in the number of nodes and the set of feasible directed graphs is rather involved. When we are only looking for a Markov equivalence class, the problem can be modified to search directly in the space of Markov equivalence classes, which reduces the search space substantially. This is the approach proposed by the Greedy Equivalence Search algorithm (GES) [Chickering, 2003]. This approach has been extended to support interventional data as well [Hauser and Bühlmann, 2012].

In some situations, recovering the causal graph only up to its Markov equivalence class is unsatisfactory. To obtain exact identifiability of the graph, we might want to restrict the model class. The causal additive model (CAM) [Bühlmann et al., 2014] employ this approach and search the space of DAG greedily. To allow for large graphs (50 nodes or more), CAM relies on a *preliminary neighborhood selection* phase which blacklists some edges in the graph via statistical tests before starting the search, thus reducing the search space.

The continuous-constrained optimization approach. The algorithms presented so far all embraced the discrete nature of the problem head on by performing some form of greedy optimization. In Section 3.2.3, we present a formulation of the structure learning problem proposed by Zheng et al. [2018] which recast this combinatorial problem as a continuous-constrained optimization problem. This formulation allows us to explore drastically different learning algorithms based on numerical optimization. The challenges of discrete optimization are replaced by those of nonconvex-constrained optimization. In its first iteration, the approach assumed a Gaussian linear model and could not make use of interventional data. The contribution of Chapter 3 shows how this formulation can be extended to support nonlinear relationships with neural networks while the contribution of Chapter 4 shows this approach can support various types of interventional data and can be extended to work with powerful density estimators such as normalizing flows [Rezende and Mohamed, 2015].

2.5. Representation learning

The success of deep learning [Goodfellow et al., 2016] is attributable in part to the idea of *learning* the features that are useful for a given task instead *engineering* them. This strategy comes with all sorts of computational and statistical challenges, but these were largely overcame, as was exemplified by the groundbreaking success of the *AlexNet* architecture at the *ImageNet Large Scale*

Visual Recognition Challenge [Krizhevsky et al., 2012]. The initial successes of deep learning were initially limited to the *supervised learning* regime, where a very large dataset of input/label pairs (x, y) is used to train a neural network to predict the label y from the input x. The intermediate representations learned in this way tend to be tightly tailored to the task on which it was trained and has limited utility when used for other tasks.

Representation learning can be understood as going one step further, i.e. learning a representation that is suitable for many tasks, sometimes even without knowing what these will be. A pre-trained representation can be used in a downstream task for instance by training a linear classifier to predict the labels from the representation using a relatively small dataset of labelled samples from the new task. One can also decide to either *fine-tune* or *freeze* the pre-trained representation when doing so. A plethora of strategies for representation learning have been contributed to the literature. Murphy [2023] classifies these approaches into the following categories:

(i) *Supervised representation learning and transfer* corresponds to reusing the representation learned via supervised learning, sometimes in a multi-tasks setting.

(ii) *Latent variable models* are probabilistic models of the form $p(\boldsymbol{x} \mid \boldsymbol{z})p(\boldsymbol{z})$ where $\boldsymbol{z}$ is hidden and taken to be the representation of the observation $\boldsymbol{x}$.

(iii) *Autoregressive models* are probabilistic models of the form $\prod_{i=1}^{d_x} p(\boldsymbol{x}_i|\boldsymbol{x}_{<i})$, which include neural architecture such as Transformers, where the representation is taken to be the output of an intermediate hidden layer.

(iv) *Autoencoders* consists in minimizing a loss of the form $\mathbb{E}\|\boldsymbol{x} - \text{Dec}(\text{Enc}(\boldsymbol{x}))\|^2$ where the representation of $\boldsymbol{x}$ is taken to be $\text{Enc}(\boldsymbol{x})$, which has a much lower dimensionality than $\boldsymbol{x}$.

(v) *Self-supervision* refers to various approaches in which a model is trained to solve a "synthetic" task such as denoising an input or classifying which transformation an input received.

(vi) *Contrastive learning* can be considered as a special case of the above in which the synthetic task consists in classifying which pairs of inputs are *positive*, and which are *negative*. For example, a positive pair could be an image together with its transformed version (e.g. rotated or cropped) while a negative pair could be two completely unrelated images. When image-caption pairs are available for training, positive pairs would be correctly matched while the negative pairs would be incorrectly matched. The latter is the strategy employed to train the now popular CLIP model [Radford et al., 2021]. In these models, the classification logit for a pair $(\boldsymbol{x}, \boldsymbol{x}')$ is typically given by $\boldsymbol{f}(\boldsymbol{x})^\top \boldsymbol{g}(\boldsymbol{x}')$ where $\boldsymbol{f}$ and $\boldsymbol{g}$ are neural networks outputting the representations of $\boldsymbol{x}$ and $\boldsymbol{x}'$.

Chapter 5 will be mainly concerned with latent variable models, Chapter 6 with representation learning via multi-task learning, and Chapter 7 with autoencoders. All three of these contributions explore the *identifiability* of the representations learned in each of these settings. To explain what we mean by representation identifiability, we will focus on the case of latent variable models.

2.5.1. Disentanglement & identifiability in latent variable models

In some cases, the primary goal of representation learning is to *discover interesting structure in the data*, as opposed to learning a representation that is suitable for multiple downstream tasks. For instance, we might want to extract an interpretable representation of brain images that exhibits a *low-dimensional* and *interpretable* description of some neural processes [Monti and Hyvärinen, 2018]. In fact, many biological systems are still not fully understood and could benefit from machine learning similarly [Lopez et al., 2023]. This idea is sometimes referred to as *disentanglement* [Bengio et al., 2013, Higgins et al., 2017, Locatello et al., 2020a], where the goal is to learn a representation in which each coordinate corresponds to a so-called "natural factor of variation" of the dataset. A canonical example would be to learn a representation of images in which the positions of the objects, their colors and orientations are represented as individual coordinates. Various strategies to achieve this have been contributed to the literature, with many of them based on heuristics. Locatello et al. [2020a] brought to light the identifiability problem in disentangled representation learning and motivated multiple works to pursue more principled strategies backed by identifiability guarantees (see Sections 5.7, 6.4 & 7.2 for more exhaustive literature reviews).

We recall the definition of identifiability covered in Section 2.2.1. Given a parameterized family of distributions $\{\mathbb{P}_{\boldsymbol{\theta}} \mid \boldsymbol{\theta} \in \Theta\}$, we say that $\boldsymbol{\theta}$ is identifiable from $\mathbb{P}_{\boldsymbol{\theta}}$ if the following holds:

$$\forall \boldsymbol{\theta}, \hat{\boldsymbol{\theta}} \in \Theta,\ \mathbb{P}_{\boldsymbol{\theta}} = \mathbb{P}_{\hat{\boldsymbol{\theta}}} \implies \boldsymbol{\theta} = \hat{\boldsymbol{\theta}}\,, \tag{2.27}$$

or, in other words, the map $\boldsymbol{\theta} \mapsto \mathbb{P}_{\boldsymbol{\theta}}$ is injective. This means that, given the distribution $\mathbb{P}_{\boldsymbol{\theta}}$, we can determine unambiguously which parameter $\boldsymbol{\theta}$ gave rise to it. In a learning setting where we assume that the ground-truth data distribution is $\mathbb{P}_{\boldsymbol{\theta}}$ and that we managed to learn it exactly such that $\mathbb{P}_{\hat{\boldsymbol{\theta}}} = \mathbb{P}_{\boldsymbol{\theta}}$ (for instance with maximum likelihood estimation in the infinite data regime), identifiability allows us to conclude that $\hat{\boldsymbol{\theta}} = \boldsymbol{\theta}$. Typically, when the model $\mathbb{P}_{\boldsymbol{\theta}}$ has many degrees of freedom, identifying $\boldsymbol{\theta}$ exactly might be impossible, so instead we aim to identify $\boldsymbol{\theta}$ *up to some equivalence relation* $\sim$. This yield the following definition:

$$\forall \boldsymbol{\theta}, \hat{\boldsymbol{\theta}} \in \Theta,\ \mathbb{P}_{\boldsymbol{\theta}} = \mathbb{P}_{\hat{\boldsymbol{\theta}}} \implies \boldsymbol{\theta} \sim \hat{\boldsymbol{\theta}}\,. \tag{2.28}$$

We will see concrete examples of equivalence relations in a few paragraphs.

In many settings, we only care about finding a model $\mathbb{P}_{\hat{\boldsymbol{\theta}}}$ that describes faithfully the data and have no interest in whether or not we found the "right" (equivalence class of) $\boldsymbol{\theta}$. In some sense, all the $\hat{\boldsymbol{\theta}}$ such that $\mathbb{P}_{\hat{\boldsymbol{\theta}}} = \mathbb{P}_{\boldsymbol{\theta}}$ are equally valid in that they describe the data distribution perfectly. That being said, it is possible that, among all parameters $\hat{\boldsymbol{\theta}}$ that fits the data perfectly, some of them are more *interpretable* than others. This is where identifiability becomes interesting. Chapter 8 provides further motivations to study identifiability, especially regarding downstream performance.

Let me make the discussion more concrete by considering a latent variable model of the form $\boldsymbol{x} = \boldsymbol{f}(\boldsymbol{z})$ where $\boldsymbol{z} \sim \mathbb{P}_{\boldsymbol{z}}$. In that case, we set our parameter to be $\boldsymbol{\theta} := (\boldsymbol{f}, \mathbb{P}_{\boldsymbol{z}})$ and our parameter space to be $\Theta := \mathcal{F} \times \mathcal{P}$, where $\mathcal{F}$ is some class of functions (assume bijective for now) and $\mathcal{P}$ is some class of distributions $\mathbb{P}_{\boldsymbol{z}}$. This model induces a distribution over $\boldsymbol{x}$ given by $\mathbb{P}_{\boldsymbol{\theta}} = \mathbb{P}_{\boldsymbol{z}} \circ \boldsymbol{f}^{-1}$, i.e. the pushforward of $\mathbb{P}_{\boldsymbol{z}}$ under $\boldsymbol{f}$. Assume that the specific model $\boldsymbol{\theta} = (\boldsymbol{f}, \mathbb{P}_{\boldsymbol{z}})$ is interpretable, in the sense that each coordinate of $\boldsymbol{z}$ corresponds to natural factors of variations in the data, such as object positions, colors and orientations. Note that Chapters 5, 6 & 7 expand further on what it could mean to be *interpretable* in different settings. Now, notice that we can very easily find another model $\hat{\boldsymbol{\theta}} = (\hat{\boldsymbol{f}}, \hat{\mathbb{P}}_{\boldsymbol{z}})$ that (i) yields exactly the same distribution over $\boldsymbol{x}$, and (ii) that does not have necessarily an interpretable representation. For any invertible transformation $\boldsymbol{v}$, we can choose

$$\hat{\boldsymbol{f}} := \boldsymbol{f} \circ \boldsymbol{v}^{-1} \text{ and } \hat{\mathbb{P}}_{\boldsymbol{z}} := \mathbb{P}_{\boldsymbol{z}} \circ \boldsymbol{v}^{-1} \text{ (the latter is the distribution of } \boldsymbol{v}(\boldsymbol{z}) \text{ when } \boldsymbol{z} \sim \mathbb{P}_{\boldsymbol{z}}) \,,$$

where $\boldsymbol{v}$ is an arbitrary bijective transformation. Intuitively, we are simply applying an invertible transformation $\boldsymbol{v}$ on $\boldsymbol{z}$ and undoing it at the input of $\boldsymbol{f}$. Of course, this is not changing the distribution over $\boldsymbol{x}$ since, formally, we have

$$\mathbb{P}_{\hat{\boldsymbol{\theta}}} = \hat{\mathbb{P}}_{\boldsymbol{z}} \circ \hat{\boldsymbol{f}}^{-1} = \mathbb{P}_{\boldsymbol{z}} \circ \boldsymbol{v}^{-1} \circ \boldsymbol{v} \circ \boldsymbol{f}^{-1} = \mathbb{P}_{\boldsymbol{z}} \circ \boldsymbol{f}^{-1} = \mathbb{P}_{\boldsymbol{\theta}} \,.$$

And, importantly, the representation of $\boldsymbol{x}$ in both models, $\boldsymbol{f}^{-1}(\boldsymbol{x})$ and $\hat{\boldsymbol{f}}^{-1}(\boldsymbol{x})$, could be drastically different. Indeed, we have that $\hat{\boldsymbol{f}}^{-1}(\boldsymbol{x}) = \boldsymbol{v} \circ \boldsymbol{f}^{-1}(\boldsymbol{x})$, i.e. both representations are related by $\boldsymbol{v}$. This means that $\hat{\boldsymbol{f}}^{-1}(\boldsymbol{x})$ might not be interpretable, even if $(\hat{\boldsymbol{f}}, \hat{\mathbb{P}}_{\boldsymbol{z}})$ matches the data distribution exactly. This is problematic, since it shows that simply finding a parameter $\hat{\boldsymbol{\theta}}$ that perfectly fits the data distribution is not enough to guarantee that the learned model is interpretable.

We would then like to restrict the classes $\mathcal{F}$ and/or $\mathcal{P}$ such that the only transformations $\boldsymbol{v}$ that keep $\boldsymbol{f} \circ \boldsymbol{v}^{-1}$ in $\mathcal{F}$ and $\mathbb{P}_{\boldsymbol{z}} \circ \boldsymbol{v}^{-1}$ in $\mathcal{P}$ are "trivial indeterminacies". In many settings, we tolerate element-wise transformations and permutations of the coordinates. This suggests a weaker notion of identifiability tailored to disentanglement:

$$\forall (\boldsymbol{f}, \mathbb{P}_{\boldsymbol{z}}), (\hat{\boldsymbol{f}}, \hat{\mathbb{P}}_{\boldsymbol{z}}) \in \mathcal{F} \times \mathcal{P}, \mathbb{P}_{(\boldsymbol{f}, \mathbb{P}_{\boldsymbol{z}})} = \mathbb{P}_{(\hat{\boldsymbol{f}}, \hat{\mathbb{P}}_{\boldsymbol{z}})} \implies \boldsymbol{f} = \hat{\boldsymbol{f}} \circ \boldsymbol{d} \circ \boldsymbol{P} \,, \tag{2.29}$$

where $\boldsymbol{d}$ is some element-wise transformation and $\boldsymbol{P}$ is some permutation. This is weaker than (2.27) because we do not require that $\boldsymbol{f} = \hat{\boldsymbol{f}}$, but only that $\boldsymbol{f}$ and $\hat{\boldsymbol{f}}$ are related by a permutation $\boldsymbol{P}$ and an element-wise transformation $\boldsymbol{d}$.

In many works on identifiable representation learning, there is an asymmetry between the assumptions made on the ground-truth model and the learned model. Instead of (2.29), these results typically show

$$\forall (\boldsymbol{f}, \mathbb{P}_{\boldsymbol{z}}) \in \mathcal{F} \times \mathcal{P}, (\hat{\boldsymbol{f}}, \hat{\mathbb{P}}_{\boldsymbol{z}}) \in \hat{\mathcal{F}} \times \hat{\mathcal{P}}, \mathbb{P}_{(\boldsymbol{f}, \mathbb{P}_{\boldsymbol{z}})} = \mathbb{P}_{(\hat{\boldsymbol{f}}, \hat{\mathbb{P}}_{\boldsymbol{z}})} \implies \boldsymbol{f} = \hat{\boldsymbol{f}} \circ \boldsymbol{d} \circ \boldsymbol{P} \,, \tag{2.30}$$

where $\mathcal{F} \times \mathcal{P} \subseteq \hat{\mathcal{F}} \times \hat{\mathcal{P}}$, i.e. the assumptions on the ground-truth are stronger than on the learned model. This latter point will be illustrated in Sections 2.5.2 & 2.5.3. These results are applied to a learning scenario as follows: we assume that the data generating process, or ground-truth, is some unknown model $(\boldsymbol{f}, \mathbb{P}_{\boldsymbol{z}}) \in \mathcal{F} \times \mathcal{P}$. Then, we search in $\hat{\mathcal{F}} \times \hat{\mathcal{P}}$ for a model $(\hat{\boldsymbol{f}}, \hat{\mathbb{P}}_{\boldsymbol{z}})$ that fits the data distribution, i.e. $\mathbb{P}_{(\hat{\boldsymbol{f}}, \hat{\mathbb{P}}_{\boldsymbol{z}})} = \mathbb{P}_{(\boldsymbol{f}, \mathbb{P}_{\boldsymbol{z}})}$.[5] Then, (2.30) guarantees that the function $\hat{\boldsymbol{f}}$ we found is the same as the ground-truth $\boldsymbol{f}$, up to permutation and element-wise rescaling. Note that we could also just search over $\mathcal{F} \times \mathcal{P}$ to fit the ground-truth distribution, but in practice, $\hat{\mathcal{F}} \times \hat{\mathcal{P}}$ is typically much easier to optimize over, and results of the form (2.30) guarantee that this is in fact enough.

The discussion so far has been fairly abstract. In Section 2.5.2 & 2.5.3, we will see how linear independent component analysis (ICA) fits nicely into this framework.

A lot of the current research in this area, including this thesis, boils down to finding expressive hypothesis classes $\hat{\mathcal{F}} \times \hat{\mathcal{P}}$ that remain identifiable in the sense of (2.30). The strategy employed in Chapter 5 consists in restricting the distribution over $\boldsymbol{z}$ to have sparse dependencies, either with an observed auxiliary variable or a past latent vector (if the data present temporal dependencies). Chapters 6 & 7 do not fit exactly in the latent variable model setting describe above, but they are similar in spirit. Chapter 6 considers a multi-task setting in which a common representation is used across tasks. The identifiability up to permutation and element-wise rescaling is guaranteed by the fact that each task requires only a sparse subset of the features to be solved. Chapter 7 considers an autoencoder approach in which identifiability is ensured by restricting $\hat{\mathcal{F}}$ to be the set of additive functions. We note that all five contributions of this thesis will be unified further under the umbrella of statistical decision theory in Chapter 8.

2.5.2. Independent component analysis

Independent component analysis (ICA) [Hyvärinen et al., 2001] consists in finding a linear transformation $\boldsymbol{L}$ of the data $\boldsymbol{x}$ such that the transformed data $\boldsymbol{Lx}$ has mutually independent components. Many principles have been proposed to achieve this, like maximizing non-Gaussianity, minimizing mutual information or fitting a likelihood model with independent latent factors. The following discussion is limited to the identifiability of ICA as a latent variable model. Details about various algorithms for ICA can be found in Hyvärinen et al. [2001].

In the language established in the previous section, ICA assumes that

$$\begin{aligned} \hat{\mathcal{F}} &:= \{\text{linear maps from } \mathbb{R}^{d_z} \text{ to } \mathbb{R}^{d_x}\} \\ \hat{\mathcal{P}} &:= \{\mathbb{P}_{\boldsymbol{z}} \text{ such that the factors } \boldsymbol{z}_i \text{ are mutually independent}\}. \end{aligned} \tag{2.31}$$

[5] In practice, this could be achieved approximately by doing maximum likelihood estimation (Section 2.2.1).

Note that throughout, we assume $d_z \leq d_x$. Furthermore, the standard result assumes that the ground-truth model $(\boldsymbol{f}, \mathbb{P}_{\boldsymbol{z}})$ belongs to

$$\mathcal{F} := \hat{\mathcal{F}} \cap \{\text{injective maps}\} \tag{2.32}$$
$$\mathcal{P} := \hat{\mathcal{P}} \cap \{\mathbb{P}_{\boldsymbol{z}} \text{ with at most one Gaussian marginal and no deterministic marginals}\},$$

so that $\mathcal{F} \times \mathcal{P} \subseteq \hat{\mathcal{F}} \times \hat{\mathcal{P}}$, as discussed in Section 2.5.1.

The goal of this section will be to established identifiability of linear ICA, in the sense of (2.30). Formally, we want to show the following.

Corollary 2.1. *Let $(\boldsymbol{A}, \mathbb{P}_{\boldsymbol{z}}) \in \mathcal{F} \times \mathcal{P}$ and $(\hat{\boldsymbol{A}}, \hat{\mathbb{P}}_{\boldsymbol{z}}) \in \hat{\mathcal{F}} \times \hat{\mathcal{P}}$ where $\mathcal{F}, \hat{\mathcal{F}}, \mathcal{P}, \hat{\mathcal{P}}$ are defined in (2.31) and (2.32). Then, whenever $\mathbb{P}_{(\hat{\boldsymbol{A}}, \hat{\mathbb{P}}_{\boldsymbol{z}})} = \mathbb{P}_{(\boldsymbol{A}, \mathbb{P}_{\boldsymbol{z}})}$, we have $\boldsymbol{A} = \hat{\boldsymbol{A}}\boldsymbol{D}\boldsymbol{P}$ where $\boldsymbol{P}$ is a permutation matrix and $\boldsymbol{D}$ is an invertible diagonal matrix.*

The proof of this result, which will be presented at the end of this section, relies on a few intermediary results. We start with the following lemma in which we use "$\overset{d}{=}$" to denote equality in distribution.

Lemma 2.2. *Let $(\boldsymbol{A}, \mathbb{P}_{\boldsymbol{z}}) \in \mathcal{F} \times \mathcal{P}$ and $(\hat{\boldsymbol{A}}, \hat{\mathbb{P}}_{\boldsymbol{z}}) \in \hat{\mathcal{F}} \times \hat{\mathcal{P}}$ where $\mathcal{F}, \hat{\mathcal{F}}, \mathcal{P}, \hat{\mathcal{P}}$ are defined in (2.31) and (2.32). Then, whenever $\mathbb{P}_{(\hat{\boldsymbol{A}}, \hat{\mathbb{P}}_{\boldsymbol{z}})} = \mathbb{P}_{(\boldsymbol{A}, \mathbb{P}_{\boldsymbol{z}})}$, we have that (i) Range$(\hat{\boldsymbol{A}})$ = Range$(\boldsymbol{A})$, and (ii) $\hat{\boldsymbol{z}} \overset{d}{=} \hat{\boldsymbol{A}}^{\dagger}\boldsymbol{A}\boldsymbol{z}$ where $\hat{\boldsymbol{z}} \sim \hat{\mathbb{P}}_{\boldsymbol{z}}$, $\boldsymbol{z} \sim \mathbb{P}_{\boldsymbol{z}}$ and $\hat{\boldsymbol{A}}^{\dagger}$ is the pseudo-inverse of $\hat{\boldsymbol{A}}$.*

Proof We start with $\mathbb{P}_{(\hat{\boldsymbol{A}}, \hat{\mathbb{P}}_{\boldsymbol{z}})} = \mathbb{P}_{(\boldsymbol{A}, \mathbb{P}_{\boldsymbol{z}})}$, which is equivalent to $\hat{\boldsymbol{A}}\hat{\boldsymbol{z}} \overset{d}{=} \boldsymbol{A}\boldsymbol{z}$. This implies that $\text{supp}(\hat{\boldsymbol{A}}\hat{\boldsymbol{z}}) = \text{supp}(\boldsymbol{A}\boldsymbol{z})$. This further implies that $\hat{\boldsymbol{A}}\text{supp}(\hat{\boldsymbol{z}}) = \boldsymbol{A}\text{supp}(\boldsymbol{z})$.[6] Note that, because $\boldsymbol{z}$ has independent components, we have that $\text{supp}(\boldsymbol{z}) = \prod_{i=1}^{d_z} \text{supp}(\boldsymbol{z}_i)$. Furthermore, because no component $\boldsymbol{z}_i$ is deterministic, we have that, for all i, there exists two distinct values $\alpha_i^0, \alpha_i^1 \in \text{supp}(\boldsymbol{z}_i)$. This means that $\text{supp}(\boldsymbol{z})$ contains $\{\alpha_1^0, \alpha_1^1\} \times \cdots \times \{\alpha_{d_z}^0, \alpha_{d_z}^1\}$, which can be thought of as the vertices of a d_z-dimensional hyperrectangle. This implies that $\text{supp}(\boldsymbol{z})$ must contain a basis of $\mathbb{R}^{d_z}$, which we denote by $\boldsymbol{z}_{(1)}, \ldots, \boldsymbol{z}_{(d_z)} \in \text{supp}(\boldsymbol{z})$. We can collect these vectors into matrices $\boldsymbol{Z} := [\boldsymbol{z}_{(1)} \cdots \boldsymbol{z}_{(d_z)}]$ and, since $\hat{\boldsymbol{A}}\text{supp}(\hat{\boldsymbol{z}}) = \boldsymbol{A}\text{supp}(\boldsymbol{z})$, we know there exists $\hat{\boldsymbol{Z}} := [\hat{\boldsymbol{z}}_{(1)} \cdots \hat{\boldsymbol{z}}_{(d_z)}]$ such that $\boldsymbol{A}\boldsymbol{Z} = \hat{\boldsymbol{A}}\hat{\boldsymbol{Z}}$. We have that $\boldsymbol{A}$ is full column-rank by hypothesis and $\boldsymbol{Z}$ is invertible since its columns are linearly independent. This means $\boldsymbol{A}\boldsymbol{Z}$ has full column-rank, and so does $\hat{\boldsymbol{A}}\hat{\boldsymbol{Z}}$. This implies that $\hat{\boldsymbol{A}}$ must also have full column-rank and $\hat{\boldsymbol{Z}}$ must be invertible. This means that $\hat{\boldsymbol{A}}$ and $\boldsymbol{A}$ must have the same range (same image). Since $\hat{\boldsymbol{A}}$ has full-column rank, its pseudo-inverse, $\hat{\boldsymbol{A}}^{\dagger}$, is a left-inverse for $\hat{\boldsymbol{A}}$, i.e. $\hat{\boldsymbol{A}}^{\dagger}\hat{\boldsymbol{A}} = \boldsymbol{I}$. We can thus write $\hat{\boldsymbol{z}} \overset{d}{=} \hat{\boldsymbol{A}}^{\dagger}\boldsymbol{A}\boldsymbol{z}$. ■

The identifiability of linear ICA relies on the *Darmois-Skitovich theorem*, which we state without proof. For a recent treatment of these classical results, including proofs, see Pavan and Miranda [2018].

[6] We have that $\text{supp}(\boldsymbol{A}\boldsymbol{z}) = \boldsymbol{A}\text{supp}(\boldsymbol{z})$ by Lemma 5.6 in the Appendix of Chapter 5 combined with the fact that finite dimensional linear subspaces are closed.

Theorem 2.1 (Darmois-Skitovich, Darmois [1953], Skitivic [1953]). *Let $x_j, j = 1, \dots, n$ with $n \geq 2$ be mutually independent random variables and let α_j, β_j be constants. Let $y_1 := \sum_{j=1}^n \alpha_j x_j$ and $y_2 := \sum_{j=1}^n \beta_j x_j$ be two independent random variables. Then, whenever $\alpha_j \beta_j \neq 0$, the variable x_j is either constant or Gaussian.*

The following presents the crux of the work and makes use of the Darmois-Skitovich theorem.

Theorem 2.2 (Identifiability of linear ICA, Comon [1992]). *Suppose that $\boldsymbol{z}$ is a d_z-dimensional random vector ($d_z \geq 2$) of mutually independent and non-deterministic random variables in which at most one component is Gaussian (in other words, the distribution of $\boldsymbol{z}$ is in $\mathcal{P}$). Let $\boldsymbol{V} \in \mathbb{R}^{d_z \times d_z}$ be an invertible real matrix and let $\boldsymbol{y} := \boldsymbol{V}\boldsymbol{z}$. If the components of $\boldsymbol{y}$ are mutually independent, then $\boldsymbol{V} = \boldsymbol{D}\boldsymbol{P}$ for some invertible diagonal matrix $\boldsymbol{D}$ and permutation matrix $\boldsymbol{P}$.*

Proof The matrix $\mathrm{cov}(\boldsymbol{z})$ is diagonal (by independence) and invertible (all $\boldsymbol{z}_i$ are non-deterministic and thus have positive variance). Furthermore, the covariance matrix of $\boldsymbol{y}$ is diagonal (independence) and has the form $\mathrm{cov}(\boldsymbol{y}) = \boldsymbol{V}\mathrm{cov}(\boldsymbol{z})\boldsymbol{V}^\top$. Since $\boldsymbol{V}$ and $\mathrm{cov}(\boldsymbol{z})$ are invertible, so is $\mathrm{cov}(\boldsymbol{y})$. We can thus write

$$\boldsymbol{I} = \mathrm{cov}(\boldsymbol{y})^{-\frac{1}{2}}\boldsymbol{V}\mathrm{cov}(\boldsymbol{z})^{\frac{1}{2}}\mathrm{cov}(\boldsymbol{z})^{\frac{1}{2}}\boldsymbol{V}^\top\mathrm{cov}(\boldsymbol{y})^{-\frac{1}{2}} \tag{2.33}$$

$$= (\mathrm{cov}(\boldsymbol{y})^{-\frac{1}{2}}\boldsymbol{V}\mathrm{cov}(\boldsymbol{z})^{\frac{1}{2}})(\mathrm{cov}(\boldsymbol{y})^{-\frac{1}{2}}\boldsymbol{V}\mathrm{cov}(\boldsymbol{z})^{\frac{1}{2}})^\top \tag{2.34}$$

$$= \boldsymbol{M}\boldsymbol{M}^\top, \tag{2.35}$$

where we defined $\boldsymbol{M} := \mathrm{cov}(\boldsymbol{y})^{-\frac{1}{2}}\boldsymbol{V}\mathrm{cov}(\boldsymbol{z})^{\frac{1}{2}}$ and showed it is orthogonal. We thus have that $\boldsymbol{V} = \mathrm{cov}(\boldsymbol{y})^{\frac{1}{2}}\boldsymbol{M}\mathrm{cov}(\boldsymbol{z})^{-\frac{1}{2}}$. One can rewrite $\boldsymbol{y} = \boldsymbol{V}\boldsymbol{z}$ as

$$\bar{\boldsymbol{y}} = \boldsymbol{M}\bar{\boldsymbol{z}}, \tag{2.36}$$

where $\bar{\boldsymbol{y}} := \mathrm{cov}(\boldsymbol{y})^{-\frac{1}{2}}\boldsymbol{y}$ and $\bar{\boldsymbol{z}} := \mathrm{cov}(\boldsymbol{z})^{-\frac{1}{2}}\boldsymbol{z}$. Of course, $\bar{\boldsymbol{y}}$ and $\bar{\boldsymbol{z}}$ still have independent components (because $\mathrm{cov}(\boldsymbol{y})^{-\frac{1}{2}}$ and $\mathrm{cov}(\boldsymbol{z})^{-\frac{1}{2}}$ are diagonal) and $\bar{\boldsymbol{z}}$ has still at most one Gaussian component (for $\alpha \in \mathbb{R} \setminus \{0\}$, we have x Gaussian iff αx Gaussian) and none of its components are deterministic ($\mathrm{cov}(\boldsymbol{z})^{-\frac{1}{2}}$ has no zero on its diagonal). Notice that $\bar{\boldsymbol{y}}_1 = \sum_{j=1}^n \boldsymbol{M}_{1,j}\bar{\boldsymbol{z}}_j$ and $\bar{\boldsymbol{y}}_2 = \sum_{j=1}^n \boldsymbol{M}_{2,j}\bar{\boldsymbol{z}}_j$ are independent. Then, by the Darmois-Skitovich theorem, whenever $\boldsymbol{M}_{1,j}\boldsymbol{M}_{2,j} \neq 0$, $\bar{\boldsymbol{z}}_j$ must be constant or Gaussian. But since none of the $\bar{\boldsymbol{z}}_j$ are constant by hypothesis, $\bar{\boldsymbol{z}}_j$ must be Gaussian. Since we allow for only one Gaussian $\bar{\boldsymbol{z}}_j$, there has to be at most one j_0 such that $\boldsymbol{M}_{1,j_0}\boldsymbol{M}_{2,j_0} \neq 0$. But since $\boldsymbol{M}$ is orthogonal, its rows must be orthogonal. Hence, $0 = \langle \boldsymbol{M}_{1,\cdot}, \boldsymbol{M}_{2,\cdot} \rangle = \boldsymbol{M}_{1,j_0}\boldsymbol{M}_{2,j_0}$. We thus have that the first and second rows of $\boldsymbol{M}$ cannot have nonzero entries at the same locations. The same argument can be repeated for every pair of rows of $\boldsymbol{M}$. Hence, all pairs of rows do not have nonzero entries at the same location. The only way this is possible is if $\boldsymbol{M}$ is a permutation $\boldsymbol{P}$.

Thus,

$$\boldsymbol{V} = \text{cov}(\boldsymbol{y})^{\frac{1}{2}} \boldsymbol{P} \text{cov}(\boldsymbol{z})^{-\frac{1}{2}} \tag{2.37}$$

$$= \text{cov}(\boldsymbol{y})^{\frac{1}{2}} \underbrace{\boldsymbol{P} \text{cov}(\boldsymbol{z})^{-\frac{1}{2}} \boldsymbol{P}^\top}_{\text{diagonal}} \boldsymbol{P} \tag{2.38}$$

$$= \boldsymbol{D}\boldsymbol{P}\,, \tag{2.39}$$

where $\boldsymbol{D} := \text{cov}(\boldsymbol{y})^{\frac{1}{2}} \boldsymbol{P} \text{cov}(\boldsymbol{z})^{-\frac{1}{2}} \boldsymbol{P}^\top$ is diagonal. ■

We are now ready to prove Corollary 2.1, which is a simple matter of putting everything we saw together.

Proof (Corollary 2.1) Let $\boldsymbol{z} \sim \mathbb{P}_{\boldsymbol{z}}$ and $\hat{\boldsymbol{z}} \sim \hat{\mathbb{P}}_{\boldsymbol{z}}$. Lemma 2.2 implies that $\hat{\boldsymbol{z}} \overset{d}{=} \hat{\boldsymbol{A}}^\dagger \boldsymbol{A}\boldsymbol{z}$, where $\boldsymbol{V} := \hat{\boldsymbol{A}}^\dagger \boldsymbol{A}$ is invertible (since both $\hat{\boldsymbol{A}}^\dagger$ and $\boldsymbol{A}$ are full rank). The random vector $\boldsymbol{z}$ has no constant component and at most one Gaussian component and $\boldsymbol{y} := \boldsymbol{V}\boldsymbol{z}$ is distributed according to $\hat{\mathbb{P}}_{\boldsymbol{z}}$, which has mutually independent components. This means we can apply Theorem 2.2 to get that $\boldsymbol{V} = \boldsymbol{D}\boldsymbol{P}$. Since $\hat{\boldsymbol{A}}$ has full column-rank, its pseudo-inverse has a closed form expression: $\hat{\boldsymbol{A}}^\dagger = (\hat{\boldsymbol{A}}^\top \hat{\boldsymbol{A}})^{-1} \hat{\boldsymbol{A}}^\top$. We thus have the following:

$$\hat{\boldsymbol{A}}^\dagger \boldsymbol{A} = \boldsymbol{D}\boldsymbol{P} \tag{2.40}$$

$$(\hat{\boldsymbol{A}}^\top \hat{\boldsymbol{A}})^{-1} \hat{\boldsymbol{A}}^\top \boldsymbol{A} = \boldsymbol{D}\boldsymbol{P} \tag{2.41}$$

$$\underbrace{\hat{\boldsymbol{A}}(\hat{\boldsymbol{A}}^\top \hat{\boldsymbol{A}})^{-1} \hat{\boldsymbol{A}}^\top}_{\text{projection on the range of } \hat{\boldsymbol{A}}} \boldsymbol{A} = \hat{\boldsymbol{A}}\boldsymbol{D}\boldsymbol{P}\,. \tag{2.42}$$

But since $\hat{\boldsymbol{A}}$ and $\boldsymbol{A}$ have the same range, $\hat{\boldsymbol{A}}(\hat{\boldsymbol{A}}^\top \hat{\boldsymbol{A}})^{-1} \hat{\boldsymbol{A}}^\top \boldsymbol{A} = \boldsymbol{A}$, and thus $\boldsymbol{A} = \hat{\boldsymbol{A}}\boldsymbol{D}\boldsymbol{P}$. ■

2.5.3. AMUSE: ICA via temporal correlations

I now present an alternative approach to ICA which leverages temporal correlations as opposed to non-Gaussianity, as in standard ICA. Although this strategy was originally introduced by Tong et al. [1990], the following presentation was in part inspired from [Hyvärinen et al., 2001]. This identifiability result can be thought of as a precursor to the contribution of Chapter 5, where we relax the linearity of the mixing function and the independence of the latent factors.

We are going to make the assumption that we observe a random sequence $\boldsymbol{x} := (\boldsymbol{x}^t)_{t=1}^\infty$ explained by a latent sequence $\boldsymbol{z} := (\boldsymbol{z}^t)_{t=1}^\infty$ via $\boldsymbol{x}^t = \boldsymbol{A}\boldsymbol{z}^t$ where $\boldsymbol{A} \in \mathbb{R}^{d_x \times d_z}$ with $d_z \leq d_x$. Furthermore, we will assume that $(\boldsymbol{z}^t)_{t=1}^\infty$ is a weak-sense stationary process.

Definition 2.9. *A sequence of random vectors $(\boldsymbol{z}^t)_{t=1}^\infty$ is said to be weak-sense stationary (WSS) if $\mathbb{E}[\boldsymbol{z}^t]$, $\text{cov}(\boldsymbol{z}^t)$ and $\text{cov}(\boldsymbol{z}^t, \boldsymbol{z}^{t-\tau})$ do not depend on t.*

We note that $\text{cov}(\boldsymbol{z}^t, \boldsymbol{z}^{t-\tau})$ can depend on τ, the time lag index. Importantly, we also assume that the matrices $\text{cov}(\boldsymbol{z}^t, \boldsymbol{z}^{t-\tau})$ are diagonal for all $0 \leq \tau < t$ (and all t, but this is implied by the WSS condition). This last condition indicates a weak form of mutual independence between the sequences $(\boldsymbol{z}_i^t)_{t=1}^\infty$. We summarize these assumptions using the framework introduced before:

$$\begin{aligned}\hat{\mathcal{P}} &:= \{\mathbb{P}_{\boldsymbol{z}} \text{ s.t. } (\boldsymbol{z}^t)_{t=1}^\infty \text{ is WSS and } \text{cov}(\boldsymbol{z}^t, \boldsymbol{z}^{t-\tau}) \text{ is diagonal for all } 0 \leq \tau < t\} \\ \hat{\mathcal{F}} &:= \{\text{linear maps } \boldsymbol{A} \text{ such that } \boldsymbol{A}(\boldsymbol{z}^t)_{t=1}^\infty = (\boldsymbol{A}\boldsymbol{z}^t)_{t=1}^\infty\}\,. \end{aligned} \tag{2.43}$$

We will require the data-generating process to satisfy these slightly stricter assumptions:

$$\begin{aligned}\mathcal{P} &:= \hat{\mathcal{P}} \cap \left\{\mathbb{P}_{\boldsymbol{z}} \text{ s.t. for all } i,\, \text{var}(\boldsymbol{z}_i^t) > 0 \text{ and } \exists \tau \text{ s.t. the } \left\{\frac{\text{cov}(\boldsymbol{z}_i^t, \boldsymbol{z}_i^{t-\tau})}{\text{var}(\boldsymbol{z}_i^t)}\right\}_{i=1}^{d_z} \text{ are distinct}\right\} \\ \mathcal{F} &:= \hat{\mathcal{F}} \cap \{\text{injective maps}\}\,. \end{aligned} \tag{2.44}$$

We now show that this model is identifiable and discuss practical considerations later on.

Theorem 2.3. *Let $(\boldsymbol{A}, \mathbb{P}_{\boldsymbol{z}}) \in \mathcal{F} \times \mathcal{P}$ and $(\hat{\boldsymbol{A}}, \hat{\mathbb{P}}_{\boldsymbol{z}}) \in \hat{\mathcal{F}} \times \hat{\mathcal{P}}$ where $\mathcal{F}, \hat{\mathcal{F}}, \mathcal{P}, \hat{\mathcal{P}}$ are defined in (2.43) and (2.44). Then, whenever $\mathbb{P}_{(\hat{\boldsymbol{A}}, \hat{\mathbb{P}}_{\boldsymbol{z}})} = \mathbb{P}_{(\boldsymbol{A}, \mathbb{P}_{\boldsymbol{z}})}$, we have $\boldsymbol{A} = \hat{\boldsymbol{A}}\boldsymbol{D}\boldsymbol{P}$ where $\boldsymbol{P}$ is a permutation matrix and $\boldsymbol{D}$ is an invertible diagonal matrix.*

Proof Let $\boldsymbol{z} \sim \mathbb{P}_{\boldsymbol{z}}$ and $\hat{\boldsymbol{z}} \sim \hat{\mathbb{P}}_{\boldsymbol{z}}$ with $\boldsymbol{x}^t := \boldsymbol{A}\boldsymbol{z}^t$ and $\hat{\boldsymbol{x}}^t = \boldsymbol{A}\hat{\boldsymbol{z}}^t$.

First, we have that $\text{cov}(\boldsymbol{x}^t)$ can be diagonalized with an orthogonal matrix since it is symmetric, i.e. $\text{cov}(\boldsymbol{x}^t) = \boldsymbol{U}\Lambda\boldsymbol{U}^\top$ where $\boldsymbol{U}$ is orthogonal and Λ is diagonal. Since $\text{cov}(\boldsymbol{x}^t)$ is positive semidefinite, the diagonal entries of Λ are nonnegative. Since $\text{cov}(\boldsymbol{x}^t) = \boldsymbol{A}\text{cov}(\boldsymbol{z}^t)\boldsymbol{A}^\top$ where $\boldsymbol{A}$ is full column-rank and $\text{cov}(\boldsymbol{z}^t)$ has no zero entries on its diagonal, the rank of $\text{cov}(\boldsymbol{x}^t)$ is d_z. Thus, Λ has d_z positive values on its diagonal. Let $\bar{\Lambda} \in \mathbb{R}^{d_z \times d_z}$ be the same as Λ but where we truncated the lines and columns corresponding to the zero diagonal entries, and analogously for $\bar{\boldsymbol{U}}$. We can thus write $\text{cov}(\boldsymbol{x}^t) = \bar{\boldsymbol{U}}\bar{\Lambda}\bar{\boldsymbol{U}}^\top$.

Now define $\bar{\boldsymbol{x}}^t := \bar{\Lambda}^{-1/2}\bar{\boldsymbol{U}}^\top\boldsymbol{x}^t$, so that $\dim(\bar{\boldsymbol{x}}^t) = d_z$, and $\bar{\boldsymbol{z}}^t := \text{cov}(\boldsymbol{z}^t)^{-1/2}\boldsymbol{z}^t$. Note that

$$\bar{\boldsymbol{x}}^t = \bar{\Lambda}^{-1/2}\bar{\boldsymbol{U}}^\top\boldsymbol{A}\boldsymbol{z}^t = \bar{\Lambda}^{-1/2}\bar{\boldsymbol{U}}^\top\boldsymbol{A}\text{cov}(\boldsymbol{z}^t)^{1/2}\bar{\boldsymbol{z}}^t = \boldsymbol{M}\bar{\boldsymbol{z}}^t\,, \tag{2.45}$$

where we defined $\boldsymbol{M} := \bar{\Lambda}^{-1/2}\bar{\boldsymbol{U}}^\top\boldsymbol{A}\text{cov}(\boldsymbol{z}^t)^{1/2}$. Note that $\boldsymbol{M}$ is orthogonal since

$$\boldsymbol{M}\boldsymbol{M}^\top = \bar{\Lambda}^{-1/2}\bar{\boldsymbol{U}}^\top\boldsymbol{A}\text{cov}(\boldsymbol{z}^t)\boldsymbol{A}^\top\bar{\boldsymbol{U}}\bar{\Lambda}^{-1/2} \tag{2.46}$$

$$= \bar{\Lambda}^{-1/2}\bar{\boldsymbol{U}}^\top\text{cov}(\boldsymbol{x}^t)\bar{\boldsymbol{U}}\bar{\Lambda}^{-1/2} \tag{2.47}$$

$$= \bar{\Lambda}^{-1/2}\bar{\Lambda}\bar{\Lambda}^{-1/2} = \boldsymbol{I}\,. \tag{2.48}$$

We now investigate the lagged covariance between $\bar{\boldsymbol{x}}^t$ and $\bar{\boldsymbol{x}}^{t-\tau}$ where τ is given by the condition on the ground-truth $\mathbb{P}_{\boldsymbol{z}}$ in (2.44).

$$\begin{aligned}
\text{cov}(\bar{\boldsymbol{x}}^t, \bar{\boldsymbol{x}}^{t-\tau}) &= \text{cov}(\boldsymbol{M}\bar{\boldsymbol{z}}^t, \boldsymbol{M}\bar{\boldsymbol{z}}^{t-\tau}) && (2.49)\\
&= \boldsymbol{M}\text{cov}(\bar{\boldsymbol{z}}^t, \bar{\boldsymbol{z}}^{t-\tau})\boldsymbol{M}^\top && (2.50)\\
&= \boldsymbol{M}\text{cov}(\boldsymbol{z}^t)^{-1/2}\text{cov}(\boldsymbol{z}^t, \boldsymbol{z}^{t-\tau})\text{cov}(\boldsymbol{z}^{t-\tau})^{-1/2}\boldsymbol{M}^\top && (2.51)\\
&= \boldsymbol{M}\text{cov}(\boldsymbol{z}^t, \boldsymbol{z}^{t-\tau})\text{cov}(\boldsymbol{z}^t)^{-1}\boldsymbol{M}^\top && (2.52)
\end{aligned}$$

Furthermore, define $\tilde{\boldsymbol{x}}^t := \bar{\Lambda}^{-1/2}\bar{\boldsymbol{U}}^\top\hat{\boldsymbol{x}}^t$ and $\tilde{\boldsymbol{z}}^t := \text{cov}(\hat{\boldsymbol{z}}^t)^{-1/2}\hat{\boldsymbol{z}}^t$. Notice that

$$\tilde{\boldsymbol{x}}^t = \bar{\Lambda}^{-1/2}\bar{\boldsymbol{U}}^\top\hat{\boldsymbol{A}}\hat{\boldsymbol{z}}^t = \bar{\Lambda}^{-1/2}\bar{\boldsymbol{U}}^\top\hat{\boldsymbol{A}}\text{cov}(\hat{\boldsymbol{z}}^t)^{1/2}\tilde{\boldsymbol{z}}^t = \hat{\boldsymbol{M}}\tilde{\boldsymbol{z}}^t\,, \tag{2.53}$$

where we defined $\hat{\boldsymbol{M}} := \bar{\Lambda}^{-1/2}\bar{\boldsymbol{U}}^\top\hat{\boldsymbol{A}}\text{cov}(\hat{\boldsymbol{z}}^t)^{1/2}$. Of course, since $\mathbb{P}_{(\hat{\boldsymbol{A}},\hat{\mathbb{P}}_{\boldsymbol{z}})} = \mathbb{P}_{(\boldsymbol{A},\mathbb{P}_{\boldsymbol{z}})}$, we have that $\text{cov}(\boldsymbol{x}^t) = \text{cov}(\hat{\boldsymbol{x}}^t)$. Using steps analogous to equations (2.46) to (2.48) the fact that $\text{cov}(\boldsymbol{x}^t) = \text{cov}(\hat{\boldsymbol{x}}^t)$, we can show that $\hat{\boldsymbol{M}}$ is orthogonal.

Moreover, we can use steps analogous to equations (2.49) to (2.52) to show that

$$\text{cov}(\tilde{\boldsymbol{x}}^t, \tilde{\boldsymbol{x}}^{t-\tau}) = \hat{\boldsymbol{M}}\text{cov}(\hat{\boldsymbol{z}}^t, \hat{\boldsymbol{z}}^{t-\tau})\text{cov}(\hat{\boldsymbol{z}}^t)^{-1}\hat{\boldsymbol{M}}^\top\,. \tag{2.54}$$

Since $\text{cov}(\bar{\boldsymbol{x}}^t, \bar{\boldsymbol{x}}^{t-\tau}) = \text{cov}(\tilde{\boldsymbol{x}}^t, \tilde{\boldsymbol{x}}^{t-\tau})$, we have that

$$\boldsymbol{M}\text{cov}(\boldsymbol{z}^t, \boldsymbol{z}^{t-\tau})\text{cov}(\boldsymbol{z}^t)^{-1}\boldsymbol{M}^\top = \hat{\boldsymbol{M}}\text{cov}(\hat{\boldsymbol{z}}^t, \hat{\boldsymbol{z}}^{t-\tau})\text{cov}(\hat{\boldsymbol{z}}^t)^{-1}\hat{\boldsymbol{M}}^\top\,. \tag{2.55}$$

Note that $\text{cov}(\boldsymbol{z}^t, \boldsymbol{z}^{t-\tau})\text{cov}(\boldsymbol{z}^t)^{-1}$ is diagonal with distinct values (by hypothesis). This decomposition indicates that these diagonal elements are eigenvalues of the matrix $\text{cov}(\bar{\boldsymbol{x}}^t, \bar{\boldsymbol{x}}^{t-\tau})$ with associated eigenvectors given by the columns of $\boldsymbol{M}$. Because these d_z eigenvalues are distinct, each associated eigenspace is one dimensional. This means that the matrix of orthogonal eigenvectors $\boldsymbol{M}$ is unique up to permutation of its columns and sign flips. This implies that $\boldsymbol{M} = \hat{\boldsymbol{M}}\bar{\boldsymbol{D}}\boldsymbol{P}$ where $\boldsymbol{P}$ is a permutation matrix and $\bar{\boldsymbol{D}}$ is a diagonal matrix made of 1 and -1. We can thus write

$$\begin{aligned}
\boldsymbol{M} &= \hat{\boldsymbol{M}}\bar{\boldsymbol{D}}\boldsymbol{P} && (2.56)\\
\bar{\Lambda}^{-1/2}\bar{\boldsymbol{U}}^\top\boldsymbol{A}\text{cov}(\boldsymbol{z}^t)^{1/2} &= \bar{\Lambda}^{-1/2}\bar{\boldsymbol{U}}^\top\hat{\boldsymbol{A}}\text{cov}(\hat{\boldsymbol{z}}^t)^{1/2}\bar{\boldsymbol{D}}\boldsymbol{P} && (2.57)\\
\bar{\boldsymbol{U}}\bar{\boldsymbol{U}}^\top\boldsymbol{A}\text{cov}(\boldsymbol{z}^t)^{1/2} &= \bar{\boldsymbol{U}}\bar{\boldsymbol{U}}^\top\hat{\boldsymbol{A}}\text{cov}(\hat{\boldsymbol{z}}^t)^{1/2}\bar{\boldsymbol{D}}\boldsymbol{P}\,. && (2.58)
\end{aligned}$$

Note that $\bar{\boldsymbol{U}}\bar{\boldsymbol{U}}^\top$ is the projection on $Range(\bar{\boldsymbol{U}})$. We have that $Range(\bar{\boldsymbol{U}}) = Range(\boldsymbol{A}) = Range(\hat{\boldsymbol{A}})$ because

$$\boldsymbol{A}\text{cov}(\boldsymbol{z}^t)\boldsymbol{A}^\top = \text{cov}(\boldsymbol{x}^t) = \bar{\boldsymbol{U}}\bar{\Lambda}\bar{\boldsymbol{U}}^\top = \text{cov}(\hat{\boldsymbol{x}}^t) = \hat{\boldsymbol{A}}\text{cov}(\hat{\boldsymbol{z}}^t)\hat{\boldsymbol{A}}^\top\,. \tag{2.59}$$

Hence, the projection $\bar{\boldsymbol{U}}\bar{\boldsymbol{U}}^\top$ act as the identity when left-multiplying $\boldsymbol{A}$ and $\hat{\boldsymbol{A}}$, so that

$$\boldsymbol{A}\text{cov}(\boldsymbol{z}^t)^{1/2} = \hat{\boldsymbol{A}}\text{cov}(\hat{\boldsymbol{z}}^t)^{1/2}\bar{\boldsymbol{D}}\boldsymbol{P} \tag{2.60}$$

$$\boldsymbol{A} = \hat{\boldsymbol{A}}\text{cov}(\hat{\boldsymbol{z}}^t)^{1/2}\bar{\boldsymbol{D}}\boldsymbol{P}\text{cov}(\boldsymbol{z}^t)^{-1/2} \tag{2.61}$$

$$\boldsymbol{A} = \hat{\boldsymbol{A}}\text{cov}(\hat{\boldsymbol{z}}^t)^{1/2}\bar{\boldsymbol{D}}\underbrace{\boldsymbol{P}\text{cov}(\boldsymbol{z}^t)^{-1/2}\boldsymbol{P}^\top}_{\text{diagonal}}\boldsymbol{P} \tag{2.62}$$

$$\boldsymbol{A} = \hat{\boldsymbol{A}}\boldsymbol{D}\boldsymbol{P}\,, \tag{2.63}$$

where $\boldsymbol{D} := \text{cov}(\hat{\boldsymbol{z}}^t)^{1/2}\bar{\boldsymbol{D}}\boldsymbol{P}\text{cov}(\boldsymbol{z}^t)^{-1/2}\boldsymbol{P}^\top$ is diagonal. ■

Practical considerations: The proof strategy presented above suggests a natural algorithm to estimate the matrix $\boldsymbol{A}$ up to permutation and rescaling. First, estimate $\text{cov}(\boldsymbol{x}^t)$ from sample, find its d_z largest eigenvalues and project the observation doing $\bar{\boldsymbol{x}}^t := \bar{\Lambda}^{-1/2}\bar{\boldsymbol{U}}^\top\boldsymbol{x}^t$ (where $\bar{\boldsymbol{U}}\bar{\Lambda}\bar{\boldsymbol{U}}^\top$ is the "truncated" decomposition of the estimated covariance). Then, we can estimate $\text{cov}(\bar{\boldsymbol{x}}^t, \bar{\boldsymbol{x}}^{t-\tau})$ empirically and compute its orthogonal eigendecomposition $\boldsymbol{M}\boldsymbol{D}\boldsymbol{M}^\top$. Assuming "infinitely many samples" and that the values $\left\{\frac{\text{cov}(\boldsymbol{z}_i^t, \boldsymbol{z}_i^{t-\tau})}{\text{var}(\boldsymbol{z}_i^t)}\right\}_{i=1}^{d_z}$ are distinct in the data-generating process, we can conclude that this decomposition is unique up to permutation and sign flips so that $\boldsymbol{M}\boldsymbol{P} \approx \bar{\Lambda}^{-1/2}\bar{\boldsymbol{U}}^\top\boldsymbol{A}\text{cov}(\boldsymbol{z}^t)^{1/2}$, allowing us to compute $\boldsymbol{A}$ as a function of $\boldsymbol{M}$ (up to permutation).

Although the matrix $\text{cov}(\bar{\boldsymbol{x}}^t, \bar{\boldsymbol{x}}^{t-\tau})$ is symmetric, its finite-sample estimation might not be, thus preventing us from computing its orthogonal eigendecomposition. To sidestep this problem, one can compute the orthogonal decomposition of $\hat{\text{cov}}(\bar{\boldsymbol{x}}^t, \bar{\boldsymbol{x}}^{t-\tau}) + \hat{\text{cov}}(\bar{\boldsymbol{x}}^t, \bar{\boldsymbol{x}}^{t-\tau})^\top$ which is symmetric. This adjustment does not change the argument since

$$(\text{cov}(\bar{\boldsymbol{x}}^t, \bar{\boldsymbol{x}}^{t-\tau}) + \text{cov}(\bar{\boldsymbol{x}}^t, \bar{\boldsymbol{x}}^{t-\tau})^\top)/2 = \boldsymbol{M}\text{cov}(\boldsymbol{z}^t, \boldsymbol{z}^{t-\tau})\text{cov}(\boldsymbol{z}^t)^{-1}\boldsymbol{M}^\top\,. \tag{2.64}$$

One can also use *simultaneous diagonalization* to leverage multiple distinct time lags τ. See Hyvärinen et al. [2001] and Tong et al. [1990] for more details.

2.5.4. Nonlinear ICA

A natural question at this point is whether we can extend the identifiability of ICA to nonlinear functions $\boldsymbol{f}$. I.e., if we take $\mathcal{F}$ to be the set of all invertible transformations from $\mathbb{R}^{d_z}$ to $\mathbb{R}^{d_x}$, and keep $\mathcal{P}$ the same, do we still get identifiability up to permutation and element-wise functions? It turns out this is not the case. Under mild conditions, Hyvärinen and Pajunen [1999] showed that, given a random vector $\boldsymbol{x} \in \mathbb{R}^{d_x}$, it is always possible to find a transformation $\boldsymbol{g} : \mathbb{R}^{d_x} \to \mathbb{R}^{d_x}$ such that $\boldsymbol{g}(\boldsymbol{x})$ has independent components, and this set of solution is highly non-unique. For instance, for any given i, we can choose $\boldsymbol{g}$ so that $\boldsymbol{g}(\boldsymbol{x})_i = \boldsymbol{x}_i$.

Recent efforts, including the contributions of this thesis, have explored various model classes in hope of finding expressive models that remain identifiable in the sense of (2.30), sometimes dropping the assumption of mutual independence. We cover these works in the literature reviews of Chapters 5, 6 & 7.

2.6. Constrained optimization

At the heart of the continuous-constrained methods for causal discovery is the *augmented Lagrangian method* which transforms a constrained optimization problem into a sequence of unconstrained problems for which the solutions converge to the solution of the original constrained problem (see Bertsekas [1999] for regularity conditions). Before diving into the augmented Lagrangian approach, we review concepts of constrained optimization necessary for its understanding. This section is inspired by the presentation of Bertsekas [1999]

In its most general form, a constrained optimization problem is written as:

$$\min_{\boldsymbol{x}} f(\boldsymbol{x}) \quad \text{subject to} \quad \boldsymbol{x} \in \mathcal{X}\,, \tag{2.65}$$

where $\mathcal{X}$ is the *feasible set*. In this presentation, we always assume $f \in C^1$ and the feasible set is compact and contained in $\mathbb{R}^n$.

A point $\boldsymbol{x}^* \in \mathcal{X}$ is a *global minimum* of (2.65) if $f(\boldsymbol{x}^*) \leq f(\boldsymbol{x})\ \forall \boldsymbol{x} \in \mathcal{X}$ and is a *local minimum* of (2.65) if there exists a scalar $\epsilon > 0$ such that, $f(\boldsymbol{x}^*) \leq f(\boldsymbol{x})\ \forall \boldsymbol{x} \in \mathcal{X} \cap B_\epsilon(\boldsymbol{x}^*)$, where $B_\epsilon(\boldsymbol{x}^*) = \{\boldsymbol{x} \mid ||\boldsymbol{x} - \boldsymbol{x}^*||_2 \leq \epsilon\}$, i.e. it is the *closed ball* of radius ϵ centered at $\boldsymbol{x}^*$.

For simplicity, we consider only problems in which $\mathcal{X}$ can be written with an equality constraint, i.e.

$$\mathcal{X} = \{\boldsymbol{x} \in \mathbb{R}^n \mid \boldsymbol{h}(\boldsymbol{x}) = 0\}\,, \tag{2.66}$$

where $\boldsymbol{h} : \mathbb{R}^n \to \mathbb{R}^m$ is a function assumed to be C^1. We assume $m < n$, i.e. the number of constraints is smaller than the number of variables. The problem can be rewritten as

$$\min_{\boldsymbol{x}} f(\boldsymbol{x}) \quad \text{subject to} \quad \boldsymbol{h}(\boldsymbol{x}) = 0\,. \tag{2.67}$$

We denote by $\nabla f(\boldsymbol{x}) \in \mathbb{R}^n$ the gradient of $\boldsymbol{f}$ and by $Dh(\boldsymbol{x}) \in \mathbb{R}^{m \times n}$ the Jacobian matrix of h.

The definition of local minimum we presented does not suggest an obvious algorithm to perform optimization. The following proposition provides a standard first-order necessary condition for a point to be a local minimum of (2.67), which lends itself more naturally to numerical optimization.

Proposition 2.1 (First-order necessary conditions)**.** *Let $\boldsymbol{x}^* \in \mathcal{X}$ be a local minimum such that $Dh(\boldsymbol{x}^*)$ is full rank. Then, there exists a vector $\boldsymbol{\lambda}^* \in \mathbb{R}^m$, called the Lagrange multiplier of $\boldsymbol{x}^*$,*

such that the following equation holds:

$$\nabla f(\boldsymbol{x}^*)^\top + \boldsymbol{\lambda}^{*\intercal} D\boldsymbol{h}(\boldsymbol{x}^*) = 0\,. \tag{2.68}$$

In practice, many algorithms will find a feasible point satisfying (2.68), also called a *stationary point of* (2.67). This is considered satisfying even though a stationary point will not necessarily be a local minimum (e.g. could be a saddle point or even a local maximum).

2.6.1. The augmented Lagrangian method

We are now ready to present an algorithm to find a stationary point of (2.67). The *augmented Lagrangian* method transforms a constrained problem into a sequence of subproblems such that their solutions converge to the solution of the original problem. The *augmented Lagrangian function* $L : \mathbb{R}^n \times \mathbb{R}^m \times \mathbb{R} \to \mathbb{R}$ is central to this method and is given by

$$L(\boldsymbol{x}, \boldsymbol{\lambda}, \mu) = f(\boldsymbol{x}) + \boldsymbol{\lambda}^\intercal \boldsymbol{h}(\boldsymbol{x}) + \frac{\mu}{2}||\boldsymbol{h}(\boldsymbol{x})||_2^2\,, \tag{2.69}$$

where μ is a positive penalty coefficient. The procedure goes like this. First, initial values for μ_0 and $\boldsymbol{\lambda}_0$ are specified. Then, we approximately minimize locally $L(\boldsymbol{x}, \boldsymbol{\lambda}_0, \mu_0)$ with respect to $\boldsymbol{x}$. Next, the value of $\boldsymbol{\lambda}$ is updated via a gradient *ascent* step on the Lagrangian function with learning rate μ_t, i.e.

$$\boldsymbol{\lambda}_{t+1} \leftarrow \boldsymbol{\lambda}_t + \mu_t \boldsymbol{h}\left(\boldsymbol{x}_{t+1}\right)\,. \tag{2.70}$$

The penalty coefficient μ is also increased by a multiplicative factor.[7] We then go back to minimizing the Lagrangian function with respect to $\boldsymbol{x}$ given the new updated $\boldsymbol{\lambda}$ and μ. The same steps are repeated until convergence. The detailed procedure is given in Algorithm 1.

Algorithm 1 The augmented Lagrangian method.

procedure AUGMENTEDLAGRANGIAN($\boldsymbol{\lambda}_0, \mu_0$)
  $t \leftarrow 0$
  while Not converged **do** ▷ Insert a stopping criterion
    $\boldsymbol{x}_{t+1} \leftarrow \arg\min_{\boldsymbol{x}} L(\boldsymbol{x}, \boldsymbol{\lambda}_t, \mu_t)$ ▷ Approximate and local minimization
    $\boldsymbol{\lambda}_{t+1} \leftarrow \boldsymbol{\lambda}_t + \mu_t \boldsymbol{h}\left(\boldsymbol{x}_{t+1}\right)$
    $\mu_{t+1} \leftarrow \eta\mu_t$ ▷ $\eta > 1$ and $0 < \gamma < 1$
    $t \leftarrow t + 1$
  end while
  return $\boldsymbol{x}_t$
end procedure

[7] Different schedule also exists. For instance, in Chapter 3 & 4, we update μ only when the constraint violation reduction from the previous minimization is not sufficient.

Under specific assumptions, it turns out that if this procedure converges, it is to a stationary point $\boldsymbol{x}^*$ of (2.68). The next proposition, adapted from Bertsekas [1999], makes this statement precise.

Proposition 2.2. *([Bertsekas, 1999, Adapted from Proposition 4.2.2]) For $t = 0, 1, ...$, let $\boldsymbol{x}_t$ satisfy*

$$||\nabla_{\boldsymbol{x}} L(\boldsymbol{x}_t, \boldsymbol{\lambda}_t, \mu_t)||_2 \leq \epsilon_t \,, \tag{2.71}$$

where $\{\epsilon_t\}$ and $\{\mu_t\}$ satisfy

$$0 < \mu_t < \mu_{t+1}, \ \forall t, \ \mu_t \to \infty \,, \tag{2.72}$$

$$0 \leq \epsilon_t, \ \forall t, \ \epsilon_t \to 0 \,. \tag{2.73}$$

Assume $(\boldsymbol{x}_t, \boldsymbol{\lambda}_t) \to (\boldsymbol{x}^*, \boldsymbol{\lambda}^*)$ and $D\boldsymbol{h}(\boldsymbol{x}^*)$ is full rank. Then $(\boldsymbol{x}^*, \boldsymbol{\lambda}^*)$ satisfies the first order necessary conditions, i.e.

$$\nabla f(\boldsymbol{x}^*)^\top + \boldsymbol{\lambda}^{*\intercal} D\boldsymbol{h}(\boldsymbol{x}^*) = 0 \ \text{ and } \ \boldsymbol{h}(\boldsymbol{x}^*) = 0 \,. \tag{2.74}$$

[Bertsekas, 1999, Section 4.2.2] provides additional arguments for why the sequence $\{\mu_t\}$ is not required to go to infinity in order to obtain convergence, contrarily to what Proposition 2.2 might suggest. This is an advantage of the augmented Lagrangian over penalty methods which require $\mu_t \to \infty$, thus inducing ill-conditioning which can slow down the minimization of the subproblems.

2.7. Important gradient estimators

In machine learning, it is frequent that we wish to solve an optimization problem of the form

$$\min_{\boldsymbol{\theta}} \mathbb{E}_{\boldsymbol{z} \sim \mathbb{P}_{\boldsymbol{\theta}}} f(\boldsymbol{z}) \,, \tag{2.75}$$

where the expectation does not allow for a simple analytical form (for instance due to f being a neural network). Stochastic gradient descent (SGD) is often employed to address this issue. The gradient $\nabla_{\boldsymbol{\theta}} \mathbb{E}_{\boldsymbol{z} \sim \mathbb{P}_{\boldsymbol{\theta}}} f(\boldsymbol{z})$ can be approximate in different ways of which two are presented next.

Technical note. In what follows, we always assume that the gradient and the integral signs can be interchanged. Note that, in general, this is not always true and that, to be rigorous, one should verify for instance that the assumptions of the "dominated convergence theorem" hold [Durrett, 2011].

2.7.1. REINFORCE (a.k.a. the log derivative trick)

The REINFORCE estimator [Glynn, 1990, Williams, 1992] makes use of a simple trick to rewrite the gradient as follows:

$$\nabla_{\boldsymbol{\theta}} \mathbb{E}_{\boldsymbol{z}\sim\mathbb{P}_{\boldsymbol{\theta}}} f(\boldsymbol{z}) = \int f(\boldsymbol{z}) \nabla_{\boldsymbol{\theta}} p(\boldsymbol{z};\boldsymbol{\theta}) d\boldsymbol{z} \tag{2.76}$$

$$= \int f(\boldsymbol{z}) \frac{\nabla_{\boldsymbol{\theta}} p(\boldsymbol{z};\boldsymbol{\theta})}{p(\boldsymbol{z};\boldsymbol{\theta})} p(\boldsymbol{z};\boldsymbol{\theta}) d\boldsymbol{z} \tag{2.77}$$

$$= \int f(\boldsymbol{z}) \nabla_{\boldsymbol{\theta}} \log p(\boldsymbol{z};\boldsymbol{\theta}) p(\boldsymbol{z};\boldsymbol{\theta}) d\boldsymbol{z} \tag{2.78}$$

$$= \mathbb{E}_{\boldsymbol{z}\sim\mathbb{P}_{\boldsymbol{\theta}}} f(\boldsymbol{z}) \nabla_{\boldsymbol{\theta}} \log p(\boldsymbol{z};\boldsymbol{\theta}) \,, \tag{2.79}$$

which can be approximated via standard Monte Carlo estimation. This estimator is unbiased but known to have high variance [Rezende et al., 2014].

2.7.2. The reparameterization trick

The reparameterization trick [Kingma and Welling, 2014, Rezende et al., 2014] uses a different approach where the random variable $\boldsymbol{z} \sim p(\boldsymbol{z};\boldsymbol{\theta})$ is rewritten as $\boldsymbol{z} = \boldsymbol{g}(\boldsymbol{\epsilon};\boldsymbol{\theta})$ with $\boldsymbol{\epsilon} \sim p(\boldsymbol{\epsilon})$ such that $\boldsymbol{g}(\boldsymbol{\epsilon};\boldsymbol{\theta}) \sim p(\boldsymbol{z};\boldsymbol{\theta})$. This allows us to rewrite

$$\nabla_{\boldsymbol{\theta}} \mathbb{E}_{\boldsymbol{z}\sim\mathbb{P}_{\boldsymbol{\theta}}} f(\boldsymbol{z}) = \nabla_{\boldsymbol{\theta}} \mathbb{E}_{\boldsymbol{\epsilon}\sim p(\boldsymbol{\epsilon})} f(\boldsymbol{g}(\boldsymbol{\epsilon};\boldsymbol{\theta})) \tag{2.80}$$

$$= \mathbb{E}_{\boldsymbol{\epsilon}\sim p(\boldsymbol{\epsilon})} \nabla_{\boldsymbol{\theta}} f(\boldsymbol{g}(\boldsymbol{\epsilon};\boldsymbol{\theta})) \,, \tag{2.81}$$

which, again, can be approximated via Monte Carlo estimation. This estimator can be applied as long as such a reparameterization $\boldsymbol{g}(\boldsymbol{\epsilon};\boldsymbol{\theta})$ exists with $\boldsymbol{g}$ differentiable with respect to $\boldsymbol{\theta}$ and as long as f is differentiable (two conditions not required by the REINFORCE estimator). This estimator works well in practice and requires very few samples to give good performance (due to its low variance).

The discrete case. What if $\boldsymbol{z}$ is a multinomial random variable? Can it be reparameterized to estimate $\nabla_{\boldsymbol{\theta}} \mathbb{E}_{\boldsymbol{z}\sim\mathbb{P}_{\boldsymbol{\theta}}} f(\boldsymbol{z})$? One can use the following reparameterization

$$\boldsymbol{g}(\boldsymbol{\epsilon},\boldsymbol{\theta}) = \texttt{one\_hot}(\arg\max_i \boldsymbol{\epsilon}_i + \log \boldsymbol{\theta}_i) \,, \tag{2.82}$$

where the $\boldsymbol{\epsilon}_i$'s are mutually independent Gumbel random variables and $\boldsymbol{\theta}_i$ is the probability of sampling class i ($\sum_i \boldsymbol{\theta}_i = 1$, $\boldsymbol{\theta}_i > 0$) [Jang et al., 2017]. The problem with this formulation is that the gradient with respect to $\boldsymbol{\theta}$ is not defined (specifically on ties) which prevents the use of the reparameterization trick. Next, we present how this reparameterization can be replaced by a differentiable surrogate.

2.7.3. The Gumbel-Softmax estimator

If we replace the $\arg\max$ in (2.82) by the softmax function (which is a differentiable analog), we obtain

$$\hat{\boldsymbol{g}}(\boldsymbol{\epsilon}; \boldsymbol{\theta}) = \text{softmax}(\boldsymbol{\epsilon} + \log \boldsymbol{\theta}) \,, \tag{2.83}$$

which was proposed by Jang et al. [2017] under the name of *Gumbel-softmax distribution* and by Maddison et al. [2017] under the name of *concrete distribution*. We should emphasize that $\hat{\boldsymbol{g}}(\boldsymbol{\epsilon}; \boldsymbol{\theta})$ does not follow the same distribution as $\boldsymbol{g}(\boldsymbol{\epsilon}, \boldsymbol{\theta})$. Hence in general,

$$\nabla_{\boldsymbol{\theta}} \mathbb{E}_{\boldsymbol{\epsilon} \sim \text{Gumbel}} f(\boldsymbol{g}(\boldsymbol{\epsilon}, \boldsymbol{\theta})) \neq \nabla_{\boldsymbol{\theta}} \mathbb{E}_{\boldsymbol{\epsilon} \sim \text{Gumbel}} f(\hat{\boldsymbol{g}}(\boldsymbol{\epsilon}, \boldsymbol{\theta})) \,, \tag{2.84}$$

But the right-hand side of (2.84) can be approximated using the reparameterization trick, we can thus use it as a biased approximation.

In Chapter 4, we make use of the Gumbel-Softmax estimator in the context of causal structure learning, as presented in Ng et al. [2019] (in the context of purely observational data). The idea is to relax a discrete optimization problem by treating the discrete variables as discrete *random* variables and minimizing the expectation of the objective. This application of the Gumbel-Softmax estimator differs from usual applications in deep learning and we believe it could be applied in various combinatorial problems. In Appendix B.3 of Chapter 4, we give more details on the specifics of our implementation. Note that Chapter 5 also makes use of the Gumbel-softmax estimator to learn a causal graph over latent variables.

Prologue to the First Contribution

Article Details

Gradient-Based Neural DAG Learning
by *Sébastien Lachapelle, Philippe Brouillard, Tristan Deleu & Simon Lacoste-Julien*. This work was presented at the Eighth International Conference on Learning Representations (ICLR 2020).

Contributions of the Authors

Sébastien Lachapelle did the majority of the redaction, came up with the adaption of the acyclicity constraint to neural networks, implemented the method in PyTorch and performed most of the experiments. **Philippe Brouillard** helped with the overall experiment pipeline and implemented cross-validation for various baselines written in R. He was also responsible for the generation of synthetic datasets as well as the "Large sample size experiment". **Tristan Deleu** integrated the baselines DAG-GNN and NOTEARS in the experiment pipeline and led the experiments for these methods. **Simon Lacoste-Julien** supervised the project and provided guidance on the writing.

Context and Limitations

At the time of writing, the framework of causality had already been proposed as a way to move past the shortcomings of deep learning methods such as robustness, transferability and robustness [Pearl, 2019, Peters et al., 2017]. Essentially, the argument was that predictive machine learning systems are good at learning statistical dependencies present in the training data but would fail when some dependencies would change at inference time. The claim is that causal modelling could help to encode which “parts” of the data-generating process are stable across environments and which are not, thus yielding a more robust system. The idea of learning a causal graph over latent variables in a generative model was already floating around and was briefly explored by Bengio et al. [2020] in the bivariate case as a way to integrate causality in deep learning methods. However, methods for learning causal graphs were based on performing sequences of conditional independence test [Spirtes et al., 2000], a sequential search in the discrete space of

DAGs [Chickering, 2003, Bühlmann et al., 2014] or on integer linear programming [Jaakkola et al., 2010, Cussens, 2011] which do not lend themselves naturally to gradient-based optimization ubiquitous in neural network training. In this context, the work of Zheng et al. [2018], which formulated the combinatorial problem of learning a DAG into a continuous constrained problem approachable via gradient-methods, appeared as an obvious candidate to be integrated in modern deep learning pipelines. However, this approach was limited to linear relationships between variables, so we proposed GraN-DAG (Chapter 3) as a way to move passed this limitation.

Although Zheng et al. [2018] opened up new possibilities when it comes to optimizing over the space of DAGs, the optimization problem remains extremely challenging, even in its continuous constrained form, which remains nonconvex. Identifiability is also a challenge since it requires strong assumptions on the data-generating process (such as additive noise) that are unlikely to hold in real-world scenarios (this limitation is addressed in Chapter 4 thanks to interventional data). The cost of computing the gradient of the acyclicity constraint is cubic in the number of variables which makes scaling up to more than 100 variables challenging. That being said, Chapter 4 highlights advantages of continuous constrained methods when it comes to scaling with dataset size, thanks to stochastic gradient optimization which is also responsible for our ability to train neural networks on humongous datasets.

The Prologue of Chapter 4 discusses recent works that tackle some of these issues. Furthermore, we note that other concurrent works have also extended the work of Zheng et al. [2018] to support nonlinear relationships [Zheng et al., 2020, Ng et al., 2019, Ke et al., 2019].

Chapter 3

Gradient-Based Neural DAG Learning

Abstract

We propose a novel score-based approach to learning a directed acyclic graph (DAG) from observational data. We adapt a recently proposed continuous constrained optimization formulation to allow for nonlinear relationships between variables using neural networks. This extension allows to model complex interactions while avoiding the combinatorial nature of the problem. In addition to comparing our method to existing continuous optimization methods, we provide missing empirical comparisons to nonlinear greedy search methods. On both synthetic and real-world data sets, this new method outperforms current continuous methods on most tasks, while being competitive with existing greedy search methods on important metrics for causal inference.

3.1. Introduction

Structure learning and causal inference have many important applications in different areas of science such as genetics [Koller and Friedman, 2009, Peters et al., 2017], biology [Sachs et al., 2005] and economics [Pearl, 2009a]. *Bayesian networks* (BN), which encode conditional independencies using *directed acyclic graphs* (DAG), are powerful models which are both interpretable and computationally tractable. *Causal graphical models* (CGM) [Peters et al., 2017] are BNs which support *interventional* queries like: *What will happen if someone external to the system intervenes on variable X?* Recent work suggests that causality could partially solve challenges faced by current machine learning systems such as robustness to out-of-distribution samples, adaptability and explainability [Pearl, 2019, Magliacane et al., 2018]. However, structure and causal learning are daunting tasks due to both the combinatorial nature of the space of structures (the number of DAGs grows *super exponentially* with the number of nodes) and the question of *structure identifiability* (see Section 3.2.2). Nevertheless, these graphical models known qualities and promises of improvement for machine intelligence renders the quest for structure/causal learning appealing.

The typical motivation for learning a causal graphical model is to predict the effect of various interventions. A CGM can be best estimated when given interventional data, but interventions are often costly or impossible to obtained. As an alternative, one can use exclusively observational data and rely on different assumptions which make the graph identifiable from the distribution (see Section 3.2.2). This is the approach employed in this paper.

We propose a score-based method [Koller and Friedman, 2009] for structure learning named GraN-DAG which makes use of a recent reformulation of the original *combinatorial problem* of finding an optimal DAG into a *continuous constrained optimization problem*. In the original method named NOTEARS [Zheng et al., 2018], the directed graph is encoded as a *weighted adjacency matrix* which represents coefficients in a linear *structural equation model* (SEM) [Pearl, 2009a] (see Section 3.2.3) and enforces acyclicity using a constraint which is both efficiently computable and easily differentiable, thus allowing the use of numerical solvers. This continuous approach improved upon popular methods while avoiding the design of greedy algorithms based on heuristics.

Our first contribution is to extend the framework of Zheng et al. [2018] to deal with nonlinear relationships between variables using neural networks (NN) [Goodfellow et al., 2016]. To adapt the acyclicity constraint to our nonlinear model, we use an argument similar to what is used in Zheng et al. [2018] and apply it first at the level of *neural network paths* and then at the level of *graph paths*. Although GraN-DAG is general enough to deal with a large variety of parametric families of conditional probability distributions, our experiments focus on the special case of nonlinear Gaussian *additive noise models* since, under specific assumptions, it provides appealing theoretical guarantees easing the comparison to other graph search procedures (see Section 3.2.2 & 3.3.3). On both synthetic and real-world tasks, we show GraN-DAG often outperforms other approaches which leverage the continuous paradigm, including DAG-GNN [Yu et al., 2019b], a recent nonlinear extension of Zheng et al. [2018] which uses an evidence lower bound as score.

Our second contribution is to provide a missing empirical comparison to existing methods that support nonlinear relationships but tackle the optimization problem in its discrete form using greedy search procedures, namely CAM [Bühlmann et al., 2014] and GSF [Huang et al., 2018a]. We show that GraN-DAG is competitive on the wide range of tasks we considered, while using pre- and post-processing steps similar to CAM.

We provide an implementation of GraN-DAG here.

3.2. Background

Before presenting GraN-DAG, we review concepts relevant to structure and causal learning.

3.2.1. Causal graphical models

We suppose the natural phenomenon of interest can be described by a random vector $\boldsymbol{x} \in \mathbb{R}^d$ entailed by an underlying CGM $(\mathbb{P}_{\boldsymbol{x}}, \mathcal{G})$ where $\mathbb{P}_{\boldsymbol{x}}$ is a probability distribution over $\boldsymbol{x}$ and $\mathcal{G} = (V, E)$ is a DAG [Peters et al., 2017]. Each node $j \in V$ corresponds to exactly one variable in the system. Let $\pi_j^{\mathcal{G}}$ denote the set of parents of node j in $\mathcal{G}$ and let $\boldsymbol{x}_{\pi_j^{\mathcal{G}}}$ denote the random vector containing the variables corresponding to the parents of j in $\mathcal{G}$. Throughout the paper, we assume there are no hidden variables. In a CGM, the distribution $\mathbb{P}_{\boldsymbol{x}}$ is said to be *Markov* to $\mathcal{G}$, i.e. we can write the probability density function (pdf) of $\mathbb{P}_{\boldsymbol{x}}$ as $p(\boldsymbol{x}) = \prod_{j=1}^{d} p_j(\boldsymbol{x}_j | \boldsymbol{x}_{\pi_j^{\mathcal{G}}})$ where $p_j(\boldsymbol{x}_j | \boldsymbol{x}_{\pi_j^{\mathcal{G}}})$ is the conditional pdf of variable $\boldsymbol{x}_j$ given $\boldsymbol{x}_{\pi_j^{\mathcal{G}}}$. A CGM can be thought of as a BN in which directed edges are given a causal meaning, allowing it to answer queries regarding *interventional distributions* [Koller and Friedman, 2009].

3.2.2. Structure identifiability

In general, it is impossible to recover $\mathcal{G}$ given only samples from $\mathbb{P}_{\boldsymbol{x}}$, i.e. without *interventional data*. It is, however, customary to rely on a set of assumptions to render the structure fully or partially *identifiable*.

Definition 3.1. *Given a set of assumptions A on a CGM $\mathcal{M} = (\mathbb{P}_{\boldsymbol{x}}, \mathcal{G})$, its graph $\mathcal{G}$ is said to be identifiable from $\mathbb{P}_{\boldsymbol{x}}$ if there exists no other CGM $\tilde{\mathcal{M}} = (\tilde{\mathbb{P}}_{\boldsymbol{x}}, \tilde{\mathcal{G}})$ satisfying all assumptions in A such that $\tilde{\mathcal{G}} \neq \mathcal{G}$ and $\tilde{\mathbb{P}}_{\boldsymbol{x}} = \mathbb{P}_{\boldsymbol{x}}$.*

There are many examples of graph identifiability results for continuous variables [Peters et al., 2014, Peters and Bühlman, 2014, Shimizu et al., 2006, Zhang and Hyvärinen, 2009] as well as for discrete variables [Peters et al., 2011]. These results are obtained by assuming that the conditional densities belong to a specific parametric family. For example, if one assumes that the distribution $\mathbb{P}_{\boldsymbol{x}}$ is entailed by a structural equation model of the form

$$\boldsymbol{x}_j := f_j(\boldsymbol{x}_{\pi_j^{\mathcal{G}}}) + n_j \quad \text{with } n_j \sim \mathcal{N}(0, \sigma_j^2) \;\; \forall j \in V \tag{3.1}$$

where f_j is a nonlinear function satisfying some mild regularity conditions and the noises n_j are mutually independent, then $\mathcal{G}$ is identifiable from $\mathbb{P}_{\boldsymbol{x}}$ (see Peters et al. [2014] for the complete theorem and its proof). This is a particular instance of *additive noise models* (ANM). We will make use of this result in our experiments in Section 3.4.

One can consider weaker assumptions such as *faithfulness* [Peters et al., 2017]. This assumption allows one to identify, not $\mathcal{G}$ itself, but the *Markov equivalence class* to which it belongs [Spirtes et al., 2000]. A Markov equivalence class is a set of DAGs which encode exactly the same set of conditional independence statements and can be characterized by a graphical object named a *completed partially directed acyclic graph* (CPDAG) [Koller and Friedman, 2009, Peters et al., 2017]. Some algorithms we use as baselines in Section 3.4 output only a CPDAG.

3.2.3. NOTEARS: Continuous optimization for structure learning

Structure learning is the problem of learning $\mathcal{G}$ using a data set of n samples $\{\boldsymbol{x}^{(1)}, ..., \boldsymbol{x}^{(n)}\}$ from $\mathbb{P}_{\boldsymbol{x}}$. Score-based approaches cast this problem as an optimization problem, i.e. $\hat{\mathcal{G}} = \arg\max_{\mathcal{G} \in \text{DAG}} \mathcal{S}(\mathcal{G})$ where $\mathcal{S}(\mathcal{G})$ is a regularized maximum likelihood under graph $\mathcal{G}$. Since the number of DAGs is super exponential in the number of nodes, most methods rely on various heuristic greedy search procedures to approximately solve the problem (see Section 3.5 for a review). We now present the work of Zheng et al. [2018] which proposes to cast this combinatorial optimization problem into a continuous constrained one.

To do so, the authors propose to encode the graph $\mathcal{G}$ on d nodes as a weighted adjacency matrix $\boldsymbol{U} \in \mathbb{R}^{d \times d}$ which represents (possibly negative) coefficients in a linear SEM of the form $\boldsymbol{x}_j := \boldsymbol{u}_j^\top \boldsymbol{x} + n_i \ \ \forall j$ where $\boldsymbol{u}_j$ is the jth column of $\boldsymbol{U}$ and n_j is a noise variable. Let $\mathcal{G}_{\boldsymbol{U}}$ be the directed graph associated with the SEM and let $\boldsymbol{A}_{\boldsymbol{U}}$ be the (binary) adjacency matrix associated with $\mathcal{G}_{\boldsymbol{U}}$. One can see that the following equivalence holds:

$$(\boldsymbol{A}_{\boldsymbol{U}})_{ij} = 0 \iff \boldsymbol{U}_{ij} = 0 \tag{3.2}$$

To make sure $\mathcal{G}_{\boldsymbol{U}}$ is acyclic, the authors propose the following constraint on $\boldsymbol{U}$:

$$\operatorname{Tr} e^{\boldsymbol{U} \odot \boldsymbol{U}} - d = 0 \tag{3.3}$$

where $e^M \triangleq \sum_{k=0}^{\infty} \frac{M^k}{k!}$ is the *matrix exponential* and $\odot$ is the Hadamard product.

To see why this constraint characterizes acyclicity, first note that $(\boldsymbol{A}_{\boldsymbol{U}}{}^k)_{jj}$ is the number of cycles of length k passing through node j in graph $\mathcal{G}_{\boldsymbol{U}}$. Clearly, for $\mathcal{G}_{\boldsymbol{U}}$ to be acyclic, we must have $\operatorname{Tr} \boldsymbol{A}_{\boldsymbol{U}}{}^k = 0$ for $k = 1, 2, ..., \infty$. By equivalence (3.2), this is true when $\operatorname{Tr}(\boldsymbol{U} \odot \boldsymbol{U})^k = 0$ for $k = 1, 2, ..., \infty$. From there, one can simply apply the definition of the matrix exponential to see why constraint (3.3) characterizes acyclicity (see Zheng et al. [2018] for the full development).

The authors propose to use a regularized negative least square score (maximum likelihood for a Gaussian noise model). The resulting continuous constrained problem is

$$\max_{\boldsymbol{U}} \mathcal{S}(\boldsymbol{U}, \boldsymbol{X}) \triangleq -\frac{1}{2n} \|\boldsymbol{X} - \boldsymbol{X}\boldsymbol{U}\|_F^2 - \lambda \|\boldsymbol{U}\|_1 \quad \text{s.t.} \quad \operatorname{Tr} e^{\boldsymbol{U} \odot \boldsymbol{U}} - d = 0 \tag{3.4}$$

where $\boldsymbol{X} \in \mathbb{R}^{n \times d}$ is the design matrix containing all n samples. The nature of the problem has been drastically changed: we went from a combinatorial to a continuous problem. The difficulties of combinatorial optimization have been replaced by those of non-convex optimization, since the feasible set is non-convex. Nevertheless, a standard numerical solver for constrained optimization such has an *augmented Lagrangian method* [Bertsekas, 1999] can be applied to get an approximate solution, hence there is no need to design a greedy search procedure. Moreover, this approach is more global than greedy methods in the sense that the whole matrix U is updated at each iteration. Continuous approaches to combinatorial optimization have sometimes demonstrated improved

performance over discrete approaches in the literature (see for example Alayrac et al. [2018, §5.2] where they solve the multiple sequence alignment problem with a continuous optimization method).

3.3. GraN-DAG: Gradient-based neural DAG learning

We propose a new nonlinear extension to the framework presented in Section 3.2.3. For each variable $\boldsymbol{x}_j$, we learn a fully connected neural network with L hidden layers parametrized by $\boldsymbol{\phi}_{(j)} := \{\boldsymbol{W}_{(j)}^{(1)}, \dots, \boldsymbol{W}_{(j)}^{(L+1)}\}$ where $\boldsymbol{W}_{(j)}^{(\ell)}$ is the ℓth weight matrix of the jth NN (biases are omitted for clarity). Each NN takes as input $\boldsymbol{x}_{-j} \in \mathbb{R}^d$, i.e. the vector $\boldsymbol{x}$ with the jth component masked to zero, and outputs $\boldsymbol{\theta}_{(j)} \in \mathbb{R}^m$, the m-dimensional parameter vector of the desired distribution family for variable $\boldsymbol{x}_j$.[1] The fully connected NNs have the following form

$$\boldsymbol{\theta}_{(j)} \triangleq \boldsymbol{W}_{(j)}^{(L+1)} \boldsymbol{g}(\dots g(\boldsymbol{W}_{(j)}^{(2)} \boldsymbol{g}(\boldsymbol{W}_{(j)}^{(1)} \boldsymbol{x}_{-j})) \dots) \quad \forall j \tag{3.5}$$

where $\boldsymbol{g}$ is a nonlinearity applied element-wise. Note that the evaluation of all NNs can be parallelized on GPU. Distribution families need not be the same for each variable. Let $\boldsymbol{\phi} \triangleq \{\boldsymbol{\phi}_{(1)}, \dots, \boldsymbol{\phi}_{(d)}\}$ represents all parameters of all d NNs. Without any constraint on its parameter $\boldsymbol{\phi}_{(j)}$, neural network j models the conditional pdf $p_j(\boldsymbol{x}_j | \boldsymbol{x}_{-j}; \boldsymbol{\phi}_{(j)})$. Note that the product $\prod_{j=1}^d p_j(\boldsymbol{x}_j | \boldsymbol{x}_{-j}; \boldsymbol{\phi}_{(j)})$ does not integrate to one (i.e. it is not a joint pdf), since it does not decompose according to a DAG. We now show how one can constrain $\boldsymbol{\phi}$ to make sure the product of all conditionals outputted by the NNs is a joint pdf. The idea is to define a new weighted adjacency matrix $A_{\boldsymbol{\phi}}$ similar to the one encountered in Section 3.2.3, which can be directly used inside the constraint of Equation 3.3 to enforce acyclicity.

3.3.1. Neural network connectivity

Before defining the weighted adjacency matrix $A_{\boldsymbol{\phi}}$, we need to focus on how one can make some NN outputs unaffected by some inputs. Since we will discuss properties of a single NN, we drop the NN subscript (j) to improve readability.

We will use the term *neural network path* to refer to a computation path in a NN. For example, in a NN with two hidden layers, the sequence of weights $(\boldsymbol{W}_{h_1 i}^{(1)}, \boldsymbol{W}_{h_2 h_1}^{(2)}, \boldsymbol{W}_{k h_2}^{(3)})$ is a NN path from input i to output k. We say that a NN path is *inactive* if at least one weight along the path is zero. We can loosely interpret the *path product* $|\boldsymbol{W}_{h_1 i}^{(1)}||\boldsymbol{W}_{h_2 h_1}^{(2)}||\boldsymbol{W}_{k h_2}^{(3)}| \geq 0$ as the strength of the NN path, where a path product is equal to zero if and only if the path is inactive. Note that if all NN paths from input i to output k are inactive (i.e. the sum of their path products is zero), then output k does not depend on input i anymore since the information in input i will never reach output k. The sum of all path products from input i to output k for all input i and output k can be easily computed by

[1] Not all parameter vectors need to have the same dimensionality, but to simplify the notation, we suppose $m_j = m \ \forall j$

taking the following matrix product.

$$\boldsymbol{C} \triangleq |\boldsymbol{W}^{(L+1)}| \dots |\boldsymbol{W}^{(2)}||\boldsymbol{W}^{(1)}| \in \mathbb{R}_{\geq 0}^{m \times d} \tag{3.6}$$

where $|\boldsymbol{W}|$ is the element-wise absolute value of $\boldsymbol{W}$. Let us name $\boldsymbol{C}$ the *neural network connectivity matrix*. It can be verified that $\boldsymbol{C}_{ki}$ is the sum of all NN path products from input i to output k. This means it is sufficient to have $\boldsymbol{C}_{ki} = 0$ to render output k independent of input i.

Remember that each NN in our model outputs a parameter vector $\boldsymbol{\theta}$ for a conditional distribution and that we want the product of all conditionals to be a valid joint pdf, i.e. we want its corresponding directed graph to be acyclic. With this in mind, we see that it could be useful to make a certain parameter $\boldsymbol{\theta}$ not dependent on certain inputs of the NN. To have $\boldsymbol{\theta}$ independent of variable $\boldsymbol{x}_i$, it is sufficient to have $\sum_{k=1}^{m} \boldsymbol{C}_{ki} = 0$.

3.3.2. A weighted adjacency matrix

We now define a weighted adjacency matrix $\boldsymbol{A}_{\boldsymbol{\phi}}$ that can be used in constraint of Equation 3.3.

$$(\boldsymbol{A}_{\boldsymbol{\phi}})_{ij} := \begin{cases} \sum_{k=1}^{m} \left(\boldsymbol{C}_{(j)}\right)_{ki}, & \text{if } j \neq i \\ 0, & \text{otherwise} \end{cases} \tag{3.7}$$

where $\boldsymbol{C}_{(j)}$ denotes the connectivity matrix of the NN associated with variable $\boldsymbol{x}_j$.

As the notation suggests, $\boldsymbol{A}_{\boldsymbol{\phi}} \in \mathbb{R}_{\geq 0}^{d \times d}$ depends on all weights of all NNs. Moreover, it can effectively be interpreted as a weighted adjacency matrix similarly to what we presented in Section 3.2.3, since we have that

$$(\boldsymbol{A}_{\boldsymbol{\phi}})_{ij} = 0 \implies \boldsymbol{\theta}_{(j)} \text{ does not depend on variable } \boldsymbol{x}_i \tag{3.8}$$

We note $\mathcal{G}_{\boldsymbol{\phi}}$ to be the directed graph entailed by parameter $\boldsymbol{\phi}$. We can now write our adapted acyclicity constraint:

$$h(\boldsymbol{\phi}) \triangleq \operatorname{Tr} e^{\boldsymbol{A}_{\boldsymbol{\phi}}} - d = 0 \tag{3.9}$$

Note that we can compute the gradient of $h(\boldsymbol{\phi})$ w.r.t. $\boldsymbol{\phi}$ (except at points of non-differentiability arising from the absolute value function, similar to standard neural networks with ReLU activations [Glorot et al., 2011]; these points did not appear problematic in our experiments using SGD).

3.3.3. A differentiable score and its optimization

We propose solving the maximum likelihood optimization problem

$$\max_{\boldsymbol{\phi}} \mathbb{E}_{\boldsymbol{x} \sim \mathbb{P}_{\boldsymbol{x}}} \sum_{j=1}^{d} \log p_j(\boldsymbol{x}_j | \boldsymbol{x}_{\pi_j^{\boldsymbol{\phi}}}; \boldsymbol{\phi}_{(j)}) \quad \text{s.t.} \quad \operatorname{Tr} e^{\boldsymbol{A}_{\boldsymbol{\phi}}} - d = 0 \tag{3.10}$$

where π_j^{ϕ} denotes the set of parents of node j in graph $\mathcal{G}_{\phi}$. Note that $\sum_{j=1}^{d} \log p_j(\boldsymbol{x}_j|\boldsymbol{x}_{\pi_j^{\phi}}; \boldsymbol{\phi}_{(j)})$ is a valid log-likelihood function when constraint (3.9) is satisfied.

As suggested in Zheng et al. [2018], we apply an augmented Lagrangian approach to get an approximate solution to program (3.10). Augmented Lagrangian methods consist of optimizing a sequence of subproblems for which the exact solutions are known to converge to a stationary point of the constrained problem under some regularity conditions [Bertsekas, 1999]. In our case, each subproblem is

$$\max_{\boldsymbol{\phi}} \mathcal{L}(\boldsymbol{\phi}, \lambda_t, \mu_t) \triangleq \mathbb{E}_{\boldsymbol{x}\sim\mathbb{P}_{\boldsymbol{x}}} \sum_{j=1}^{d} \log p_j(\boldsymbol{x}_j|\boldsymbol{x}_{\pi_j^{\phi}}; \boldsymbol{\phi}_{(j)}) - \lambda_t h(\boldsymbol{\phi}) - \frac{\mu_t}{2} h(\boldsymbol{\phi})^2 \tag{3.11}$$

where λ_t and μ_t are the Lagrangian and penalty coefficients of the tth subproblem, respectively. These coefficients are updated after each subproblem is solved. Since GraN-DAG rests on neural networks, we propose to approximately solve each subproblem using a well-known stochastic gradient algorithm popular for NN in part for its implicit regularizing effect [Poggio et al., 2018]. See Appendix A for details regarding the optimization procedure.

In the current section, we presented GraN-DAG in a general manner without specifying explicitly which distribution family is parameterized by $\boldsymbol{\theta}_{(j)}$. In principle, any distribution family could be employed as long as its log-likelihood can be computed and differentiated with respect to its parameter $\boldsymbol{\theta}$. However, it is not always clear whether the exact solution of problem (3.10) recovers the ground truth graph $\mathcal{G}$. It will depend on both the modelling choice of GraN-DAG and the underlying CGM $(\mathbb{P}_{\boldsymbol{x}}, \mathcal{G})$.

Proposition 3.1. *Let $\boldsymbol{\phi}^*$ and $\mathcal{G}_{\boldsymbol{\phi}^*}$ be the optimal solution to (3.10) and its corresponding graph, respectively. Let $\mathcal{M}(\mathcal{A})$ be the set of CGM $(\mathbb{P}', \mathcal{G}')$ for which the assumptions in $\mathcal{A}$ are satisfied and let $\mathcal{C}$ be the set of CGM $(\mathbb{P}', \mathcal{G}')$ which can be represented by the model (e.g. NN outputting a Gaussian distribution). If the underlying CGM $(\mathbb{P}_{\boldsymbol{x}}, \mathcal{G}) \in \mathcal{C}$ and $\mathcal{C} = \mathcal{M}(\mathcal{A})$ for a specific set of assumptions A such that $\mathcal{G}$ is identifiable from $\mathbb{P}_{\boldsymbol{x}}$, then $\mathcal{G}_{\boldsymbol{\phi}^*} = \mathcal{G}$.*

Proof: Let $\mathbb{P}_{\boldsymbol{\phi}}$ be the joint distribution entailed by parameter $\boldsymbol{\phi}$. Note that the population log-likelihood $\mathbb{E}_{\boldsymbol{x}\sim\mathbb{P}_{\boldsymbol{x}}} \log p_{\boldsymbol{\phi}}(\boldsymbol{x})$ is maximal iff $\mathbb{P}_{\boldsymbol{\phi}} = \mathbb{P}_{\boldsymbol{x}}$. We know this maximum can be achieved by a specific parameter $\boldsymbol{\phi}^*$ since by hypothesis $(\mathbb{P}_{\boldsymbol{x}}, \mathcal{G}) \in \mathcal{C}$. Since $\mathcal{G}$ is identifiable from $\mathbb{P}_{\boldsymbol{x}}$, we know there exists no other CGM $(\tilde{\mathbb{P}}_{\boldsymbol{x}}, \tilde{\mathcal{G}}) \in \mathcal{C}$ such that $\tilde{\mathcal{G}} \neq \mathcal{G}$ and $\tilde{\mathbb{P}}_{\boldsymbol{x}} = \mathbb{P}_{\boldsymbol{x}}$. Hence $\mathcal{G}_{\boldsymbol{\phi}^*}$ has to be equal to $\mathcal{G}$.■

In Section 3.4.1, we empirically explore the identifiable setting of nonlinear Gaussian ANMs introduced in Section 3.2.2. In practice, one should keep in mind that solving (3.10) exactly is hard since the problem is non-convex (the augmented Lagrangian converges only to a stationary point) and moreover we only have access to the empirical log-likelihood (Proposition 3.1 holds for the population case).

3.3.4. Thresholding

The solution outputted by the augmented Lagrangian will satisfy the constraint only up to numerical precision, thus several entries of $\boldsymbol{A}_\phi$ might not be exactly zero and require thresholding. To do so, we mask the inputs of each NN j using a binary matrix $\boldsymbol{M}_{(j)} \in \{0,1\}^{d\times d}$ initialized to have $(\boldsymbol{M}_{(j)})_{ii} = 1 \;\; \forall i \neq j$ and zeros everywhere else. Having $(\boldsymbol{M}_{(j)})_{ii} = 0$ means the input i of NN j has been thresholded. This mask is integrated in the product of Equation 3.6 by doing $\boldsymbol{C}_{(j)} \triangleq |\boldsymbol{W}_{(j)}^{(L+1)}| \dots |\boldsymbol{W}_{(j)}^{(1)}| \boldsymbol{M}_{(j)}$ without changing the interpretation of $\boldsymbol{C}_{(j)}$ ($\boldsymbol{M}_{(j)}$ can be seen simply as an extra layer in the NN). During optimization, if the entry $(\boldsymbol{A}_{\boldsymbol{\phi}})_{ij}$ is smaller than the threshold $\epsilon = 10^{-4}$, the corresponding mask entry $(\boldsymbol{M}_{(j)})_{ii}$ is set to zero, permanently. The masks $\boldsymbol{M}_{(j)}$ are never updated via gradient descent. We also add an iterative thresholding step at the end to ensure the estimated graph $\mathcal{G}_\phi$ is acyclic (described in Appendix B).

3.3.5. Overfitting

In practice, we maximize an empirical estimate of the objective of problem (3.10). It is well known that this maximum likelihood score is prone to overfitting in the sense that adding edges can never reduce the maximal likelihood [Koller and Friedman, 2009]. GraN-DAG gets around this issue in four ways. First, as we optimize a subproblem, we evaluate its objective on a held-out data set and declare convergence once it has stopped improving. This approach is known as *early stopping* [Prechelt, 1997]. Second, to optimize (3.11), we use a stochastic gradient algorithm variant which is now known to have an implicit regularizing effect [Poggio et al., 2018]. Third, once we have thresholded our graph estimate to be a DAG, we apply a final pruning step identical to what is done in CAM [Bühlmann et al., 2014] to remove spurious edges. This step performs a regression of each node against its parents and uses a significance test to decide which parents should be kept or not. Fourth, for graphs of 50 nodes or more, we apply a *preliminary neighbors selection* (PNS) before running the optimization procedure as was also recommended in Bühlmann et al. [2014]. This procedure selects a set of potential parents for each variables. See Appendix C for details on PNS and pruning. Many score-based approaches control overfitting by penalizing the number of edges in their score. For example, NOTEARS includes the L1 norm of its weighted adjacency matrix $\boldsymbol{U}$ in its objective. GraN-DAG regularizes using PNS and pruning for ease of comparision to CAM, the most competitive approach in our experiments. The importance of PNS and pruning and their ability to reduce overfitting is illustrated in an ablation study presented in Appendix C. The study shows that PNS and pruning are both very important for the performance of GraN-DAG in terms of SHD, but do not have a significant effect in terms of SID. In these experiments, we also present NOTEARS and DAG-GNN with PNS and pruning, without noting a significant improvement.

3.3.6. Computational Complexity

To learn a graph, GraN-DAG relies on the proper training of neural networks on which it is built. We thus propose using a stochastic gradient method which is a standard choice when it comes to NN training because it scales well with both the sample size and the number of parameters and it implicitly regularizes learning. Similarly to NOTEARS, GraN-DAG requires the evaluation of the matrix exponential of $\boldsymbol{A}_\phi$ at each iteration costing $\mathcal{O}(d^3)$. NOTEARS justifies the use of a batch proximal quasi-Newton algorithm by the low number of $\mathcal{O}(d^3)$ iterations required to converge. Since GraN-DAG uses a stochastic gradient method, one would expect it will require more iterations to converge. However, in practice we observe that GraN-DAG performs fewer iterations than NOTEARS before the augmented Lagrangian converges (see Table 3.4 of Appendix A). We hypothesize this is due to early stopping which avoids having to wait until the full convergence of the subproblems hence limiting the total number of iterations. Moreover, for the graph sizes considered in this paper ($d \leq 100$), the evaluation of $h(\phi)$ in GraN-DAG, which includes the matrix exponentiation, does not dominate the cost of each iteration ($\approx 4\%$ for 20 nodes and $\approx 13\%$ for 100 nodes graphs). Evaluating the approximate gradient of the log-likelihood (costing $\mathcal{O}(d^2)$ assuming a fixed minibatch size, NN depth and width) appears to be of greater importance for $d \leq 100$.

3.4. Experiments

In this section, we compare GraN-DAG to various baselines in the continuous paradigm, namely DAG-GNN [Yu et al., 2019b] and NOTEARS [Zheng et al., 2018], and also in the combinatorial paradigm, namely CAM [Bühlmann et al., 2014], GSF [Huang et al., 2018a], GES [Chickering, 2003] and PC [Spirtes et al., 2000]. These methods are discussed in Section 3.5. In all experiments, each NN learned by GraN-DAG outputs the mean of a Gaussian distribution $\hat{\mu}_{(j)}$, i.e. $\boldsymbol{\theta}_{(j)} := \hat{\mu}_{(j)}$ and $\boldsymbol{x}_j | \boldsymbol{x}_{\pi_j^\mathcal{G}} \sim \mathcal{N}(\hat{\mu}_{(j)}, \hat{\sigma}^2_{(j)}) \;\; \forall j$. The parameters $\hat{\sigma}^2_{(j)}$ are learned as well, but do not depend on the parent variables $\boldsymbol{x}_{\pi_j^\mathcal{G}}$ (unless otherwise stated). Note that this modelling choice matches the nonlinear Gaussian ANM introduced in Section 3.2.2.

We report the performance of random graphs sampled using the *Erdős-Rényi* (ER) scheme described in Appendix E (denoted by RANDOM). For each approach, we evaluate the estimated graph on two metrics: the *structural hamming distance* (SHD) and the *structural interventional distance* (SID) [Peters and Bühlmann, 2015]. The former simply counts the number of missing, falsely detected or reversed edges. The latter is especially well suited for causal inference since it counts the number of couples (i, j) such that the interventional distribution $p(\boldsymbol{x}_j | do(\boldsymbol{x}_i = \bar{x}))$ would be miscalculated if we were to use the estimated graph to form the parent adjustement set. Note that GSF, GES and PC output only a CPDAG, hence the need to report a lower and an upper bound on the SID. See Appendix G for more details on SHD and SID. All experiments were ran

with publicly available code from the authors website. See Appendix H for the details of their hyperparameters. In Appendix I, we explain how one could use a held-out data set to select the hyperparameters of score-based approaches and report the results of such a procedure on almost every settings discussed in the present section.

3.4.1. Synthetic data

We have generated different *data set types* which vary along four dimensions: data generating process, number of nodes, level of edge sparsity and graph type. We consider two graph sampling schemes: *Erdős-Rényi* (ER) and *scale-free* (SF) (see Appendix E for details). For each data set type, we sampled 10 data sets of 1000 examples as follows: First, a ground truth DAG $\mathcal{G}$ is randomly sampled following either the ER or the SF scheme. Then, the data is generated according to a specific sampling scheme.

The first data generating process we consider is the nonlinear Gaussian ANM (Gauss-ANM) introduced in Section 3.2.2 in which data is sampled following $\boldsymbol{x}_j := f_j(\boldsymbol{x}_{\pi_j^{\mathcal{G}}}) + n_j$ with mutually independent noises $n_j \sim \mathcal{N}(0, \sigma_j^2)\ \ \forall j$ where the functions f_j are independently sampled from a Gaussian process with a unit bandwidth RBF kernel and with σ_j^2 sampled uniformly. As mentioned in Section 3.2.2, we know $\mathcal{G}$ to be identifiable from the distribution. Proposition 3.1 indicates that the modelling choice of GraN-DAG together with this synthetic data ensure that solving (3.10) to optimality would recover the correct graph. Note that NOTEARS and CAM also make the correct Gaussian noise assumption, but do not have enough capacity to represent the f_j functions properly.

We considered graphs of 10, 20, 50 and 100 nodes. Tables 3.1 & 3.2 present results only for 10 and 50 nodes since the conclusions do not change with graphs of 20 or 100 nodes (see Appendix F for these additional experiments). We consider graphs of d and $4d$ edges (respectively denoted by ER1 and ER4 in the case of ER graphs). We report the performance of the popular GES and PC in Appendix F since they are almost never on par with the best methods presented in this section.

	ER1		ER4		SF1		SF4	
	SHD	SID	SHD	SID	SHD	SID	SHD	SID
GraN-DAG	**1.7±2.5**	**1.7±3.1**	**8.3±2.8**	**21.8±8.9**	**1.2±1.1**	**4.1±6.1**	**9.9±4.0**	**16.4±6.0**
DAG-GNN	11.4±3.1	37.6±14.4	35.1±1.5	81.9±4.7	9.9±1.1	29.7±15.8	20.8±1.9	48.4±15.6
NOTEARS	12.2±2.9	36.6±13.1	32.6±3.2	79.0±4.1	10.7±2.2	32.0±15.3	20.8±2.7	49.8±15.6
CAM	**1.1±1.1**	**1.1±2.4**	**12.2±2.7**	**30.9±13.2**	**1.4±1.6**	**5.4±6.1**	**9.8±4.3**	**19.3±7.5**
GSF	6.5±2.6	[6.2±10.8 17.7±12.3]	21.7±8.4	[37.2±19.2 62.7±14.9]	**1.8±1.7**	**[2.0±5.1 6.9±6.2]**	**8.5±4.2**	**[13.2±6.8 20.6±12.1]**
RANDOM	26.3±9.8	25.8±10.4	31.8±5.0	76.6±7.0	25.1±10.2	24.5±10.5	28.5±4.0	47.2±12.2

Table 3.1. Results for ER and SF graphs of 10 nodes with Gauss-ANM data

	ER1		ER4		SF1		SF4	
	SHD	SID	SHD	SID	SHD	SID	SHD	SID
GraN-DAG	**5.1±2.8**	**22.4±17.8**	**102.6±21.2**	**1060.1±109.4**	**25.5±6.2**	**90.0±18.9**	**111.3±12.3**	**271.2±65.4**
DAG-GNN	49.2±7.9	304.4±105.1	191.9±15.2	2146.2±64	49.8±1.3	182.8±42.9	144.9±13.3	540.8±151.1
NOTEARS	62.8±9.2	327.3±119.9	202.3±14.3	2149.1±76.3	57.7±3.5	195.7±54.9	153.7±11.8	558.4±153.5
CAM	**4.3±1.9**	**22.0±17.9**	**98.8±20.7**	**1197.2±125.9**	**24.1±6.2**	**85.7±31.9**	**111.2±13.3**	**320.7±152.6**
GSF	25.6±5.1	[21.1±23.1 79.2±33.5]	**81.8±18.8**	**[906.6±214.7 1030.2±172.6]**	**31.6±6.7**	[85.8±29.9 147.3±49.9]	**120.2±10.9**	[284.7±80.2 379.9±98.3]
RANDOM	535.7±401.2	272.3±125.5	708.4±234.4	1921.3±203.5	514.0±360.0	381.3±190.3	660.6±194.9	1198.9±304.6

Table 3.2. Results for ER and SF graphs of 50 nodes with Gauss-ANM data

We now examine Tables 3.1 & 3.2 (the errors bars represent the standard deviation across datasets per task). We can see that, across all settings, GraN-DAG and CAM are the best performing methods, both in terms of SHD and SID, while GSF is not too far behind. The poor performance of NOTEARS can be explained by its inability to model nonlinear functions. In terms of SHD, DAG-GNN performs rarely better than NOTEARS while in terms of SID, it performs similarly to RANDOM in almost all cases except in scale-free networks of 50 nodes or more. Its poor performance might be due to its incorrect modelling assumptions and because its architecture uses a strong form of parameter sharing between the f_j functions, which is not justified in a setup like ours. GSF performs always better than DAG-GNN and NOTEARS but performs as good as CAM and GraN-DAG only about half the time. Among the continuous approaches considered, GraN-DAG is the best performing on these synthetic tasks.

Even though CAM (wrongly) assumes that the functions f_j are additive, i.e. $f_j(\boldsymbol{x}_{\pi_j^{\mathcal{G}}}) = \sum_{i \in \pi_j^{\mathcal{G}}} f_{ij}(\boldsymbol{x}_j) \ \ \forall j$, it manages to compete with GraN-DAG which does not make this incorrect modelling assumption[2]. This might partly be explained by a bias-variance trade-off. CAM is biased but has a lower variance than GraN-DAG due to its restricted capacity, resulting in both methods performing similarly. In Appendix D, we present an experiment showing that GraN-DAG can outperform CAM in higher sample size settings, suggesting this explanation is reasonable.

Having confirmed that GraN-DAG is competitive on the ideal Gauss-ANM data, we experimented with settings better adjusted to other models to see whether GraN-DAG remains competitive. We considered linear Gaussian data (better adjusted to NOTEARS) and nonlinear Gaussian data with additive functions (better adjusted to CAM) named LIN and ADD-FUNC, respectively. See Appendix E for the details of their generation. We report the results of GraN-DAG and other baselines in Table 3.12 & 3.13 of the appendix. On linear Gaussian data, most methods score poorly in terms of SID which is probably due to the unidentifiability of the linear Gaussian model (when the noise variances are unequal). GraN-DAG and CAM perform similarly to NOTEARS in terms of SHD. On ADD-FUNC, CAM dominates all methods on most graph types considered (GraN-DAG is on par only for the 10 nodes ER1 graph). However, GraN-DAG outperforms all other methods

[2]Although it is true that GraN-DAG does not wrongly assume that the functions f_j are additive, it is not clear whether its neural networks can exactly represent functions sampled from the Gaussian process.

which can be explained by the fact that the conditions of Proposition 3.1 are satisfied (supposing the functions $\sum_{i\in\pi_j^{\mathcal{G}}} f_{ij}(\boldsymbol{x}_i)$ can be represented by the NNs).

We also considered synthetic data sets which do not satisfy the additive Gaussian noise assumption present in GraN-DAG, NOTEARS and CAM. We considered two kinds of *post nonlinear causal models* [Zhang and Hyvärinen, 2009], PNL-GP and PNL-MULT (see Appendix E for details about their generation). A post nonlinear model has the form $\boldsymbol{x}_j := g_j(f_j(\boldsymbol{x}_{\pi_j^{\mathcal{G}}}) + n_j)$ where n_j is a noise variable. Note that GraN-DAG (with the current modelling choice) and CAM do not have the representational power to express these conditional distributions, hence violating an assumption of Proposition 3.1. However, these data sets differ from the previous additive noise setup only by the nonlinearity g_j, hence offering a case of mild model misspecification. The results are reported in Table 3.14 of the appendix. GraN-DAG and CAM are outperforming DAG-GNN and NOTEARS except in SID for certain data sets where all methods score similarly to RANDOM. GraN-DAG and CAM have similar performance on all data sets except one where CAM is better. GSF performs worst than GraN-DAG (in both SHD and SID) on PNL-GP but not on PNL-MULT where it performs better in SID.

3.4.2. Real and pseudo-real data

We have tested all methods considered so far on a well known data set which measures the expression level of different proteins and phospholipids in human cells [Sachs et al., 2005]. We trained only on the $n = 853$ observational samples. This dataset and its ground truth graph proposed in Sachs et al. [2005] (11 nodes and 17 edges) are often used in the probabilistic graphical model literature [Koller and Friedman, 2009]. We also consider pseudo-real data sets sampled from the SynTReN generator [Van den Bulcke, 2006]. This generator was designed to create synthetic transcriptional regulatory networks and produces simulated gene expression data that approximates experimental data. See Appendix E for details of the generation.

In applications, it is not clear whether the conditions of Proposition 3.1 hold since we do not know whether $(\mathbb{P}_{\boldsymbol{x}}, \mathcal{G}) \in \mathcal{C}$. This departure from identifiable settings is an occasion to explore a different modelling choice for GraN-DAG. In addition to the model presented at the beginning of this section, we consider an alternative, denoted GraN-DAG++, which allows the variance parameters $\hat{\sigma}^2_{(i)}$ to depend on the parent variables $\boldsymbol{x}_{\pi_i^{\mathcal{G}}}$ through the NN, i.e. $\boldsymbol{\theta}_{(i)} := (\hat{\mu}_{(i)}, \log \hat{\sigma}^2_{(i)})$. Note that this is violating the additive noise assumption (in ANMs, the noise is independent of the parent variables).

In addition to metrics used in Section 3.4.1, we also report SHD-C. To compute the SHD-C between two DAGs, we first map each of them to their corresponding CPDAG and measure the SHD between the two. This metric is useful to compare algorithms which only outputs a CPDAG like GSF, GES and PC to other methods which outputs a DAG. Results are reported in Table 3.3.

	Protein signaling data set			SynTReN (20 nodes)		
	SHD	SHD-C	SID	SHD	SHD-C	SID
GraN-DAG	13	11	47	**34.0±8.5**	**36.4±8.3**	161.7±53.4
GraN-DAG++	13	10	48	**33.7±3.7**	**39.4±4.9**	127.5±52.8
DAG-GNN	16	21	44	93.6±9.2	97.6±10.3	157.5±74.6
NOTEARS	21	21	44	151.8±28.2	156.1±28.7	**110.7±66.7**
CAM	**12**	**9**	55	**40.5±6.8**	**41.4±7.1**	152.3±48
GSF	18	10	[44, 61]	61.8±9.6	63.3±11.4	**[76.7±51.1, 109.9±39.9]**
GES	26	28	**[34, 45]**	82.6±9.3	85.6±10	[157.2±48.3, 168.8±47.8]
PC	17	11	[47, 62]	**41.2±5.1**	**42.4±4.6**	[154.8±47.6, 179.3±55.6]
RANDOM	21	20	60	84.7±53.8	86.7±55.8	175.8±64.7

Table 3.3. Results on real and pseudo-real data

First, all methods perform worse than what was reported for graphs of similar size in Section 3.4.1, both in terms of SID and SHD. This might be due to the lack of identifiability guarantees we face in applications. On the protein data set, GraN-DAG outperforms CAM in terms of SID (which differs from the general trend of Section 3.4.1) and arrive almost on par in terms of SHD and SHD-C. On this data set, DAG-GNN has a reasonable performance, beating GraN-DAG in SID, but not in SHD. On SynTReN, GraN-DAG obtains the best SHD but not the best SID. Overall, GraN-DAG is always competitive with the best methods of each task.

3.5. Related Work

Most methods for structure learning from observational data make use of some identifiability results similar to the ones raised in Section 3.2.2. Roughly speaking, there are two classes of methods: *independence-based* and *score-based* methods. GraN-DAG falls into the second class.

Score-based methods [Koller and Friedman, 2009, Peters et al., 2017] cast the problem of structure learning as an optimization problem over the space of structures (DAGs or CPDAGs). Many popular algorithms tackle the combinatorial nature of the problem by performing a form of greedy search. GES [Chickering, 2003] is a popular example. It usually assumes a linear parametric model with Gaussian noise and greedily search the space of CPDAGs in order to optimize the Bayesian information criterion. GSF [Huang et al., 2018a], is based on the same search algorithm as GES, but uses a generalized score function which can model nonlinear relationships. Other greedy approaches rely on parametric assumptions which render $\mathcal{G}$ fully identifiable. For example, Peters and Bühlman [2014] relies on a linear Gaussian model with equal variances to render the DAG identifiable. RESIT [Peters et al., 2014], assumes nonlinear relationships with additive Gaussian noise and greedily maximizes an independence-based score. However, RESIT does not scale well to graph of more than 20 nodes. CAM [Bühlmann et al., 2014] decouples the search for the optimal node ordering from the parents selection for each node and assumes an additive noise model (ANM) [Peters et al., 2017] in which the nonlinear functions are additive. As mentioned

in Section 3.2.3, NOTEARS, proposed in Zheng et al. [2018], tackles the problem of finding an optimal DAG as a continuous constrained optimization program. This is a drastic departure from previous combinatorial approaches which enables the application of well studied numerical solvers for continuous optimizations. Recently, Yu et al. [2019b] proposed DAG-GNN, a graph neural network architecture (GNN) which can be used to learn DAGs via the maximization of an evidence lower bound. By design, a GNN makes use of parameter sharing which we hypothesize is not well suited for most DAG learning tasks. To the best of our knowledge, DAG-GNN is the first approach extending the NOTEARS algorithm for structure learning to support nonlinear relationships. Although Yu et al. [2019b] provides empirical comparisons to linear approaches, namely NOTEARS and FGS (a faster extension of GES) [Ramsey et al., 2017], comparisons to greedy approaches supporting nonlinear relationships such as CAM and GSF are missing. Moreover, GraN-DAG significantly outperforms DAG-GNN on our benchmarks. There exists certain score-based approaches which uses integer linear programming (ILP) [Jaakkola et al., 2010, Cussens, 2011] which internally solve continuous linear relaxations. Connections between such methods and the continuous constrained approaches are yet to be explored.

When used with the additive Gaussian noise assumption, the theoretical guarantee of GraN-DAG rests on the identifiability of nonlinear Gaussian ANMs. Analogously to CAM and NOTEARS, this guarantee holds only if the correct identifiability assumptions hold in the data and if the score maximization problem is solved exactly (which is not the case in all three algorithms). DAG-GNN provides no theoretical justification for its approach. NOTEARS and CAM are designed to handle what is sometimes called the *high-dimensional setting* in which the number of samples is significantly smaller than the number of nodes. Bühlmann et al. [2014] provides consistency results for CAM in this setting. GraN-DAG and DAG-GNN were not designed with this setting in mind and would most likely fail if confronted to it. Solutions for fitting a neural network on less data points than input dimensions are not common in the NN literature.

Methods for causal discovery using NNs have already been proposed. SAM [Kalainathan et al., 2018] learns conditional NN generators using adversarial losses but does not enforce acyclicity. CGNN [Goudet et al., 2018], when used for multivariate data, requires an initial skeleton to learn the different functional relationships.

GraN-DAG has strong connections with MADE [Germain et al., 2015], a method used to learn distributions using a masked NN which enforces the so-called *autoregressive property*. The autoregressive property and acyclicity are in fact equivalent. MADE does not learn the weight masking, it fixes it at the beginning of the procedure. GraN-DAG could be used with a unique NN taking as input all variables and outputting parameters for all conditional distributions. In this case, it would be similar to MADE, except the variable ordering would be learned from data instead of fixed *a priori*.

3.6. Conclusion

The continuous constrained approach to structure learning has the advantage of being more global than other approximate greedy methods (since it updates all edges at each step based on the gradient of the score but also the acyclicity constraint) and allows to replace task-specific greedy algorithms by appropriate off-the-shelf numerical solvers. In this work, we have introduced GraN-DAG, a novel score-based approach for structure learning supporting nonlinear relationships while leveraging a continuous optimization paradigm. The method rests on a novel characterization of acyclicity for NNs based on the work of Zheng et al. [2018]. We showed GraN-DAG outperforms other gradient-based approaches, namely NOTEARS and its recent nonlinear extension DAG-GNN, on the synthetic data sets considered in Section 3.4.1 while being competitive on real and pseudo-real data sets of Section 3.4.2. Compared to greedy approaches, GraN-DAG is competitive across all datasets considered. To the best of our knowledge, GraN-DAG is the first approach leveraging the continuous paradigm introduced in Zheng et al. [2018] which has been shown to be competitive with state of the art methods supporting nonlinear relationships.

Appendices of Chapter 3

A. Optimization

Let us recall the augmented Lagrangian:

$$\max_{\boldsymbol{\phi}} \mathcal{L}(\boldsymbol{\phi}, \lambda_t, \mu_t) \triangleq \mathbb{E}_{\boldsymbol{x}\sim\mathbb{P}_{\boldsymbol{x}}} \sum_{i=1}^{d} \log p_i(\boldsymbol{x}_i|\boldsymbol{x}_{\pi_i^{\boldsymbol{\phi}}}; \boldsymbol{\phi}_{(i)}) - \lambda_t h(\boldsymbol{\phi}) - \frac{\mu_t}{2} h(\boldsymbol{\phi})^2 \tag{3.12}$$

where λ_t and μ_t are the Lagrangian and penalty coefficients of the tth subproblem, respectively. In all our experiments, we initialize those coefficients using $\lambda_0 = 0$ and $\mu_0 = 10^{-3}$. We approximately solve each non-convex subproblem using RMSprop [Tieleman and Hinton, 2012], a stochastic gradient descent variant popular for NNs. We use the following gradient estimate:

$$\begin{aligned} &\nabla_{\boldsymbol{\phi}} \mathcal{L}(\boldsymbol{\phi}, \lambda_t, \mu_t) \approx \nabla_{\boldsymbol{\phi}} \hat{\mathcal{L}}_B(\boldsymbol{\phi}, \lambda_t, \mu_t) \\ &\text{with } \hat{\mathcal{L}}_B(\boldsymbol{\phi}, \lambda_t, \mu_t) \triangleq \frac{1}{|B|} \sum_{\boldsymbol{x}\in B} \sum_{i=1}^{d} \log p_i(\boldsymbol{x}_i|\boldsymbol{x}_{\pi_i^{\boldsymbol{\phi}}}; \boldsymbol{\phi}_{(i)}) - \lambda_t h(\boldsymbol{\phi}) - \frac{\mu_t}{2} h(\boldsymbol{\phi})^2 \end{aligned} \tag{3.13}$$

where B is a minibatch sampled from the data set and $|B|$ is the minibatch size. The gradient estimate $\nabla_{\boldsymbol{\phi}} \hat{\mathcal{L}}_B(\boldsymbol{\phi}, \lambda_t, \mu_t)$ can be computed using standard deep learning libraries. We consider a subproblem has converged when $\hat{\mathcal{L}}_H(\boldsymbol{\phi}, \lambda_t, \mu_t)$ evaluated on a held-out data set H stops increasing. Let $\boldsymbol{\phi}_t^*$ be the approximate solution to subproblem t. Then, λ_t and μ_t are updated according to the following rule:

$$\begin{aligned} \lambda_{t+1} &\leftarrow \lambda_t + \mu_t h\left(\boldsymbol{\phi}_t^*\right) \\ \mu_{t+1} &\leftarrow \begin{cases} \eta\mu_t, & \text{if } h\left(\boldsymbol{\phi}_t^*\right) > \gamma h\left(\boldsymbol{\phi}_{t-1}^*\right) \\ \mu_t, & \text{otherwise} \end{cases} \end{aligned} \tag{3.14}$$

with $\eta = 10$ and $\gamma = 0.9$. Each subproblem t is initialized using the previous subproblem solution $\boldsymbol{\phi}_{t-1}^*$. The augmented Lagrangian method stops when $h(\boldsymbol{\phi}) \leq 10^{-8}$.

Total number of iterations before augmented Lagrangian converges: In GraN-DAG and NOTEARS, every subproblem is approximately solved using an iterative algorithm. Let T be the number of subproblems solved before the convergence of the augmented Lagrangian. For a given subproblem t, let K_t be the number of iterations executed to approximately solve it. Let

$I = \sum_{t=1}^{T} K_t$ be the *total number of iterations* before the augmented Lagrangian converges. Table 3.4 reports the total number of iterations I for GraN-DAG and NOTEARS, averaged over ten data sets. Note that the matrix exponential is evaluated once per iteration. Even though GraN-DAG uses a stochastic gradient algorithm, it requires less iterations than NOTEARS which uses a batch proximal quasi-Newton method. We hypothesize early stopping avoids having to wait until full convergence before moving to the next subproblem, hence reducing the total number of iterations. Note that GraN-DAG total run time is still larger than NOTEARS due to its gradient requiring more computation to evaluate (total runtime $\approx$ 10 minutes against $\approx$ 1 minute for 20 nodes graphs and $\approx$ 4 hours against $\approx$ 1 hour for 100 nodes graphs). GraN-DAG runtime on 100 nodes graphs can be roughly halved when executed on GPU.

	20 nodes ER1	20 nodes ER4	100 nodes ER1	100 nodes ER4
GraN-DAG	27.3 ± 3.6	30.4 ± 4.2	23.1 ± 0.7	23.1 ± 0.8
NOTEARS	67.1 ± 35.3	72.3 ± 24.3	243.6 ± 12.3	232.4 ± 12.9

Table 3.4. Total number of iterations ($\times 10^3$) before augmented Lagrangian converges on Gauss-ANM data.

B. Thresholding to ensure acyclicity

The augmented Lagrangian outputs $\boldsymbol{\phi}_T^*$ where T is the number of subproblems solved before declaring convergence. Note that the weighted adjacency matrix $A_{\boldsymbol{\phi}_T^*}$ will most likely not represent an acyclic graph, even if we threshold as we learn, as explained in Section 3.3.4. We need to remove additional edges to obtain a DAG (edges are removed using the mask presented in Section 3.3.4). One option would be to remove edges one by one until a DAG is obtained, starting from the edge (i, j) with the lowest $(A_{\boldsymbol{\phi}_T^*})_{ij}$ up to the edge with the highest $(A_{\boldsymbol{\phi}_T^*})_{ij}$. This amounts to gradually increasing the threshold ϵ until $A_{\boldsymbol{\phi}_T^*}$ is acyclic. However, this approach has the following flaw: It is possible to have $(A_{\boldsymbol{\phi}_T^*})_{ij}$ significantly higher than zero while having $\theta_{(j)}$ almost completely independent of variable $\boldsymbol{x}_i$. This can happen for at least two reasons. First, the NN paths from input i to output k might end up cancelling each others, rendering the input i inactive. Second, some neurons of the NNs might always be saturated for the observed range of inputs, rendering some NN paths *effectively inactive* without being inactive in the sense described in Section 3.3.1. Those two observations illustrate the fact that having $(A_{\boldsymbol{\phi}_T^*})_{ij} = 0$ is only a sufficient condition to have $\theta_{(j)}$ independent of variable $\boldsymbol{x}_i$ and not a necessary one.

To avoid this issue, we consider the following alternative. Consider the function $\mathcal{L} : \mathbb{R}^d \mapsto \mathbb{R}^d$ which maps all d variables to their respective conditional likelihoods, i.e. $\mathcal{L}_i(\boldsymbol{x}) \triangleq p_i(\boldsymbol{x}_i \mid \boldsymbol{x}_{\pi_i^{\boldsymbol{\phi}_T^*}}) \ \forall i$.

We consider the following expected Jacobian matrix

$$\boldsymbol{J} \triangleq \mathbb{E}_{\boldsymbol{x}\sim\mathbb{P}_{\boldsymbol{x}}} \left|\frac{\partial \mathcal{L}}{\partial \boldsymbol{x}}\right|^{\top} \tag{3.15}$$

where $\left|\frac{\partial \mathcal{L}}{\partial \boldsymbol{x}}\right|$ is the Jacobian matrix of $\mathcal{L}$ evaluated at $\boldsymbol{x}$, in absolute value (element-wise). Similarly to $(\boldsymbol{A}_{\phi_T^*})_{ij}$, the entry $\boldsymbol{J}_{ij}$ can be loosely interpreted as the strength of edge (i, j). We propose removing edges starting from the lowest $\boldsymbol{J}_{ij}$ to the highest, stopping as soon as acyclicity is achieved. We believe $\boldsymbol{J}$ is better than $\boldsymbol{A}_{\phi_T^*}$ at capturing which NN inputs are effectively inactive since it takes into account NN paths cancelling each others and saturated neurons. Empirically, we found that using $\boldsymbol{J}$ instead of $\boldsymbol{A}_{\phi_T^*}$ yields better results, and thus we report the results with $\boldsymbol{J}$ in this paper.

C. Preliminary neighborhood selection and DAG Pruning

PNS: For graphs of 50 nodes or more, GraN-DAG performs a *preliminary neighborhood selection* (PNS) similar to what has been proposed in Bühlmann et al. [2014]. This procedure applies a variable selection method to get a set of possible parents for each node. This is done by fitting an *extremely randomized trees* [Geurts et al., 2006] (using `ExtraTreesRegressor` from `scikit-learn`) for each variable against all the other variables. For each node a feature importance score based on the gain of purity is calculated. Only nodes that have a feature importance score higher than $0.75 \cdot$ `mean` are kept as potential parent, where `mean` is the mean of the feature importance scores of all nodes. Although the use of PNS in CAM was motivated by gains in computation time, GraN-DAG uses it to avoid overfitting, without reducing the computation time.

Pruning: Once the thresholding is performed and a DAG is obtained as described in B, GraN-DAG performs a pruning step identical to CAM [Bühlmann et al., 2014] in order to remove spurious edges. We use the implementation of Bühlmann et al. [2014] based on the R function `gamboost` from the `mboost` package. For each variable $\boldsymbol{x}_i$, a generalized additive model is fitted against the current parents of $\boldsymbol{x}_i$ and a significance test of covariates is applied. Parents with a p-value higher than 0.001 are removed from the parent set. Similarly to what Bühlmann et al. [2014] observed, this pruning phase generally has the effect of greatly reducing the SHD without considerably changing the SID.

Ablation study: In Table 3.5, we present an ablation study which shows the effect of adding PNS and pruning to GraN-DAG on different performance metrics and on the negative log-likelihood (NLL) of the training and validation set. Note that, before computing both NLL, we reset all parameters of GraN-DAG except the mask and retrained the model on the training set without any acyclicity constraint (acyclicity is already ensure by the masks at this point). This retraining procedure is important since the pruning removes edges (i.e. some additional NN inputs are masked) and it affects the likelihood of the model (hence the need to retrain).

PNS	Pruning	SHD	SID	NLL (train)	NLL (validation)
False	False	1086.8±48.8	31.6±23.6	0.36±0.07	1.44±0.21
True	False	540.4±70.3	17.4±16.7	0.52±0.08	1.16±0.17
False	True	11.8±5.0	39.7±25.5	0.78±0.12	0.84±0.12
True	True	6.1±3.3	29.3±19.5	0.78±0.13	0.83±0.12

Table 3.5. PNS and pruning ablation study for GraN-DAG (averaged over 10 datasets from ER1 with 50 nodes)

A first observation is that adding PNS and pruning improve the NLL on the validation set while deteriorating the NLL on the training set, showing that those two steps are indeed reducing overfitting. Secondly, the effect on SHD is really important while the effect on SID is almost nonexistent. This can be explained by the fact that SID has more to do with the ordering of the nodes than with false positive edges. For instance, if we have a complete DAG with a node ordering coherent with the ground truth graph, the SID is zero, but the SHD is not due to all the false positive edges. Without the regularizing effect of PNS and pruning, GraN-DAG manages to find a DAG with a good ordering but with many spurious edges (explaining the poor SHD, the good SID and the big gap between the NLL of the training set and validation set). PNS and pruning helps reducing the number of spurious edges, hence improving SHD.

We also implemented PNS and pruning for NOTEARS and DAG-GNN to see whether their performance could also be improved. Table 3.6 reports an ablation study for DAG-GNN and NOTEARS. First, the SHD improvement is not as important as for GraN-DAG and is almost not statistically significant. The improved SHD does not come close to performance of GraN-DAG. Second, PNS and pruning do not have a significant effect of SID, as was the case for GraN-DAG. The lack of improvement for those methods is probably due to the fact that they are not overfitting like GraN-DAG, as the training and validation (unregularized) scores shows. NOTEARS captures only linear relationships, thus it will have a hard time overfitting nonlinear data and DAG-GNN uses a strong form of parameter sharing between its conditional densities which possibly cause underfitting in a setup where all the parameters of the conditionals are sampled independently.

Moreover, DAG-GNN and NOTEARS threshold aggressively their respective weighted adjacency matrix at the end of training (with the default parameters used in the code), which also acts as a form of heavy regularization, and allow them to remove many spurious edges. GraN-DAG without PNS and pruning does not threshold as strongly by default which explains the high SHD of Table 3.5. To test this explanation, we removed all edges (i, j) for which $(\boldsymbol{A}_\phi)_{ij} < 0.3$[3] for GraN-DAG and obtained an SHD of 29.4±15.9 and an SID of 85.6±45.7, showing a significant improvement over NOTEARS and DAG-GNN, even without PNS and pruning.

[3]This was the default value of thresholding used in NOTEARS and DAG-GNN.

Algorithm	PNS	Pruning	SHD	SID	Score (train)	Score (validation)
DAG-GNN	False	False	56.8±11.1	322.9±103.8	-2.8±1.5	-2.2±1.6
	True	False	55.5±10.2	314.5±107.6	-2.1±1.6	-2.1±1.7
	False	True	49.4±7.8	325.1±103.7	-1.8±1.1	-1.8±1.2
	True	True	47.7±7.3	316.5±105.6	-1.9±1.6	-1.9±1.6
NOTEARS	False	False	64.2±9.5	327.1±110.9	-23.1±1.8	-23.2±2.1
	True	False	54.1±10.9	321.5±104.5	-25.2±2.7	-25.4±2.8
	False	True	49.5±8.8	327.7±111.3	-26.7±2.0	-26.8±2.1
	True	True	49.0±7.6	326.4±106.9	-26.23±2.2	-26.4±2.4

Table 3.6. PNS and pruning ablation study for DAG-GNN and NOTEARS (averaged over 10 datasets from ER1 with 50 nodes)

D. Large Sample Size Experiment

In this section, we test the bias-variance hypothesis which attempts to explain why CAM is on par with GraN-DAG on Gauss-ANM data even if its model wrongly assumes that the f_j functions are additive. Table 3.7 reports the performance of GraN-DAG and CAM for different sample sizes. We can see that, as the sample size grows, GraN-DAG ends up outperforming CAM in terms of SID while staying on par in terms of SHD. We explain this observation by the fact that a larger sample size reduces variance for GraN-DAG thus allowing it to leverage its greater capacity against CAM which is stuck with its modelling bias. Both algorithms were run with their respective default hyperparameter combination.

This experiment suggests GraN-DAG could be an appealing option in settings where the sample size is substantial. The present paper focuses on sample sizes typically encountered in the structure/causal learning litterature and leave this question for future work.

Sample size	Method	SHD	SID
500	CAM	123.5 ± 13.9	1181.2 ± 160.8
	GraN-DAG	130.2 ± 14.4	1246.4 ± 126.1
1000	CAM	103.7 ± 15.2	1074.7 ± 125.8
	GraN-DAG	104.4 ± 15.3	942.1 ± 69.8
5000	CAM	74.1 ± 13.2	845.0 ± 159.8
	GraN-DAG	71.9 ± 15.9	554.1 ± 117.9
10000	CAM	66.3 ± 16.0	808.1 ± 142.9
	GraN-DAG	65.9 ± 19.8	453.4 ± 171.7

Table 3.7. Effect of sample size - Gauss-ANM 50 nodes ER4 (averaged over 10 datasets)

E. Details on data sets generation

Synthetic data sets: For each data set type, 10 data sets are sampled with 1000 examples each. As the synthetic data introduced in Section 3.4.1, for each data set, a ground truth DAG $\mathcal{G}$ is randomly sampled following the ER scheme and then the data is generated. Unless otherwise stated, all root variables are sampled from $\mathcal{U}[-1, 1]$.

- *Gauss-ANM* is generated following $\boldsymbol{x}_j := f_j(\boldsymbol{x}_{\pi_j^{\mathcal{G}}}) + n_j \; \forall j$ with mutually independent noises $n_j \sim \mathcal{N}(0, \sigma_j^2) \;\; \forall j$ where the functions f_j are independently sampled from a Gaussian process with a unit bandwidth RBF kernel and $\sigma_j^2 \sim \mathcal{U}[0.4, 0.8]$. Source nodes are Gaussian with zero mean and variance sampled from $\mathcal{U}[1, 2]$
- *LIN* is generated following $\boldsymbol{x}_j | \boldsymbol{x}_{\pi_j^{\mathcal{G}}} \sim \boldsymbol{w}_j^T \boldsymbol{x}_{\pi_j^{\mathcal{G}}} + 0.2 \cdot \mathcal{N}(0, \sigma_j^2) \;\; \forall j$ where $\sigma_j^2 \sim \mathcal{U}[1, 2]$ and $\boldsymbol{w}_j$ is a vector of $|\pi_j^{\mathcal{G}}|$ coefficients each sampled from $\mathcal{U}[0, 1]$.
- *ADD-FUNC* is generated following $\boldsymbol{x}_j | \boldsymbol{x}_{\pi_j^{\mathcal{G}}} \sim \sum_{i \in \pi_j^{\mathcal{G}}} f_{j,i}(\boldsymbol{x}_i) + 0.2 \cdot \mathcal{N}(0, \sigma_j^2) \;\; \forall j$ where $\sigma_j^2 \sim \mathcal{U}[1, 2]$ and the functions $f_{j,i}$ are independently sampled from a Gaussian process with bandwidth one. This model is adapted from Bühlmann et al. [2014].
- *PNL-GP* is generated following $\boldsymbol{x}_j | \boldsymbol{x}_{\pi_j^{\mathcal{G}}} \sim \sigma(f_j(\boldsymbol{x}_{\pi_j^{\mathcal{G}}}) + Laplace(0, l_j)) \;\; \forall j$ with the functions f_j independently sampled from a Gaussian process with bandwidth one and $l_j \sim \mathcal{U}[0, 1]$. In the two-variable case, this model is identifiable following the Corollary 9 from Zhang and Hyvärinen [2009]. To get identifiability according to this corollary, it is important to use non-Gaussian noise, explaining our design choices.
- *PNL-MULT* is generated following $\boldsymbol{x}_j | \boldsymbol{x}_{\pi_j^{\mathcal{G}}} \sim \exp(\log(\sum_{i \in \pi_j^{\mathcal{G}}} \boldsymbol{x}_i) + |\mathcal{N}(0, \sigma_j^2)|) \;\; \forall j$ where $\sigma_j^2 \sim \mathcal{U}[0, 1]$. Root variables are sampled from $\mathcal{U}[0, 2]$. This model is adapted from Zhang et al. [2015].

SynTReN: Ten datasets have been generated using the SynTReN generator (`http://bioinformatics.intec.ugent.be/kmarchal/SynTReN/index.html`) using the software default parameters except for the *probability for complex 2-regulator interactions* that was set to 1 and the random seeds used were 0 to 9. Each dataset contains 500 samples and comes from a 20 nodes graph.

Graph types: *Erdős-Rényi* (ER) graphs are generated by randomly sampling a topological order and by adding directed edges were it is allowed independently with probability $p = \frac{2e}{d^2 - d}$ were e is the expected number of edges in the resulting DAG. *Scale-free* (SF) graphs were generated using the Barabási-Albert model [Barabási and Albert, 1999] which is based on preferential attachment. Nodes are added one by one. Between the new node and the existing nodes, m edges (where m is equal to d or $4d$) will be added. An existing node i have the probability $p(k_i) = \frac{k_i}{\sum_j k_j}$ to be chosen, where k_i represents the degree of the node i. While ER graphs have a degree distribution following a Poisson distribution, SF graphs have a degree distribution following a power law: few nodes, often called *hubs*, have a high degree. Barabási [2009] have stated that these types of graphs have

similar properties to real-world networks which can be found in many different fields, although these claims remain controversial [Clauset et al., 2009].

F. Supplementary experiments

Gauss-ANM: The results for 20 and 100 nodes are presented in Table 3.8 and 3.9 using the same Gauss-ANM data set types introduced in Section 3.4.1. The conclusions drawn remains similar to the 10 and 50 nodes experiments. For GES and PC, the SHD and SID are respectively presented in Table 3.10 and 3.11. Their performances do not compare favorably to the GraN-DAG nor CAM. Figure 3.1 shows the entries of the weighted adjacency matrix $\boldsymbol{A}_\phi$ as training proceeds in a typical run for 10 nodes.

LIN & ADD-FUNC: Experiments with LIN and ADD-FUNC data is reported in Table 3.12 & 3.13. The details of their generation are given in Appendix E.

PNL-GP & PNL-MULT: Table 3.14 contains the performance of GraN-DAG and other baselines on post nonlinear data discussed in Section 3.4.1.

	ER1		ER4		SF1		SF4	
	SHD	SID	SHD	SID	SHD	SID	SHD	SID
GraN-DAG	**4.0 ±3.4**	**17.9±19.5**	**45.2±10.7**	**165.1±21.0**	**7.6±2.5**	28.8±10.4	**36.8±5.1**	**62.5±18.8**
DAG-GNN	25.6±7.5	109.1±53.1	75.0±7.7	344.8±17.0	19.5±1.8	60.1±12.8	49.5±5.4	115.2±33.3
NOTEARS	30.3±7.8	107.3±47.6	79.0±8.0	346.6±13.2	23.9±3.5	69.4±19.7	52.0±4.5	120.5±32.5
CAM	**2.7±1.8**	**10.6±8.6**	**41.0±11.9**	**157.9±41.2**	**5.7±2.6**	**23.3±18.0**	**35.6±4.5**	**59.1±18.8**
GSF	12.3±4.6	[15.0±19.9 45.6±22.9]	**41.8±13.8**	**[153.7±49.4 201.6±37.9]**	**7.4±3.5**	**[5.7±7.1 27.3±13.2]**	**38.6±3.6**	**[54.9±14.4 86.7±24.2]**
RANDOM	103.0±39.6	94.3±53.0	117.5±25.9	298.5±28.7	105.2±48.8	81.1±54.4	121.5±28.5	204.8±38.5

Table 3.8. Results for ER and SF graphs of 20 nodes with Gauss-ANM data

	ER1		ER4		SF1		SF4	
	SHD	SID	SHD	SID	SHD	SID	SHD	SID
GraN-DAG	**15.1±6.0**	**83.9±46.0**	**206.6±31.5**	**4207.3±419.7**	**59.2±7.7**	**265.4±64.2**	**262.7±19.6**	**872.0±130.4**
DAG-GNN	110.2±10.5	883.0±320.9	379.5±24.7	8036.1±656.2	97.6±1.5	438.6±112.7	316.0±14.3	1394.6±165.9
NOTEARS	125.6±12.1	913.1±343.8	387.8±25.3	8124.7±577.4	111.7±5.4	484.3±138.4	327.2±15.8	1442.8±210.1
CAM	**17.3±4.5**	**124.9±65.0**	**186.4±28.8**	**4601.9±482.7**	**52.7±9.3**	**230.3±36.9**	**255.6±21.7**	**845.8±161.3**
GSF	66.8±7.3	[104.7±59.5 238.6±59.3]	$>$ 12 hours[4]	— —	**71.4±11.2**	**[212.7±71.1 325.3±105.2]**	**275.9±21.0**	**[793.9±152.5 993.4±149.2]**
RANDOM	1561.6±1133.4	1175.3±547.9	2380.9±1458.0	7729.7±1056.0	2222.2±1141.2	1164.2±593.3	2485.0±1403.9	4206.4±1642.1

Table 3.9. Results for ER and SF graphs of 100 nodes with Gauss-ANM data

[4]Note that GSF results are missing for two data set types in Tables 3.9 and 3.14. This is because the search algorithm could not finish within 12 hours, even when the maximal in-degree was limited to 5. All other methods could run in less than 6 hours.

	10 nodes		20 nodes		50 nodes		100 nodes	
	ER1	ER4	ER1	ER4	ER1	ER4	ER1	ER4
GraN-DAG	1.7±2.5	8.3±2.8	4.0 ±3.4	45.2±10.7	5.1±2.8	102.6±21.2	15.1±6.0	206.6±31.5
GES	13.8±4.8	32.3±4.3	43.3±12.4	94.6±9.8	106.6±24.7	254.4±39.3	292.9±33.6	542.6±51.2
PC	8.4±3	34±2.6	20.136.4±6.5	73.1±5.8	44.0±11.6	183.6±20	95.2±9.1	358.8±20.6
	SF1	SF4	SF1	SF4	SF1	SF4	SF1	SF4
GraN-DAG	1.2±1.1	9.9±4.0	7.6±2.5	36.8±5.1	25.5±6.2	111.3±12.3	59.2±7.7	262.7±19.6
GES	8.1±2.4	17.4±4.5	26.2±7.5	50.7±6.2	73.9±7.4	178.8±16.5	190.3±22	408.7±24.9
PC	4.8±2.4	16.4±2.8	13.6±2.1	44.4±4.6	43.1±5.7	135.4±10.7	97.6±6.6	314.2±17.5

Table 3.10. SHD for GES and PC (against GraN-DAG for reference) with Gauss-ANM data

	10 nodes		20 nodes		50 nodes		100 nodes	
	ER1	ER4	ER1	ER4	ER1	ER4	ER1	ER4
GraN-DAG	1.7±3.1	21.8±8.9	17.9±19.5	165.1±21.0	22.4±17.8	1060.1±109.4	83.9±46.0	4207.3±419.7
GES	[24.1±17.3 27.2±17.5]	[68.5±10.5 75±7]	[62.1±44 65.7±44.5]	[301.9±19.4 319.3±13.3]	[150.9±72.7 155.1±74]	[1996.6±73.1 2032.9±88.7]	[582.5±391.1 598.9±408.6]	[8054±524.8 8124.2±470.2]
PC	[22.6±15.5 27.3±13.1]	[78.1±7.4 79.2±5.7]	[80.9±51.1 94.9±46.1]	[316.7±25.7 328.7±25.6]	[222.7±138 256.7±127.3]	[2167.9±88.4 2178.8±80.8]	[620.7±240.9 702.5±255.8]	[8236.9±478.5 8265.4±470.2]
	SF1	SF4	SF1	SF4	SF1	SF4	SF1	SF4
GraN-DAG	4.1±6.1	16.4±6.0	28.8±10.4	62.5±18.8	90.0±18.9	271.2±65.4	265.4±64.2	872.0±130.4
GES	[11.6±9.2 16.4±11.7]	[39.3±11.2 43.9±14.9]	[54.9±23.1 57.9±24.6]	[89.5±38.4 105.1±44.3]	[171.6±70.1 182.7±77]	[496.3±154.1 529.7±184.5]	[511.5±257.6 524±252.2]	[1421.7±247.4 1485.4±233.6]
PC	[8.3±4.6 16.8±12.3]	[36.5±6.2 41.7±6.9]	[42.2±14 59.7±14.9]	[95.6±37 118.5±30]	[124.2±38.3 167.1±41.4]	[453.2±115.9 538±143.7]	[414.5±124.4 486.5±120.9]	[1369.2±259.9 1513.7±296.2]

Table 3.11. Lower and upper bound on the SID for GES and PC (against GraN-DAG for reference) with Gauss-ANM data. See Appendix G for details on how to compute SID for CPDAGs.

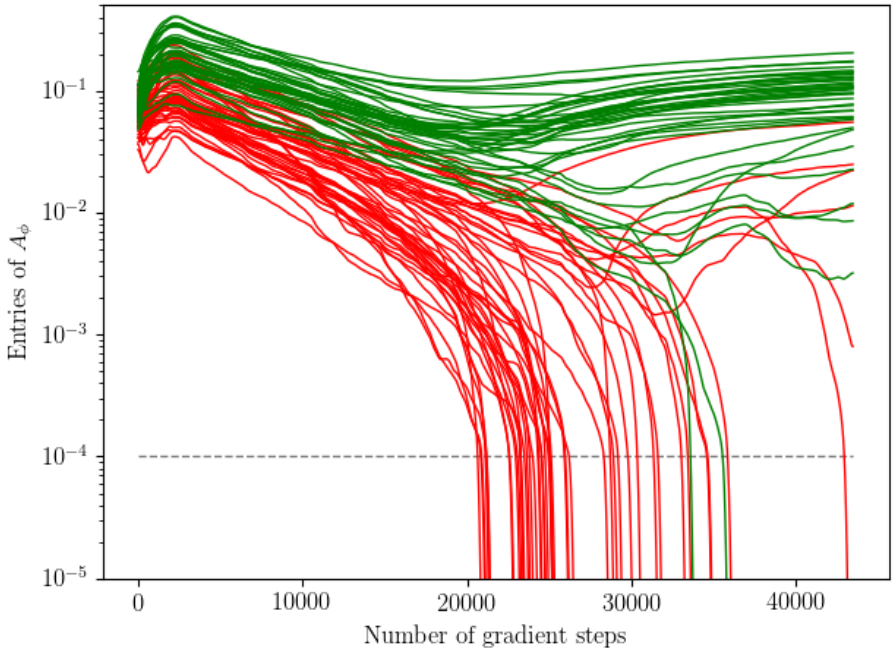


Figure 3.1. Entries of the weighted adjacency matrix $\boldsymbol{A}_\phi$ as training proceeds in GraN-DAG for a synthetic data set ER4 with 10 nodes. Green curves represent edges which appear in the ground truth graph while red ones represent edges which do not. The horizontal dashed line at 10^{-4} is the threshold ϵ introduced in Section 3.3.4. We can see that GraN-DAG successfully recovers most edges correctly while keeping few spurious edges.

#Nodes	10				50			
Graph Type	ER1		ER4		ER1		ER4	
Metrics	SHD	SID	SHD	SID	SHD	SID	SHD	SID
Method								
GraN-DAG	$\mathbf{7.2 \pm 2.0}$	27.3 ± 8.1	30.7 ± 3.3	75.8 ± 6.9	$\mathbf{33.9 \pm 8.6}$	$\mathbf{255.8 \pm 158.4}$	181.9 ± 24.0	2035.8 ± 137.2
DAG-GNN	10.3 ± 3.5	39.6 ± 14.7	$\mathbf{18.9 \pm 4.8}$	$\mathbf{63.7 \pm 8.9}$	54.1 ± 9.2	330.4 ± 117.1	$\mathbf{130.3 \pm 17.3}$	$\mathbf{1937.5 \pm 89.8}$
NOTEARS	$\mathbf{9.0 \pm 3.0}$	35.3 ± 13.4	27.9 ± 4.3	$\mathbf{72.1 \pm 7.9}$	45.5 ± 7.8	310.7 ± 125.9	$\mathbf{126.1 \pm 13.0}$	$\mathbf{1971.1 \pm 134.3}$
CAM	10.2 ± 6.3	31.2 ± 10.9	33.6 ± 3.3	77.5 ± 2.3	$\mathbf{36.2 \pm 5.8}$	$\mathbf{234.8 \pm 105.1}$	182.5 ± 17.6	$\mathbf{1948.7 \pm 113.5}$
GSF	9.2 ± 2.9	$[19.5 \pm 14.6$ $31.6 \pm 17.3]$	38.6 ± 3.7	$[73.8 \pm 7.6$ $85.2 \pm 8.3]$	46.7 ± 4.1	$[\mathbf{176.4 \pm 98.8}$ $\mathbf{215.0 \pm 108.9}]$	> 12 hours	
RANDOM	22.0 ± 2.9	30.0 ± 13.8	34.4 ± 2.4	78.8 ± 5.5	692.6 ± 7.5	360.3 ± 141.4	715.9 ± 16.0	$\mathbf{1932.7 \pm 40.2}$

Table 3.12. Experiments on LIN data

#Nodes	10				50			
Graph Type	ER1		ER4		ER1		ER4	
Metrics	SHD	SID	SHD	SID	SHD	SID	SHD	SID
Method								
GraN-DAG	$\mathbf{2.8 \pm 2.5}$	$\mathbf{7.5 \pm 7.7}$	14.5 ± 5.2	52.6 ± 10.8	16.6 ± 5.3	103.6 ± 52.9	86.4 ± 21.6	1320.6 ± 145.8
DAG-GNN	10.1 ± 3.4	23.3 ± 11.5	18.3 ± 3.6	56.4 ± 6.1	45.5 ± 7.9	261.1 ± 88.8	224.3 ± 31.6	1741.0 ± 138.3
NOTEARS	11.1 ± 5.0	16.9 ± 11.3	20.3 ± 4.9	53.5 ± 10.5	53.7 ± 9.5	276.1 ± 96.8	201.8 ± 22.1	1813.6 ± 148.4
CAM	$\mathbf{2.5 \pm 2.0}$	$\mathbf{7.9 \pm 6.4}$	$\mathbf{6.0 \pm 5.6}$	$\mathbf{29.3 \pm 19.3}$	$\mathbf{9.6 \pm 5.1}$	$\mathbf{39.0 \pm 34.1}$	$\mathbf{42.9 \pm 6.6}$	$\mathbf{857.0 \pm 184.5}$
GSF	9.3 ± 3.9	$[13.9 \pm 8.3$ $24.1 \pm 12.5]$	29.5 ± 4.3	$[60.3 \pm 11.6$ $75.0 \pm 4.5]$	49.5 ± 5.1	$[151.5 \pm 73.8$ $213.9 \pm 82.5]$	> 12 hours	
RANDOM	23.0 ± 2.2	26.9 ± 18.1	33.5 ± 2.3	76.0 ± 6.2	689.7 ± 6.1	340.0 ± 113.6	711.5 ± 9.0	1916.2 ± 65.8

Table 3.13. Experiments on ADD-FUNC data

		PNL-GP		PNL-MULT	
		SHD	SID	SHD	SID
10 nodes ER1	GraN-DAG	**1.6±3.0**	**3.9±8.0**	**13.1±3.8**	35.7±12.3
	DAG-GNN	11.5±6.8	32.4±19.3	17.900±6.2	40.700±14.743
	NOTEARS	10.7±5.5	34.4±19.1	**14.0±4.0**	38.6±11.9
	CAM	**1.5±2.6**	**6.8±12.1**	**12.0±6.4**	36.3±17.7
	GSF	**6.2±3.3**	[7.7±8.7, 18.9±12.4]	**10.7±3.0**	**[9.8±11.9, 25.3±11.5]**
	RANDOM	23.8±2.9	36.8±19.1	23.7±2.9	37.7±20.7
10 nodes ER4	GraN-DAG	**10.9±6.8**	**39.8±21.1**	32.1±4.5	77.7±5.9
	DAG-GNN	32.3±4.3	75.8±9.3	37.0±3.1	82.7±6.4
	NOTEARS	34.1±3.2	80.8±5.5	37.7±3.0	81.700±7.258
	CAM	**8.4±4.8**	**30.5±20.0**	34.4±3.9	79.6±3.8
	GSF	25.0±6.0	[44.3±14.5, 66.1±10.1]	31.3±5.4	**[58.6±8.1, 76.4±9.9]**
	RANDOM	35.0±3.3	80.0±5.1	33.6±3.5	76.2±7.3
50 nodes ER1	GraN-DAG	16.5±7.0	64.1±35.4	**38.2±11.4**	213.8±114.4
	DAG-GNN	56.5±11.1	334.3±80.3	83.9±23.8	507.7±253.4
	NOTEARS	50.1±9.9	319.1±76.9	78.5±21.5	425.7±197.0
	CAM	**5.1±2.6**	**10.7±12.4**	**44.9±9.9**	284.3±124.9
	GSF	31.2±6.0	[59.5±34.1, 122.4±32.0]	**46.3±12.1**	**[65.8±62.2, 141.6±72.6]**
	RANDOM	688.4±4.9	307.0±98.5	691.3±7.3	488.0±247.8
50 nodes ER4	GraN-DAG	**68.7±17.0**	**1127.0±188.5**	**211.7±12.6**	2047.7±77.7
	DAG-GNN	203.8±18.9	2173.1±87.7	246.7±16.1	2239.1±42.3
	NOTEARS	189.5±16.0	2134.2±125.6	**220.0±9.9**	2175.2±58.3
	CAM	**48.2±10.3**	**899.5±195.6**	**208.1±14.8**	2029.7±55.4
	GSF	105.2±15.5	[1573.7±121.2, 1620±102.8]	> 12 hours	—
	RANDOM	722.3±9.0	1897.4±83.7	710.2±9.5	1935.8±56.9

Table 3.14. Synthetic post nonlinear data sets

G. Metrics

SHD takes two partially directed acyclic graphs (PDAG) and counts the number of edge for which the edge type differs in both PDAGs. There are four edge types: $i \leftarrow j$, $i \rightarrow j$, $i - j$ and $i \quad j$. Since this distance is defined over the space of PDAGs, we can use it to compare DAGs with DAGs, DAGs with CPDAGs and CPDAGs with CPDAGs. When comparing a DAG with a CPDAG, having $i \leftarrow j$ instead of $i - j$ counts as a mistake.

SHD-C is very similar to SHD. The only difference is that both DAGs are first mapped to their respective CPDAGs before measuring the SHD.

Introduced by Peters and Bühlmann [2015], SID counts the number of interventional distribution of the form $p(\boldsymbol{x}_i|\ do(\boldsymbol{x}_j = \hat{\boldsymbol{x}}_j))$ that would be miscalculated using the *parent adjustment formula* [Pearl, 2009a] if we were to use the predicted DAG instead of the ground truth DAG to form the parent adjustment set. Some care needs to be taken to evaluate the SID for methods outputting a CPDAG such as GES and PC. Peters and Bühlmann [2015] proposes to report the SID of the DAGs which have approximately the minimal and the maximal SID in the Markov equivalence class given by the CPDAG. See Peters and Bühlmann [2015] for more details.

H. Hyperparameters

All GraN-DAG runs up to this point were performed using the following set of hyperparameters. We used RMSprop as optimizer with learning rate of 10^{-2} for the first subproblem and 10^{-4} for all subsequent suproblems. Each NN has two hidden layers with 10 units (except for the real and pseudo-real data experiments of Table 3.3 which uses only 1 hidden layer). Leaky-ReLU is used as activation functions. The NN are initialized using the initialization scheme proposed in Glorot and Bengio [2010a] also known as *Xavier initialization*. We used minibatches of 64 samples. This hyperparameter combination have been selected via a small scale experiment in which many hyperparameter combinations have been tried manually on a single data set of type ER1 with 10 nodes until one yielding a satisfactory SHD was obtained. Of course in practice one cannot select hyperparameters in this way since we do not have access to the ground truth DAG. In Appendix I, we explain how one could use a held-out data set to select the hyperparameters of score-based approaches and report the results of such a procedure on almost settings presented in this paper.

For NOTEARS, DAG-GNN, and GSF, we used the default hyperparameters found in the authors code. It (rarely) happens that NOTEARS and DAG-GNN returns a cyclic graph. In those cases, we removed edges starting from the weaker ones to the strongest (according to their respective weighted adjacency matrices), stopping as soon as acyclicity is achieved (similarly to what was explained in Appendix B for GraN-DAG). For GES and PC, we used default hyperparameters of the `pcalg` R package. For CAM, we used the the default hyperparameters found in the `CAM` R package, with default PNS and DAG pruning.

I. Hyperparameter Selection via Held-out Score

Most structure/causal learning algorithms have hyperparameters which must be selected prior to learning. For instance, NOTEARS and GES have a regularizing term in their score controlling the sparsity level of the resulting graph while CAM has a thresholding level for its pruning phase (also controlling the sparsity of the DAG). GraN-DAG and DAG-GNN have many hyperparameters such

as the learning rate and the architecture choice for the neural networks (i.e. number of hidden layers and hidden units per layer). One approach to selecting hyperparameters in practice consists in trying multiple hyperparameter combinations and keeping the one yielding the best score evaluated on a held-out set [Koller and Friedman, 2009, p. 960]. By doing so, one can hopefully avoid finding a DAG which is too dense or too sparse since if the estimated graph contains many spurious edges, the score on the held-out data set should be penalized. In the section, we experiment with this approach on almost all settings and all methods covered in the present paper.

Experiments: We explored multiple hyperparameter combinations using random search [Bergstra and Bengio, 2012]. Table 3.15 to Table 3.23 report results for each dataset types. Each table reports the SHD and SID averaged over 10 data sets and for each data set, we tried 50 hyperparameter combinations sampled randomly (see Table 3.24 for sampling schemes). The hyperparameter combination yielding the best held-out score among all 50 runs is selected *per data set* (i.e. the average of SHD and SID scores correspond to potentially different hyperparameter combinations on different data sets). 80% of the data was used for training and 20% was held out (GraN-DAG uses the same data for early stopping and hyperparameter selection). Note that the held-out score is always evaluated without the regularizing term (e.g. the held-out score of NOTEARS was evaluated without its L1 regularizer).

The symbols $^{++}$ and $^{+}$ indicate the hyperparameter search improved performance against default hyperparameter runs above one standard deviation and within one standard deviation, respectively. Analogously for $^{--}$ and $^{-}$ which indicate a performance reduction. The flag $_{***}$ indicate that, on average, less than 10 hyperparameter combinations among the 50 tried allowed the method to converge in less than 12 hours. Analogously, $_{**}$ indicates between 10 and 25 runs converged and $_{*}$ indicates between 25 and 45 runs converged.

Discussion: GraN-DAG and DAG-GNN are the methods benefiting the most from the hyperparameter selection procedure (although rarely significantly). This might be explained by the fact that neural networks are in general very sensitive to the choice of hyperparameters. However, not all methods improved their performance and no method improves its performance in all settings. GES and GSF for instance, often have significantly worse results. This might be due to some degree of model misspecification which renders the held-out score a poor proxy for graph quality. Moreover, for some methods the gain from the hyperparameter tuning might be outweighed by the loss due to the 20% reduction in training samples.

Additional implementation details for held-out score evaluation: GraN-DAG makes use of a final pruning step to remove spurious edges. One could simply mask the inputs of the NN corresponding to removed edges and evaluate the held-out score. However, doing so yields an unrepresentative score since some masked inputs have an important role in the learned function and once these inputs are masked, the quality of the fit might greatly suffer. To avoid this, we retrained

the whole model from scratch on the training set with the masking fixed to the one recovered after pruning. Then, we evaluate the held-out score with this retrained architecture. During this retraining phase, the estimated graph is fixed, only the conditional densities are relearned. Since NOTEARS and DAG-GNN are not always guaranteed to return a DAG (although they almost always do), some extra thresholding might be needed as mentioned in Appendix H. Similarly to GraN-DAG's pruning phase, this step can seriously reduce the quality of the fit. To avoid this, we also perform a retraining phase for NOTEARS and DAG-GNN. The model of CAM is also retrained after its pruning phase prior to evaluating its held-out score.

Graph Type	ER1		ER4		SF1		SF4	
Metrics	SHD	SID	SHD	SID	SHD	SID	SHD	SID
Method								
GraN-DAG	$\mathbf{1.0 \pm 1.6}^{+}$	$\mathbf{0.4 \pm 1.3}^{++}$	$\mathbf{5.5 \pm 2.8}^{+}$	$\mathbf{9.7 \pm 8.0}^{++}$	$\mathbf{1.3 \pm 1.8}^{-}$	$\mathbf{3.0 \pm 3.4}^{+}$	$\mathbf{9.6 \pm 4.5}^{+}$	$\mathbf{15.1 \pm 6.1}^{+}$
DAG-GNN	$10.9 \pm 2.6^{+}$	$35.5 \pm 13.6^{+}$	$38.3 \pm 2.9^{--}$	$84.4 \pm 3.5^{-}$	$9.9 \pm 1.7^{+}$	$30.3 \pm 18.8^{-}$	$21.4 \pm 2.1^{-}$	$44.0 \pm 15.5^{+}$
NOTEARS	$26.7 \pm 6.9^{--}$	$35.2 \pm 10.6^{+}$	$20.9 \pm 6.6^{++}$	$62.0 \pm 6.7^{++}$	$20.4 \pm 9.6^{--}$	$38.8 \pm 16.7^{-}$	$26.9 \pm 7.4^{-}$	$61.1 \pm 13.8^{-}$
CAM	$3.0 \pm 4.2^{-}$	$2.2 \pm 5.7^{-}$	$\mathbf{7.7 \pm 3.1}^{++}$	$23.2 \pm 14.7^{+}$	$\mathbf{2.4 \pm 2.5}^{-}$	$\mathbf{5.2 \pm 5.5}^{+}$	$\mathbf{9.6 \pm 3.1}^{+}$	$\mathbf{20.1 \pm 6.8}^{-}$
GSF	$5.3 \pm 3.3^{+}$	$[8.3 \pm 13.2^{+}$ $15.4 \pm 13.5]$	$23.1 \pm 7.9^{-}$	$[56.1 \pm 20.4^{-}$ $65.1 \pm 19.3]$	$3.3 \pm 2.5^{-}$	$[7.0 \pm 11.6^{-}$ $12.2 \pm 11.0]$	$14.2 \pm 5.6^{--}$	$[26.2 \pm 11.1^{-}$ $36.9 \pm 21.6]$
GES	$38.6 \pm 2.1^{--}$	$[20.3 \pm 15.4^{+}$ $28.3 \pm 18.4]$	$33.0 \pm 3.4^{-}$	$[66.2 \pm 7.0^{+}$ $76.6 \pm 4.3]$	$38.3 \pm 2.4^{--}$	$[8.8 \pm 5.2^{-}$ $25.5 \pm 18.2]$	$33.6 \pm 4.8^{--}$	$[32.7 \pm 12.7^{-}$ $52.0 \pm 14.0]$

Table 3.15. Gauss-ANM - 10 nodes with hyperparameter search

Graph Type	ER1		ER4		SF1		SF4	
Metrics	SHD	SID	SHD	SID	SHD	SID	SHD	SID
Method								
GraN-DAG	$\mathbf{3.8 \pm 3.3}^{+}$	$\mathbf{15.0 \pm 14.0}^{+}$	$\mathbf{105.6 \pm 16.5}^{-}$	$\mathbf{1131.7 \pm 91.0}^{-}$	$\mathbf{24.7 \pm 6.4}^{+}$	$\mathbf{86.5 \pm 34.6}^{+}$	$\mathbf{112.7 \pm 15.5}^{-}$	$\mathbf{268.3 \pm 85.8}^{+}$
DAG-GNN	$47.0 \pm 7.8^{+}$	$268.1 \pm 118.0^{+}$	$196.2 \pm 14.4^{-}$	$1972.8 \pm 110.6^{++}$	$51.8 \pm 5.6^{-}$	$166.5 \pm 48.9^{+}$	$144.2 \pm 11.6^{+}$	$473.4 \pm 105.4^{+}$
NOTEARS	$193.5 \pm 77.3^{--}$	$326.0 \pm 99.1^{+}$	$369.5 \pm 81.9^{--}$	$2062.0 \pm 107.7^{+}$	$104.8 \pm 22.4^{--}$	$290.3 \pm 136.8^{-}$	$213.0 \pm 35.1^{--}$	$722.7 \pm 177.3^{-}$
CAM	$\mathbf{4.0 \pm 2.7}^{+}$	$\mathbf{21.1 \pm 22.1}^{+}$	$\mathbf{105.6 \pm 20.9}^{-}$	$1225.9 \pm 205.7^{-}$	$\mathbf{23.8 \pm 6.0}^{+}$	$\mathbf{81.5 \pm 15.3}^{+}$	$\mathbf{112.2 \pm 14.0}^{-}$	$\mathbf{333.8 \pm 156.0}^{-}$
GSF	$24.9 \pm 7.4^{+}_{*}$	$[40.0 \pm 26.3^{-}_{*}$ $77.5 \pm 45.3]$	$129.3 \pm 20.4^{--}_{*}$	$[1280.8 \pm 202.3^{--}_{*}$ $1364.1 \pm 186.7]$	$35.3 \pm 6.9^{-}_{*}$	$[99.7 \pm 41.7^{-}_{*}$ $151.9 \pm 59.7]$	$\mathbf{121.6 \pm 11.7}^{-}_{***}$	$[\mathbf{310.8 \pm 108.1}^{-}_{***}$ $\mathbf{391.9 \pm 93.3}]$
GES	$1150.1 \pm 9.8^{--}$	$[112.7 \pm 71.1^{+}$ $132.0 \pm 89.0]$	$1066.1 \pm 11.7^{--}$	$[1394.3 \pm 81.8^{++}$ $1464.8 \pm 63.8]$	$1161.7 \pm 7.0^{--}$	$[322.8 \pm 211.1^{-}$ $336.0 \pm 215.4]$	$1116.1 \pm 14.2^{--}$	$[1002.7 \pm 310.9^{--}$ $1094.0 \pm 345.1]$

Table 3.16. Gauss-ANM - 50 nodes with hyperparameter search

Graph Type	ER1		ER4		SF1		SF4	
Metrics	SHD	SID	SHD	SID	SHD	SID	SHD	SID
Method								
GraN-DAG	$\mathbf{2.7 \pm 2.3}^{+}$	$\mathbf{9.6 \pm 10.3}^{+}$	$\mathbf{35.9 \pm 11.8}^{+}$	$\mathbf{120.4 \pm 37.0}^{++}$	$\mathbf{6.5 \pm 2.4}^{+}$	$\mathbf{17.5 \pm 6.3}^{++}$	$\mathbf{35.6 \pm 4.1}^{+}$	$\mathbf{54.8 \pm 14.3}^{+}$
DAG-GNN	$21.0 \pm 6.1^{+}$	$98.8 \pm 42.2^{+}$	$77.2 \pm 6.5^{-}$	$345.6 \pm 18.6^{-}$	$19.1 \pm 0.7^{+}$	$55.0 \pm 20.1^{+}$	$50.2 \pm 5.4^{-}$	$118.7 \pm 33.2^{-}$
NOTEARS	$101.5 \pm 39.6^{--}$	$100.4 \pm 47.0^{+}$	$124.0 \pm 16.3^{--}$	$267.0 \pm 46.5^{++}$	$55.0 \pm 28.2^{--}$	$87.6 \pm 26.9^{-}$	$66.7 \pm 8.3^{--}$	$154.6 \pm 43.0^{-}$
CAM	$\mathbf{2.8 \pm 2.2}^{-}$	$\mathbf{11.5 \pm 10.2}^{-}$	$64.3 \pm 29.3^{-}$	$\mathbf{121.7 \pm 73.1}^{+}$	$\mathbf{5.5 \pm 1.6}^{+}$	$\mathbf{19.3 \pm 7.8}^{+}$	$\mathbf{36.0 \pm 5.1}^{-}$	$\mathbf{66.3 \pm 28.6}^{-}$
GSF	$11.6 \pm 3.0^{+}$	$[26.4 \pm 13.3^{-}$ $49.8 \pm 26.5]$	$\mathbf{46.2 \pm 12.6}^{-}$	$[172.7 \pm 40.8^{-}$ $213.5 \pm 38.6]$	$12.8 \pm 2.1^{--}$	$[32.1 \pm 14.0^{--}$ $56.2 \pm 13.8]$	$42.3 \pm 5.1^{-}$	$[68.9 \pm 27.7^{-}$ $95.1 \pm 33.8]$
GES	$169.9 \pm 5.0^{--}$	$[45.4 \pm 29.2^{+}$ $57.2 \pm 36.6]$	$142.8 \pm 7.7^{--}_{*}$	$[223.3 \pm 33.6^{++}_{*}$ $254.7 \pm 22.0]$	$168.1 \pm 3.3^{--}$	$[46.7 \pm 21.7^{+}$ $53.3 \pm 20.0]$	$162.2 \pm 10.4^{--}$	$[151.1 \pm 57.4^{--}$ $195.8 \pm 57.4]$

Table 3.17. Gauss-ANM - 20 nodes with hyperparameter search

Graph Type	ER1		ER4		SF1		SF4	
Metrics	SHD	SID	SHD	SID	SHD	SID	SHD	SID
Method								
GraN-DAG	$\mathbf{15.1 \pm 7.5}^{+}$	$\mathbf{65.1 \pm 33.2}^{+}$	$\mathbf{191.6 \pm 17.8}^{+}$	$\mathbf{4090.7 \pm 418.0}^{+}$	$\mathbf{51.6 \pm 10.2}^{+}$	$\mathbf{210.6 \pm 51.9}^{++}$	$\mathbf{255.7 \pm 21.1}^{+}$	$\mathbf{790.5 \pm 159.7}^{+}$
DAG-GNN	$103.9 \pm 9.1^{+}$	$757.6 \pm 215.0^{+}$	$387.1 \pm 25.3^{-}$	$7741.9 \pm 522.5^{+}$	$103.5 \pm 8.2^{-}$	$391.7 \pm 60.0^{+}$	$314.8 \pm 16.3^{+}$	$1257.3 \pm 185.2^{+}$
NOTEARS	$421.3 \pm 207.0^{--}$	$945.7 \pm 339.7^{-}$	$631.1 \pm 136.6^{--}$	$8272.4 \pm 444.2^{-}$	$244.3 \pm 63.8^{--}$	$815.6 \pm 346.5^{-}$	$482.3 \pm 114.1^{--}$	$1929.7 \pm 363.1^{--}$
CAM	$\mathbf{12.3 \pm 4.9}^{++}$	$128.0 \pm 66.3^{-}$	$\mathbf{198.8 \pm 22.2}^{-}$	$4602.2 \pm 523.7^{-}$	$\mathbf{51.1 \pm 9.4}^{+}$	$\mathbf{233.6 \pm 62.3}^{-}$	$\mathbf{255.7 \pm 22.2}^{-}$	$\mathbf{851.4 \pm 206.0}^{-}$
GSF	$100.2 \pm 9.9^{--}_{**}$	$[719.8 \pm 242.1^{--}_{**}$ $721.1 \pm 242.9]$	$387.6 \pm 23.9_{***}$	$[7535.1 \pm 595.2_{***}$ $7535.1 \pm 595.2]$	$67.3 \pm 14.0^{+}_{***}$	$[254.5 \pm 35.4^{-}_{***}$ $340.4 \pm 70.4]$	$315.1 \pm 16.7^{--}_{***}$	$[1214.0 \pm 156.4^{--}_{***}$ $1214.0 \pm 156.4]$
GES	$4782.5 \pm 22.9^{--}$	$[362.3 \pm 267.7^{+}$ $384.1 \pm 293.6]$	$4570.1 \pm 27.9^{--}$	$[5400.7 \pm 299.2^{++}$ $5511.5 \pm 308.5]$	$4769.1 \pm 26.7^{--}$	$[1311.1 \pm 616.6^{--}$ $1386.2 \pm 713.9]$	$4691.3 \pm 47.3^{--}$	$[3882.7 \pm 1010.6^{--}$ $3996.7 \pm 1075.7]$

Table 3.18. Gauss-ANM - 100 nodes with hyperparameter search

#Nodes	10				50			
Graph Type	ER1		ER4		ER1		ER4	
Metrics	SHD	SID	SHD	SID	SHD	SID	SHD	SID
Method								
GraN-DAG	$\mathbf{1.2 \pm 2.2}^{+}$	$\mathbf{1.9 \pm 4.2}^{+}$	$\mathbf{9.8 \pm 4.9}^{+}$	$\mathbf{29.0 \pm 17.6}^{+}$	$12.8 \pm 4.9^{+}$	$55.3 \pm 24.2^{+}$	$73.9 \pm 16.8^{-}$	$\mathbf{1107.2 \pm 144.7}^{+}$
DAG-GNN	$10.6 \pm 4.9^{+}$	$35.8 \pm 19.6^{-}$	$38.6 \pm 2.0^{--}$	$82.2 \pm 5.7^{--}$	$48.1 \pm 8.4^{+}$	$330.4 \pm 69.9^{+}$	$192.5 \pm 19.2^{+}$	$2079.5 \pm 120.9^{+}$
NOTEARS	$20.6 \pm 11.4^{-}$	$30.5 \pm 18.8^{+}$	$24.2 \pm 6.5^{++}$	$66.4 \pm 6.9^{++}$	$102.1 \pm 27.3^{--}$	$299.8 \pm 85.8^{+}$	$660.0 \pm 258.2^{--}$	$1744.0 \pm 232.9^{++}$
CAM	$\mathbf{2.7 \pm 4.0}^{-}$	$6.4 \pm 11.8^{+}$	$\mathbf{8.7 \pm 4.5}^{-}$	$\mathbf{30.9 \pm 20.4}^{-}$	$\mathbf{4.0 \pm 2.4}^{+}$	$\mathbf{10.7 \pm 12.4}^{+}$	$\mathbf{52.3 \pm 8.5}^{-}$	$\mathbf{913.9 \pm 209.3}^{-}$
GSF	$12.9 \pm 3.9^{--}_{***}$	$[10.5 \pm 8.7^{--}_{***}$ $53.6 \pm 23.8]$	$40.7 \pm 1.3^{--}_{**}$	$[79.2 \pm 3.8^{--}_{**}$ $79.2 \pm 3.8]$	$48.8 \pm 3.9^{--}_{**}$	$[281.6 \pm 70.7^{--}_{**}$ $281.6 \pm 70.7]$	$199.9 \pm 15.2^{--}_{***}$	$[1878.0 \pm 122.4^{--}_{***}$ $1948.4 \pm 139.6]$

Table 3.19. PNL-GP with hyperparameter search

#Nodes	10				50			
Graph Type	ER1		ER4		ER1		ER4	
Metrics	SHD	SID	SHD	SID	SHD	SID	SHD	SID
Method								
GraN-DAG	$\mathbf{10.0 \pm 4.5}^{+}$	$\mathbf{29.1 \pm 9.7}^{+}$	$\mathbf{32.9 \pm 3.3}^{-}$	$76.7 \pm 4.1^{+}$	$\mathbf{59.8 \pm 28.2}^{-}$	$\mathbf{213.6 \pm 97.3}^{+}$	$272.1 \pm 69.4^{-}$	$2021.6 \pm 185.8^{+}$
DAG-GNN	$14.6 \pm 3.1^{++}$	$\mathbf{36.9 \pm 10.6}^{+}$	$38.9 \pm 2.0^{-}$	$85.8 \pm 1.2^{--}$	$\mathbf{64.3 \pm 27.8}^{+}$	$508.8 \pm 317.2^{-}$	$\mathbf{212.5 \pm 12.3}^{++}$	$2216.9 \pm 95.6^{+}$
NOTEARS	$28.8 \pm 9.1^{--}$	$\mathbf{30.3 \pm 11.8}^{+}$	$\mathbf{35.4 \pm 3.8}^{+}$	$78.4 \pm 7.5^{+}$	$160.2 \pm 67.5^{--}$	$443.5 \pm 205.1^{-}$	$229.2 \pm 25.4^{-}$	$2158.8 \pm 70.3^{+}$
CAM	$17.2 \pm 8.0^{-}$	$\mathbf{33.7 \pm 14.4}^{+}$	$\mathbf{32.3 \pm 6.5}^{+}$	$76.6 \pm 8.2^{+}$	$97.5 \pm 71.1^{-}$	$\mathbf{282.3 \pm 123.8}^{+}$	$251.0 \pm 25.9^{--}$	$2026.2 \pm 58.2^{+}$
GSF	$15.6 \pm 4.4^{--}_{**}$	$[\mathbf{10.0 \pm 6.3}^{--}_{**}$ $\mathbf{60.1 \pm 17.2}]$	$39.3 \pm 2.2^{--}_{**}$	$[76.0 \pm 9.6^{--}_{**}$ $79.9 \pm 5.3]$	$\mathbf{66.4 \pm 14.4}^{--}_{***}$	$[145.1 \pm 96.1^{--}_{***}$ $618.8 \pm 257.0]$	> 12 hours	

Table 3.20. PNL-MULT with hyperparameter search

#Nodes	10				50			
Graph Type	ER1		ER4		ER1		ER4	
Metrics	SHD	SID	SHD	SID	SHD	SID	SHD	SID
Method								
GraN-DAG	$\mathbf{10.1 \pm 3.9}^{-}$	$\mathbf{28.7 \pm 14.7}^{-}$	$34.7 \pm 2.9^{--}$	$79.5 \pm 4.4^{-}$	$\mathbf{40.8 \pm 10.3}^{-}$	$\mathbf{236.3 \pm 101.7}^{+}$	$256.9 \pm 55.7^{--}$	$2151.4 \pm 144.3^{-}$
DAG-GNN	$\mathbf{9.0 \pm 2.7}^{++}$	$\mathbf{35.6 \pm 11.4}^{-}$	$\mathbf{19.6 \pm 4.6}^{+}$	$\mathbf{63.9 \pm 7.5}^{-}$	$48.3 \pm 6.8^{+}$	$381.7 \pm 145.4^{-}$	$\mathbf{149.7 \pm 17.2}^{++}$	$2070.7 \pm 51.9^{--}$
NOTEARS	$\mathbf{14.0 \pm 4.1}^{--}_{*}$	$\mathbf{32.2 \pm 7.9}^{+}_{*}$	$\mathbf{20.7 \pm 5.1}^{++}$	$\mathbf{63.1 \pm 8.0}^{++}$	$87.7 \pm 44.3^{-}_{*}$	$\mathbf{294.3 \pm 99.3}^{+}_{*}$	$200.3 \pm 67.1^{--}$	$\mathbf{1772.7 \pm 143.7}^{++}$
CAM	$\mathbf{8.8 \pm 6.0}^{+}$	$\mathbf{25.8 \pm 13.5}^{+}$	$33.9 \pm 2.8^{-}$	$77.1 \pm 4.5^{+}$	$\mathbf{34.8 \pm 7.0}^{+}$	$\mathbf{221.2 \pm 98.3}^{+}$	$202.2 \pm 14.3^{--}$	$1990.8 \pm 97.5^{-}$
GSF	$\mathbf{10.7 \pm 3.5}^{-}$	$[\mathbf{15.8 \pm 8.4}^{-}$ $\mathbf{45.2 \pm 20.2}]$	$33.4 \pm 3.3^{++}$	$[71.7 \pm 11.5^{+}$ $77.3 \pm 6.1]$	$54.4 \pm 6.5^{--}_{*}$	$[158.1 \pm 115.9^{-}_{*}$ $560.9 \pm 220.7]$	$195.6 \pm 9.9_{**}$	$[2004.9 \pm 85.2_{**}$ $2004.9 \pm 85.2]$

Table 3.21. LIN with hyperparameter search

#Nodes	10				50			
Graph Type	ER1		ER4		ER1		ER4	
Metrics	SHD	SID	SHD	SID	SHD	SID	SHD	SID
Method								
GraN-DAG	$\mathbf{2.6 \pm 2.4}^{+}$	$\mathbf{4.3 \pm 4.3}^{+}$	$\mathbf{7.0 \pm 3.1}^{++}$	$\mathbf{37.1 \pm 12.4}^{++}$	$\mathbf{13.2 \pm 6.7}^{+}$	$\mathbf{72.1 \pm 55.2}^{+}$	$90.1 \pm 25.6^{-}$	$1241.7 \pm 289.8^{+}$
DAG-GNN	$8.7 \pm 2.8^{++}$	$22.3 \pm 9.4^{+}$	$25.3 \pm 3.8^{++}$	$63.6 \pm 8.6^{++}$	$44.7 \pm 9.7^{++}$	$306.9 \pm 114.7^{+}$	$194.0 \pm 20.4^{+}$	$1949.3 \pm 107.1^{+}$
NOTEARS	$21.2 \pm 11.5^{-}_{*}$	$15.5 \pm 9.9^{+}_{*}$	$13.3 \pm 4.3^{++}$	$\mathbf{41.3 \pm 11.5}^{++}$	$186.8 \pm 83.0^{--}$	$276.9 \pm 92.1^{-}$	$718.4 \pm 170.4^{--}$	$1105.9 \pm 250.1^{++}$
CAM	$\mathbf{3.0 \pm 2.2}^{-}$	$\mathbf{8.1 \pm 6.3}^{-}$	$\mathbf{6.2 \pm 5.5}^{-}$	$\mathbf{28.5 \pm 21.5}^{+}$	$\mathbf{10.0 \pm 4.6}^{-}$	$\mathbf{44.2 \pm 32.1}^{-}$	$\mathbf{46.6 \pm 9.5}^{-}$	$\mathbf{882.5 \pm 186.5}^{-}$
GSF	$5.5 \pm 4.1^{+}$	$[7.5 \pm 12.3^{+}$ $16.3 \pm 12.9]$	$19.1 \pm 7.0^{++}$	$[44.5 \pm 19.7^{+}$ $60.4 \pm 16.5]$	$29.8 \pm 7.6^{++}_{*}$	$[\mathbf{44.6 \pm 42.6}^{++}_{*}$ $\mathbf{96.8 \pm 46.7}]$	$140.4 \pm 31.7_{***}$	$[1674.4 \pm 133.9_{***}$ $1727.6 \pm 145.2]$

Table 3.22. ADD-FUNC with hyperparameter search

Data Type	Protein signaling data set			SynTReN - 20 nodes		
Metrics	SHD	SHD-C	SID	SHD	SHD-C	SID
Method						
GraN-DAG	12.0^{+}	$\mathbf{9.0}^{+}$	48.0^{-}	$41.2 \pm 9.6^{-}$	$43.7 \pm 8.3^{-}$	$144.3 \pm 61.3^{+}$
GraN-DAG++	14.0^{-}	11.0^{-}	57.0^{-}	$46.9 \pm 14.9^{-}$	$49.5 \pm 14.7^{-}$	$158.4 \pm 61.5^{-}$
DAG-GNN	16.0	14.0^{+}	59.0^{-}	$\mathbf{32.2 \pm 5.0}^{++}$	$\mathbf{32.3 \pm 5.6}^{++}$	$194.2 \pm 50.2^{-}$
NOTEARS	15.0^{+}	14.0^{+}	58.0^{-}	$44.2 \pm 27.5^{++}$	$45.8 \pm 27.7^{++}$	$183.1 \pm 48.4^{--}$
CAM	$\mathbf{11.0}^{+}$	$\mathbf{9.0}$	51.0^{+}	$101.7 \pm 37.2^{--}$	$105.6 \pm 36.6^{--}$	$\mathbf{111.5 \pm 25.3}^{++}$
GSF	20.0^{-}	14.0^{-}	$[37.0^{+}$ $60.0]$	$\mathbf{27.8 \pm 5.4}^{++}_{*}$	$\mathbf{27.8 \pm 5.4}^{++}_{*}$	$[207.6 \pm 55.4^{--}_{*}$ $209.6 \pm 59.1]$
GES	47.0^{-}	50.0^{-}	$[\mathbf{37.0}^{+}$ $\mathbf{47.0}]$	$167.5 \pm 5.6^{--}$	$172.2 \pm 7.0^{--}$	$[\mathbf{75.3 \pm 24.4}^{++}$ $\mathbf{97.6 \pm 30.8}]$

Table 3.23. Results for real and pseudo real data sets with hyperparameter search

	Hyperparameter space
GraN-DAG	Log(learning rate) $\sim U[-2, -3]$ (first subproblem) Log(learning rate) $\sim U[-3, -4]$ (other subproblems) $\epsilon \sim U\{10^{-3}, 10^{-4}, 10^{-5}\}$ Log(pruning cutoff) $\sim U\{-5, -4, -3, -2, -1\}$ # hidden units $\sim U\{4, 8, 16, 32\}$ # hidden layers $\sim U\{1, 2, 3\}$ Constraint convergence tolerance $\sim U\{10^{-6}, 10^{-8}, 10^{-10}\}$ PNS threshold $\sim U[0.5, 0.75, 1, 2]$
DAG-GNN	Log(learning rate) $\sim U[-4, -2]$ # hidden units in encoder $\sim U\{16, 32, 64, 128, 256\}$ # hidden units in decoder $\sim U\{16, 32, 64, 128, 256\}$ Bottleneck dimension (dimension of Z) $\sim U\{1, 5, 10, 50, 100\}$ Constraint convergence tolerance $\sim U\{10^{-6}, 10^{-8}, 10^{-10}\}$
NOTEARS	L1 regularizer coefficient $\sim U\{0.001, 0.005, 0.01, 0.05, 0.1, 0.5\}$ Final threshold $\sim U\{0.001, 0.003, 0.01, 0.03, 0.1, 0.3, 1\}$ Constraint convergence tolerance $\sim U\{10^{-6}, 10^{-8}, 10^{-10}\}$
CAM	Log(Pruning cutoff) $\sim U[-6, 0]$
GSF	Log(RKHS regression regularizer) $\sim U[-4, 4]$
GES	Log(Regularizer coefficient) $\sim U[-4, 4]$

Table 3.24. Hyperparameter search spaces for each algorithm

Prologue to the Second Contribution

Article Details

Differentiable Causal Discovery from Interventional Data
by *Philippe Brouillard* *, *Sébastien Lachapelle* *, *Alexandre Lacoste, Simon Lacoste-Julien & Alexandre Drouin*. This work was published at the Thirty-fourth Conference on Neural Information Processing Systems (NeurIPS 2020) with a **spotlight**.
*Equal contributions.

Contributions of the Authors

Philippe Brouillard implemented the method, led the experiments, made an extensive literature review on structure learning methods using interventional data, integrated baseline methods in our experiment pipeline. **Sébastien Lachapelle** suggested the objective that DCDI maximizes, wrote the theory and proofs and led the scalability experiments. **Alexandre Lacoste** and **Simon Lacoste-Julien** contributed to the writing and provided supervision. **Alexandre Drouin** provided supervision, contributed to the writing, led large scale experiments with hyperparameter search and integrated the *deep sigmoidal flow* in the code base.

Context and Limitations

Differentiable causal discovery from interventional data (DCDI) can be thought of as a direct extension of GraN-DAG along two axes: (i) DCDI can leverage interventional data (which alleviates the problem of identifiability); and (ii) it can model much more complex causal dependencies than GraN-DAG, thanks to normalizing flows. We also highlighted an advantage of continuous constrained methods for DAG learning: they scale seamlessly with dataset size, especially when contrasted with constrained-based methods which rely on kernel-based independence tests.

Although optimization was improved by the usage of Gumbel-sigmoid masks to learn the causal graph (see also Ng et al. [2019], Ke et al. [2019] which were also using such masks) as opposed to GraN-DAG which was using a holistic constraint on the weights of the neural networks, optimization

remains perhaps the most important challenge. Computationally, DCDI suffers from the same issues as GraN-DAG, namely the cubic cost as a function of the number of variables. However, more recent developments have proposed solutions, which we discuss next.

Recent developments

Lopez et al. [2022] proposed differentiable causal discovery of factor graphs (DCD-FG), an extension of DCDI which scales to thousands of variables and showed it could beat the state of the art when predicting the effect of unseen perturbations in gene regulatory networks (also see Weinstock et al. [2023] for a similar application of causal discovery). The idea is to limit the space of graphs to some type of low-rank graphs they call factor-DAGs, which encodes the assumption that sets of variables tend to act together as parents of other variables. This constraint on the structure of the dependencies can be combined with the computationally cheaper algebraic characterization of acyclicity proposed by Lee et al. [2020] to get a cost of $O(md)$ for the evaluation of the gradient of the constraint, down from $O(d^3)$, where d is the number of nodes and m is the number of factors (which is picked so that $m << d$). This contribution illustrates the flexibility of gradient-based approaches to structure learning. Indeed, the neural networks used to model the conditional densities can be adapted to encode specific types of inductive biases that are suitable for the task at hand. In the case of DCD-FG, the low-rank assumption appears to be very well suited for learning a gene regulatory network, where genes are believed to act in group. Note that Fang et al. [2023] also explored low-rank assumptions on the weighted adjacency graph and proposed a low-rank variant of GraN-DAG. Different variations of NOTEARS [Zheng et al., 2018] and DCDI have been proposed, for instance SDCD [Nazaret et al., 2023], which proposed a more stable acyclicity constraint, and NODAGS-Flow [Guruswamy Sethuraman et al., 2023], which extends differentiable causal discovery to cyclic graphs.

The optimization of most differentiable causal discovery methods rely on the augmented Lagrangian (Section 2.6.1) to enforce the acyclicity constraint. The original motivation to use this approach in this context is that, the penalty term does not have to go to infinity in order to converge to a feasible solution, contrarily to the penalty method, a simpler alternative to the augmented Lagrangian [Zheng et al., 2018, Bertsekas, 1999]. In Ng et al. [2022], we clarified that, for this argument to hold, the constraint must satisfy some regularity condition which is not satisfied by the acyclicity constraints. We further show empirically that, in the differentiable causal discovery setting, the augmented Lagrangian method and the penalty method have very similar behaviors. A consequence of this observation is that, in order to converge to a feasible solution, the penalty coefficient must go to infinity, which might result in an ill-conditioned loss landscape making optimization especially challenging.

DCDI treats the estimated causal graph, which is discrete, as random to enable the use of gradient descent to optimize the parameters of its distribution, which are continuous. Even if this approach effectively learns a distribution over graphs, this should be thought of as a trick to allow for gradient-based optimization where the end goal is still to output a single causal graph, not a distribution over graphs (the distribution converges to a point mass on a single graph in practice). More recent works have proposed Bayesian approaches where one commits to a prior distribution over graphs, $p(\mathcal{G})$, and somehow estimates a posterior over graphs given a dataset, $p(\mathcal{G} \mid \mathcal{D})$, which quantifies our uncertainty about the graph, either due to limited data or lack of identifiability. For instances, Annadani et al. [2021] and [Lorch et al., 2021] propose variational methods to estimate the intractable posterior over graphs while Deleu et al. [2022] and Nishikawa-Toomey et al. [2023] rely on the recently proposed framework of generative flow networks (GFlowNets) [Bengio et al., 2023]. This class of approach seems particularly promising when integrated within an active learning loop which, based on the uncertainty of various edges, decides which intervention is more likely to reduce our uncertainty about the causal graph or a specific causal query of interest [Toth et al., 2022, Scherrer et al., 2022].

A whole other approach to causal discovery consists in training a black-box model to predict a causal graph from a dataset of observations [Lopez-Paz et al., 2015, Li et al., 2020, Wu et al., 2024], possibly with interventions [Ke et al., 2023]. These models are trained on synthetically generated datasets sampled from randomly generated causal models where the known causal graph can be used as a label for supervised learning. The methodology is sound: (i) choose assumptions you are willing to make about the ground-truth causal model, (ii) generate datasets sampled from causal models satisfying your assumptions, (iii) train a model to predict causal graphs from datasets and (iv) use that predictor as a causal discovery algorithm for new datasets. If the assumptions made in the first place are sufficient to have identifiability, the graph predictor should be able to predict the causal graph correctly, given it was trained on sufficiently many causal discovery tasks. This approach mirrors more classical discovery techniques: Choose assumptions and then design an algorithm that can leverage these assumptions to estimate a causal graph from (interventional) observations. These black-box supervised methods show a surprising ability to generalize to novel synthetic causal discovery tasks, with potentially different kinds of functional relationships, but it remains difficult to show that this is not due to the black-box predictor "picking up" on artifacts specific to how the synthetic data is generated. This is less of a concern for more standard causal discovery algorithms which are not "discovered" by training on generated dataset-graph pairs. Encouragingly, some of these studies have shown that these learned predictors significantly outperform more standard methods (including DCDI) on more realistic data such as Sachs et al. [2005] and the BnLearn repository [Elidan, 2001]. It will be exciting to see whether further studies on real-data will confirm this trend.

Chapter 4

Differentiable Causal Discovery from Interventional Data

Abstract

Learning a causal directed acyclic graph from data is a challenging task that involves solving a combinatorial problem for which the solution is not always identifiable. A new line of work reformulates this problem as a continuous constrained optimization one, which is solved via the augmented Lagrangian method. However, most methods based on this idea do not make use of interventional data, which can significantly alleviate identifiability issues. This work constitutes a new step in this direction by proposing a theoretically-grounded method based on neural networks that can leverage interventional data. We illustrate the flexibility of the continuous-constrained framework by taking advantage of expressive neural architectures such as normalizing flows. We show that our approach compares favorably to the state of the art in a variety of settings, including perfect and imperfect interventions for which the targeted nodes may even be unknown.

4.1. Introduction

The inference of causal relationships is a problem of fundamental interest in science. In all fields of research, experiments are systematically performed with the goal of elucidating the underlying causal dynamics of systems. This quest for causality is motivated by the desire to take actions that induce a controlled change in a system. Achieving this requires to answer questions, such as "what would be the impact on the system if this variable were changed from value x to y?", which cannot be answered without causal knowledge [Pearl, 2009b].

In this work, we address the problem of data-driven causal discovery [Heinze-Deml et al., 2018a]. Our goal is to design an algorithm that can automatically discover causal relationships from data. More formally, we aim to learn a *causal graphical model* (CGM) [Peters et al., 2017], which

consists of a joint distribution coupled with a directed acyclic graph (DAG), where edges indicate direct causal relationships. Achieving this based on observational data alone is challenging since, under the faithfulness assumption, the true DAG is only identifiable up to a *Markov equivalence class* [Verma and Pearl, 1990]. Fortunately, identifiability can be improved by considering interventional data, i.e., the outcome of some experiments. In this case, the DAG is identifiable up to an *interventional Markov equivalence class*, which is a subset of the Markov equivalence class [Yang et al., 2018, Hauser and Bühlmann, 2012], and, when observing enough interventions [Eberhardt, 2008, Eberhardt et al., 2005], the DAG is exactly identifiable. In practice, it may be possible for domain experts to collect such interventional data, resulting in clear gains in identifiability. For instance, in genomics, recent advances in gene editing technologies have given rise to high-throughput methods for interventional gene expression data [Dixit et al., 2016].

Nevertheless, even with interventional data at hand, finding the right DAG is challenging. The solution space is immense and grows super-exponentially with the number of variables. Recently, Zheng et al. [2018] proposed to cast this search problem as a constrained continuous-optimization problem, avoiding the computationally-intensive search typically performed by score-based and constraint-based methods [Peters et al., 2017]. The work of Zheng et al. [2018] was limited to linear relationships, but was quickly extended to nonlinear ones via neural networks [Lachapelle et al., 2020, Yu et al., 2019a, Zheng et al., 2020, Ng et al., 2019, Kalainathan et al., 2018, Zhu and Chen, 2020]. Yet, these approaches do not make use of interventional data and must therefore rely on strong parametric assumptions (e.g., gaussian additive noise models). Bengio et al. [2020] leveraged interventions and continuous optimization to learn the causal direction in the bivariate setting. The follow-up work of Ke et al. [2019] generalized to the multivariate setting by optimizing an unconstrained objective with regularization inspired by Zheng et al. [2018], but lacked theoretical guarantees. In this work, we propose a theoretically-grounded differentiable approach to causal discovery that can make use of *interventional* data (with potentially unknown targets) and that relies on the constrained-optimization framework of Zheng et al. [2018] without making strong assumptions about the functional form of causal mechanisms, thanks to expressive density estimators.

4.1.1. Contributions

- We propose Differentiable Causal Discovery with Interventions (DCDI): a general differentiable causal structure learning method that can leverage perfect, imperfect and unknown-target interventions (Section 4.3). We propose two instantiations, one of which is a universal density approximator that relies on normalizing flows (Section 4.3.4).
- We show that the exact maximization of the proposed score will identify the $\mathcal{I}$-Markov equivalence class [Yang et al., 2018] of the ground truth graph (under regularity conditions)

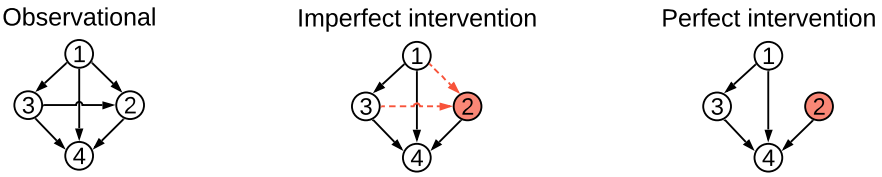


Figure 4.1. Different intervention types (shown in red). In imperfect interventions, the causal relationships are altered. In perfect interventions, the targeted node is cut out from its parents.

for both the known- and unknown-target settings (Thm. 4.1 in Section 4.3.1 & Thm. 4.2 in Section 4.3.3, respectively).

- We provide an extensive comparison of DCDI to state-of-the-art methods in a wide variety of conditions, including multiple functional forms and types of interventions (Section 4.4).

4.2. Background and related work

4.2.1. Definitions

Causal graphical models. A CGM is defined by a distribution $\mathbb{P}_{\boldsymbol{x}}$ over a random vector $\boldsymbol{x} = (\boldsymbol{x}_1, \cdots, \boldsymbol{x}_d)$ and a DAG $\mathcal{G} = (V, E)$. Each node $i \in V = \{1, \cdots d\}$ is associated with a random variable $\boldsymbol{x}_i$ and each edge $(i, j) \in E$ represents a direct causal relation from variable $\boldsymbol{x}_i$ to $\boldsymbol{x}_j$. The distribution $\mathbb{P}_{\boldsymbol{x}}$ is Markov to the graph $\mathcal{G}$, which means that the joint distribution can be factorized as

$$p(\boldsymbol{x}_1, \cdots, \boldsymbol{x}_d) = \prod_{j=1}^{d} p_j(\boldsymbol{x}_j | \boldsymbol{x}_{\pi_j^{\mathcal{G}}}) \,, \tag{4.1}$$

where $\pi_j^{\mathcal{G}}$ is the set of parents of the node j in the graph $\mathcal{G}$, and $\boldsymbol{x}_B$, for a subset $B \subseteq V$, denotes the entries of the vector $\boldsymbol{x}$ with indices in B. In this work, we assume *causal sufficiency*, i.e., there is no hidden common cause that is causing more than one variable in $\boldsymbol{x}$ Peters et al. [2017].

Interventions. In contrast with standard Bayesian Networks, CGMs support interventions. Formally, an intervention on a variable $\boldsymbol{x}_j$ corresponds to replacing its conditional $p_j(\boldsymbol{x}_j | \boldsymbol{x}_{\pi_j^{\mathcal{G}}})$ by a new conditional $\tilde{p}_j(\boldsymbol{x}_j | \boldsymbol{x}_{\pi_j^{\mathcal{G}}})$ in Equation (4.1), thus modifying the distribution only locally. Interventions can be performed on multiple variables simultaneously and we call the *interventional target* the set $I \subseteq V$ of such variables. When considering more than one intervention, we denote the interventional target of the kth intervention by I_k. Throughout this paper, we assume that the observational distribution (the original distribution without interventions) is observed, and denote it by $I_1 := \emptyset$. We define the *interventional family* by $\mathcal{I} := (I_1, \cdots, I_K)$, where K is the number of interventions (including the observational setting). Finally, the kth interventional joint density is

$$p^{(k)}(\boldsymbol{x}_1, \cdots, \boldsymbol{x}_d) := \prod_{j \notin I_k} p_j^{(1)}(\boldsymbol{x}_j | \boldsymbol{x}_{\pi_j^{\mathcal{G}}}) \prod_{j \in I_k} p_j^{(k)}(\boldsymbol{x}_j | \boldsymbol{x}_{\pi_j^{\mathcal{G}}}) \,, \tag{4.2}$$

where the assumption of causal sufficiency is implicit to this definition of interventions.

Type of interventions. The general type of interventions described in (4.2) are called imperfect (or soft, parametric) Peters et al. [2017], Eaton and Murphy [2007], Eberhardt [2007]. A specific case that is often considered is (stochastic) perfect interventions (or hard, structural) Eberhardt and Scheines [2007], Yang et al. [2018], Korb et al. [2004] where $p_j^{(k)}(\boldsymbol{x}_j|\boldsymbol{x}_{\pi_j^{\mathcal{G}}}) = p_j^{(k)}(\boldsymbol{x}_j)$ for all $j \in I_k$, thus removing the dependencies with their parents (see Figure 4.1). Real-world examples of these types of interventions include gene knockout/knockdown in biology. Analogous to a perfect intervention, a gene knockout completely suppresses the expression of one gene and removes dependencies to regulators of gene expression. In contrast, a gene knockdown hinders the expression of one gene without removing dependencies with regulators [Zimmer et al., 2019], and is thus an imperfect intervention.

4.2.2. Causal structure learning

In causal structure learning, the goal is to recover the causal DAG $\mathcal{G}$ using samples from $\mathbb{P}_x$ and, when available, from interventional distributions. This problem presents two main challenges: 1) the size of the search space is super-exponential in the number of nodes [Chickering, 2003] and 2) the true DAG is not always identifiable (more severe without interventional data). Methods for this task are often divided into three groups: constraint-based, score-based, and hybrid methods. We briefly review these below.

Constraint-based methods typically rely on conditional independence testing to identify edges in $\mathcal{G}$. The PC algorithm [Spirtes et al., 2000] is a classical example that works with observational data. It performs conditional independence tests with a conditioning set that increases at each step of the algorithm and finds an equivalence class that satisfies all independencies. Methods that support interventional data include COmbINE [Triantafillou and Tsamardinos, 2015], HEJ [Hyttinen et al., 2014], which both rely on Boolean satisfiability solvers to find a graph that satisfies all constraints; and Kocaoglu et al. [2019], which proposes an algorithm inspired by FCI Spirtes et al. [2000]. In contrast with our method, these methods account for latent confounders. The *Joint causal inference* framework (JCI) Mooij et al. [2020] supports latent confounders and can deal with interventions with unknown targets. This framework can be used with various observational constraint-based algorithms such as PC or FCI. Another type of constraint-based method exploits the invariance of causal mechanisms across interventional distributions, e.g., ICP [Peters et al., 2016, Heinze-Deml et al., 2018b]. As will later be presented in Section 4.3, our loss function also accounts for such invariances.

Score-based methods formulate the problem of estimating the ground truth DAG $\mathcal{G}^*$ by optimizing a score function $\mathcal{S}$ over the space of DAGs. The estimated DAG $\hat{\mathcal{G}}$ is given by

$$\hat{\mathcal{G}} \in \arg\max_{\mathcal{G} \in \text{DAG}} \mathcal{S}(\mathcal{G}) \,. \tag{4.3}$$

A typical choice of score in the purely observational setting is the regularized maximum likelihood:

$$\mathcal{S}(\mathcal{G}) := \max_{\boldsymbol{\theta}} \mathbb{E}_{\boldsymbol{x} \sim \mathbb{P}_{\boldsymbol{x}}} \log f_{\boldsymbol{\theta}}(\boldsymbol{x}) - \lambda |\mathcal{G}| \,, \tag{4.4}$$

where $f_{\boldsymbol{\theta}}$ is a density function parameterized by $\boldsymbol{\theta}$, $|\mathcal{G}|$ is the number of edges in $\mathcal{G}$ and λ is a positive scalar.[1] Since the space of DAGs is super-exponential in the number of nodes, these methods often rely on greedy combinatorial search algorithms. A typical example is GIES [Hauser and Bühlmann, 2012], an adaptation of GES [Chickering, 2003] to perfect interventions. In contrast with our method, GIES assumes a *linear* gaussian model and optimizes the Bayesian information criterion (BIC) over the space of $\mathcal{I}$-Markov equivalence classes (see Definition 4.3 in Appendix A.1). CAM [Bühlmann et al., 2014] is also a score-based method using greedy search, but it is nonlinear: it assumes an additive noise model where the nonlinear functions are additive. In the original paper, CAM only addresses the observational case where additive noise models are identifiable, however code is available to support perfect interventions.

Hybrid methods combine constraint and score-based approaches. Among these, IGSP [Wang et al., 2017, Yang et al., 2018] is a method that optimizes a score based on conditional independence tests. Contrary to GIES, this method has been shown to be consistent under the faithfulness assumption. Furthermore, this method has recently been extended to support interventions with unknown targets (UT-IGSP) [Squires et al., 2020], which are also supported by our method.

4.2.3. Continuous constrained optimization for structure learning

A new line of research initiated by Zheng et al. [2018], which serves as the basis for our work, reformulates the combinatorial problem of finding the optimal DAG as a continuous constrained-optimization problem, effectively avoiding the combinatorial search. Analogous to standard score-based approaches, these methods rely on a model $f_{\boldsymbol{\theta}}$ parametrized by $\boldsymbol{\theta}$, though $\boldsymbol{\theta}$ also encodes the graph $\mathcal{G}$. Central to this class of methods are both the use a *weighted adjacency matrix* $\boldsymbol{A}_{\boldsymbol{\theta}} \in \mathbb{R}_{\geq 0}^{d \times d}$ (which depends on the parameters of the model) and the acyclicity constraint introduced by Zheng et al. [2018] in the context of linear models:

$$\operatorname{Tr} e^{\boldsymbol{A}_{\boldsymbol{\theta}}} - d = 0 \,. \tag{4.5}$$

[1] This turns into the BIC score when the expectation is estimated with n samples, the model has one parameter per edge (like in linear models) and $\lambda = \frac{\log n}{2n}$ [Peters et al., 2017, Section 7.2.2].

The weighted adjacency matrix encodes the DAG estimator $\hat{\mathcal{G}}$ as $(\boldsymbol{A}_{\boldsymbol{\theta}})_{ij} > 0 \iff i \to j \in \hat{\mathcal{G}}$. Zheng et al. [2018] showed, in the context of linear models, that $\hat{\mathcal{G}}$ is acyclic if and only if the constraint $\operatorname{Tr} e^{\boldsymbol{A}_{\boldsymbol{\theta}}} - d = 0$ is satisfied. The general optimization problem is then

$$\max_{\boldsymbol{\theta}} \mathbb{E}_{\boldsymbol{x} \sim \mathbb{P}_{\boldsymbol{x}}} \log f_{\boldsymbol{\theta}}(\boldsymbol{x}) - \lambda \Omega(\boldsymbol{\theta}) \text{ s.t. } \operatorname{Tr} e^{\boldsymbol{A}_{\boldsymbol{\theta}}} - d = 0\,, \tag{4.6}$$

where $\Omega(\boldsymbol{\theta})$ is a regularizing term penalizing the number of edges in $\hat{\mathcal{G}}$. This problem is then approximately solved using an augmented Lagrangian procedure, as proposed by Zheng et al. [2018]. Note that the problem in Equation (4.6) is very similar to the one resulting from Equations (4.3) and (4.4).

Continuous-constrained methods differ in their choice of model, weighted adjacency matrix, and the specifics of their optimization procedures. For instance, NOTEARS [Zheng et al., 2018] assumes a Gaussian linear model with equal variances where $\boldsymbol{\theta} := \boldsymbol{W} \in \mathbb{R}^{d \times d}$ is the matrix of regression coefficients, $\Omega(\boldsymbol{\theta}) := ||\boldsymbol{W}||_1$ and $\boldsymbol{A}_{\boldsymbol{\theta}} := \boldsymbol{W} \odot \boldsymbol{W}$ is the weighted adjacency matrix. Several other methods use neural networks to model nonlinear relations via $f_{\boldsymbol{\theta}}$ and have been shown to be competitive with classical methods [Lachapelle et al., 2020, Zheng et al., 2020]. In some methods, the parameter $\boldsymbol{\theta}$ can be partitioned into $\boldsymbol{\theta}_1$ and $\boldsymbol{\theta}_2$ such that $f_{\boldsymbol{\theta}} = f_{\boldsymbol{\theta}_1}$ and $A_{\boldsymbol{\theta}} = \boldsymbol{A}_{\boldsymbol{\theta}_2}$ [Kalainathan et al., 2018, Ng et al., 2019, Ke et al., 2019] while in others, such a decoupling is not possible, i.e., the adjacency matrix $\boldsymbol{A}_{\boldsymbol{\theta}}$ is a function of the neural networks parameters [Lachapelle et al., 2020, Zheng et al., 2020]. In terms of scoring, most methods rely on maximum likelihood or variants like implicit maximum likelihood [Kalainathan et al., 2018] and evidence lower bound [Yu et al., 2019a]. Zhu and Chen [2020] also rely on the acyclicity constraint, but use reinforcement learning as a search strategy to estimate the DAG. Ke et al. [2019] learn a DAG from interventional data by optimizing an unconstrained objective with a regularization term inspired by the acyclicity constraint, but that penalizes only cycles of length two. However, their work is limited to discrete distributions and single-node interventions. To the best of our knowledge, no work has investigated, in a general manner, the use of continuous-constrained approaches in the context of interventions as we present in the next section.

4.3. DCDI: Differentiable causal discovery from interventional data

In this section, we present a score for imperfect interventions, provide a theorem showing its validity, and show how it can be maximized using the continuous-constrained approach to structure learning. We also provide a theoretically grounded extension to interventions with unknown targets.

4.3.1. A score for imperfect interventions

The model we consider uses neural networks to model conditional densities. Moreover, we encode the DAG $\mathcal{G}$ with a binary adjacency matrix $\boldsymbol{M}^{\mathcal{G}} \in \{0,1\}^{d\times d}$ which acts as a mask on the neural networks inputs. We similarly encode the interventional family $\mathcal{I}$ with a binary matrix $\boldsymbol{R}^{\mathcal{I}} \in \{0,1\}^{K\times d}$, where $\boldsymbol{R}^{\mathcal{I}}_{kj} = 1$ means that $\boldsymbol{x}_j$ is a target in I_k. In line with the definition of interventions in Equation (4.2), we model the joint density of the kth intervention by

$$f^{(k)}(\boldsymbol{x}; \boldsymbol{M}^{\mathcal{G}}, \boldsymbol{R}^{\mathcal{I}}, \boldsymbol{\phi}) := \prod_{j=1}^{d} \tilde{f}(\boldsymbol{x}_j; \mathrm{NN}(\boldsymbol{M}^{\mathcal{G}}_j \odot \boldsymbol{x}; \boldsymbol{\phi}^{(1)}_j))^{1-\boldsymbol{R}^{\mathcal{I}}_{kj}} \tilde{f}(\boldsymbol{x}_j; \mathrm{NN}(\boldsymbol{M}^{\mathcal{G}}_j \odot \boldsymbol{x}; \boldsymbol{\phi}^{(k)}_j))^{\boldsymbol{R}^{\mathcal{I}}_{kj}}, \tag{4.7}$$

where $\boldsymbol{\phi} := \{\boldsymbol{\phi}^{(1)}, \cdots, \boldsymbol{\phi}^{(K)}\}$, the NN's are neural networks parameterized by $\boldsymbol{\phi}^{(1)}_j$ or $\boldsymbol{\phi}^{(k)}_j$, the operator $\odot$ denotes the Hadamard product (element-wise) and $\boldsymbol{M}^{\mathcal{G}}_j$ denotes the jth column of $\boldsymbol{M}^{\mathcal{G}}$, which enables selecting the parents of node j in the graph $\mathcal{G}$. The neural networks output the parameters of a density function $\tilde{f}$, which in principle, could be any density. We experiment with Gaussian distributions and more expressive normalizing flows (see Section 4.3.4).

We denote $\mathcal{G}^*$ and $\mathcal{I}^* := (I^*_1, ..., I^*_K)$ to be the ground truth causal DAG and ground truth interventional family, respectively. In this section, we assume that $\mathcal{I}^*$ is known, but we will relax this assumption in Section 4.3.3. We propose maximizing with respect to $\mathcal{G}$ the following regularized maximum log-likelihood score:

$$\mathcal{S}_{\mathcal{I}^*}(\mathcal{G}) := \sup_{\boldsymbol{\phi}} \sum_{k=1}^{K} \mathbb{E}_{\boldsymbol{x}\sim p^{(k)}} \log f^{(k)}(\boldsymbol{x}; \boldsymbol{M}^{\mathcal{G}}, \boldsymbol{R}^{\mathcal{I}^*}, \boldsymbol{\phi}) - \lambda|\mathcal{G}|\,, \tag{4.8}$$

where $p^{(k)}$ stands for the kth ground truth interventional distribution from which the data is sampled. A careful inspection of (4.7) reveals that the conditionals of the model are invariant across interventions *in which they are not targeted*. Intuitively, this means that maximizing (4.8) will favor graphs $\mathcal{G}$ in which a conditional $p(\boldsymbol{x}_j|\boldsymbol{x}_{\pi^{\mathcal{G}}_j})$ is invariant across all interventional distributions in which $\boldsymbol{x}_j$ is not a target, i.e., $j \notin I^*_k$. This is a fundamental property of causal graphical models.

We now present our first theoretical result (see Appendix A.2 for the proof). This theorem states that, under appropriate assumptions, maximizing $\mathcal{S}_{\mathcal{I}^*}(\mathcal{G})$ yields an estimated DAG $\hat{\mathcal{G}}$ that is $\mathcal{I}^*$-Markov equivalent to the true DAG $\mathcal{G}^*$. We use the notion of $\mathcal{I}^*$-Markov equivalence introduced by Yang et al. [2018] and recall its meaning in Definition 4.3 of Appendix A.1. Briefly, the $\mathcal{I}^*$-Markov equivalence class of $\mathcal{G}^*$ is a set of DAGs which are indistinguishable from $\mathcal{G}^*$ given the interventional targets in $\mathcal{I}^*$. This means identifying the $\mathcal{I}^*$-Markov equivalence class of $\mathcal{G}^*$ is the *best* one can hope for given the interventions $\mathcal{I}^*$ *without making further distributional assumptions*.

Theorem 4.1 (Identification via score maximization)**.** *Suppose the interventional family $\mathcal{I}^*$ is such that $I^*_1 := \emptyset$. Let $\mathcal{G}^*$ be the ground truth DAG and $\hat{\mathcal{G}} \in \arg\max_{\mathcal{G}\in DAG} \mathcal{S}_{\mathcal{I}^*}(\mathcal{G})$. Assume that the*

density model has enough capacity to represent the ground truth distributions, that $\mathcal{I}^*$-faithfulness holds, that the density model is strictly positive and that the ground truth densities $p^{(k)}$ have finite differential entropy, respectively Assumptions 4.1, 4.2, 4.3 & 4.4 (see Appendix A.2 for precise statements). Then for $\lambda > 0$ small enough, we have that $\hat{\mathcal{G}}$ is $\mathcal{I}^*$-Markov equivalent to $\mathcal{G}^*$.

Proof idea. Using the graphical characterization of $\mathcal{I}$-Markov equivalence from Yang et al. [2018], we verify that every graph outside the equivalence class has a lower score than that of the ground truth graph. We show this by noticing that any such graph will either have more edges than $\mathcal{G}^*$ or limit the distributions expressible by the model in such a way as to prevent it from properly fitting the ground truth. Moreover, the coefficient λ must be chosen small enough to avoid too sparse solutions. □

$\mathcal{I}^*$-faithfulness (Assumption 4.2) enforces two conditions. The first one is the usual faithfulness condition, i.e., whenever a conditional independence statement holds in the observational distribution, the corresponding d-separation holds in $\mathcal{G}^*$. The second one requires that the interventions are non-pathological in the sense that every variable that can be potentially affected by the intervention are indeed affected. See Appendix A.2 for more details and examples of $\mathcal{I}^*$-faithfulness violations.

To interpret this result, note that the $\mathcal{I}^*$-Markov equivalence class of $\mathcal{G}^*$ tends to get smaller as we add interventional targets to the interventional family $\mathcal{I}^*$. As an example, when $\mathcal{I}^* = (\emptyset, \{1\}, \cdots, \{d\})$, i.e., when each node is individually targeted by an intervention, $\mathcal{G}^*$ is alone in its equivalence class and, if assumptions of Theorem 4.1 hold, $\hat{\mathcal{G}} = \mathcal{G}^*$. See Corollary 4.1 in Appendix A.1 for details.

Perfect interventions. The score $\mathcal{S}_{\mathcal{I}^*}(\mathcal{G})$ can be specialized for perfect interventions, i.e., where the targeted nodes are completely disconnected from their parents. The idea is to leverage the fact that the conditionals targeted by the intervention in Equation (4.7) should not depend on the graph $\mathcal{G}$ anymore. This means that these terms can be removed without affecting the maximization w.r.t. $\mathcal{G}$. We use this version of the score when experimenting with perfect interventions and present it in Appendix A.4.

4.3.2. A continuous-constrained formulation

To allow for gradient-based stochastic optimization, we follow Kalainathan et al. [2018], Ng et al. [2019] and treat the adjacency matrix $\boldsymbol{M}^{\mathcal{G}}$ as *random*, where the entries $\boldsymbol{M}_{ij}^{\mathcal{G}}$ are independent Bernoulli variables with success probability $\sigma(\alpha_{ij})$ (σ is the sigmoid function) and α_{ij} is a scalar parameter. We group these α_{ij}'s into a matrix $\boldsymbol{\Lambda} \in \mathbb{R}^{d \times d}$. We then replace the score $\mathcal{S}_{\mathcal{I}^*}(\mathcal{G})$ (4.8) with the following relaxation:

$$\hat{\mathcal{S}}_{\mathcal{I}^*}(\boldsymbol{\Lambda}) := \sup_{\boldsymbol{\phi}} \mathop{\mathbb{E}}_{\boldsymbol{M} \sim \sigma(\boldsymbol{\Lambda})} \left[\sum_{k=1}^{K} \mathop{\mathbb{E}}_{\boldsymbol{x} \sim p^{(k)}} \log f^{(k)}(\boldsymbol{x}; \boldsymbol{M}, \boldsymbol{R}^{\mathcal{I}^*}, \boldsymbol{\phi}) - \lambda ||\boldsymbol{M}||_0 \right], \tag{4.9}$$

where we dropped the $\mathcal{G}$ superscript in $\boldsymbol{M}$ to lighten notation. This score tends asymptotically to $\mathcal{S}_{\mathcal{I}^*}(\mathcal{G})$ as $\sigma(\boldsymbol{\Lambda})$ progressively concentrates its mass on $\mathcal{G}$.[2] While the expectation of the log-likelihood term is intractable, the expectation of the regularizing term simply evaluates to $\lambda||\sigma(\boldsymbol{\Lambda})||_1$. This score can then be maximized under the acyclicity constraint presented in Section 4.2.3:

$$\sup_{\boldsymbol{\Lambda}} \hat{\mathcal{S}}_{\mathcal{I}^*}(\boldsymbol{\Lambda}) \quad \text{s.t.} \quad \operatorname{Tr} e^{\sigma(\boldsymbol{\Lambda})} - d = 0\,. \tag{4.10}$$

This problem presents two main challenges: it is a constrained problem and it contains intractable expectations. As proposed by Zheng et al. [2018], we rely on the *augmented Lagrangian* procedure to optimize $\boldsymbol{\phi}$ and $\boldsymbol{\Lambda}$ jointly under the acyclicity constraint. This procedure transforms the constrained problem into a sequence of unconstrained subproblems which can themselves be optimized via a standard stochastic gradient descent algorithm for neural networks such as RMSprop. The procedure should converge to a stationary point of the original constrained problem (which is not necessarily the global optimum due to the non-convexity of the problem). In Appendix B.3, we give details on the augmented Lagrangian procedure and show the learning process in details with a concrete example.

The gradient of the likelihood part of $\hat{\mathcal{S}}_{\mathcal{I}^*}(\boldsymbol{\Lambda})$ w.r.t. $\boldsymbol{\Lambda}$ is estimated using the Straight-Through Gumbel estimator. This amounts to using Bernoulli samples in the forward pass and Gumbel-Softmax samples in the backward pass which can be differentiated w.r.t. $\boldsymbol{\Lambda}$ via the reparametrization trick Jang et al. [2017], Maddison et al. [2017]. This approach was already shown to give good results in the context of continuous optimization for causal discovery in the purely observational case Ng et al. [2019], Kalainathan et al. [2018]. We emphasize that our approach belongs to the general framework presented in Section 4.2.3 where the global parameter $\boldsymbol{\theta}$ is $\{\boldsymbol{\phi}, \boldsymbol{\Lambda}\}$, the weighted adjacency matrix $\boldsymbol{A}_{\boldsymbol{\theta}}$ is $\sigma(\boldsymbol{\Lambda})$ and the regularizing term $\Omega(\boldsymbol{\theta})$ is $||\sigma(\boldsymbol{\Lambda})||_1$.

4.3.3. Interventions with unknown targets

Until now, we have assumed that the ground truth interventional family $\mathcal{I}^*$ is known. We now consider the case were it is unknown and, thus, needs to be learned. To do so, we propose a simple modification of score (4.8) which consists in adding regularization to favor sparse interventional families.

$$\mathcal{S}(\mathcal{G}, \mathcal{I}) := \sup_{\boldsymbol{\phi}} \sum_{k=1}^{K} \mathbb{E}_{\boldsymbol{x}\sim p^{(k)}} \log f^{(k)}(\boldsymbol{x}; \boldsymbol{M}^{\mathcal{G}}, \boldsymbol{R}^{\mathcal{I}}, \boldsymbol{\phi}) - \lambda|\mathcal{G}| - \lambda_{\boldsymbol{R}}|\mathcal{I}|\,, \tag{4.11}$$

where $|\mathcal{I}| = \sum_{k=1}^{K} |I_k|$. The following theorem, proved in Appendix A.3, extends Theorem 4.1 by showing that, under the same assumptions, maximizing $\mathcal{S}(\mathcal{G}, \mathcal{I})$ with respect to both $\mathcal{G}$ and $\mathcal{I}$ recovers both the $\mathcal{I}^*$-Markov equivalence class of $\mathcal{G}^*$ and the ground truth interventional family $\mathcal{I}^*$.

[2] In practice, we observe that $\sigma(\boldsymbol{\Lambda})$ tends to become deterministic as we optimize.

Theorem 4.2 (Unknown targets identification)**.** *Suppose $\mathcal{I}^*$ is such that $I_1^* := \emptyset$. Let $\mathcal{G}^*$ be the ground truth DAG and $(\hat{\mathcal{G}}, \hat{\mathcal{I}}) \in \arg\max_{\mathcal{G} \in DAG, \mathcal{I}} \mathcal{S}(\mathcal{G}, \mathcal{I})$. Under the same assumptions as Theorem 4.1 and for $\lambda, \lambda_{\boldsymbol{R}} > 0$ small enough, $\hat{\mathcal{G}}$ is $\mathcal{I}^*$-Markov equivalent to $\mathcal{G}^*$ and $\hat{\mathcal{I}} = \mathcal{I}^*$.*

Proof idea. We simply append a few steps at the beginning of the proof of Theorem 4.1 which show that whenever $\mathcal{I} \neq \mathcal{I}^*$, the resulting score is worse than $\mathcal{S}(\mathcal{G}^*, \mathcal{I}^*)$, and hence is not optimal. This is done using arguments very similar to Theorem 4.1 and choosing λ and $\lambda_{\boldsymbol{R}}$ small enough. □

Theorem 4.2 informs us that ignoring which nodes are targeted during interventions does not affect identifiability. However, this result assumes implicitly that the learner knows which data set is the observational one.

Similarly to the development of Section 4.3.2, the score $\mathcal{S}(\mathcal{G}, \mathcal{I})$ can be relaxed by treating entries of $\boldsymbol{M}^{\mathcal{G}}$ and $\boldsymbol{R}^{\mathcal{I}}$ as independent Bernoulli random variables parameterized by $\sigma(\alpha_{ij})$ and $\sigma(\beta_{kj})$, respectively. We thus introduced a new learnable parameter β. The resulting relaxed score is similar to (4.9), but the expectation is taken w.r.t. to $\boldsymbol{M}$ and $\boldsymbol{R}$. Similarly to $\boldsymbol{\Lambda}$, the Straight-Through Gumbel estimator is used to estimate the gradient of the score w.r.t. the parameters β_{kj}. For perfect interventions, we adapt this score by masking all inputs of the neural networks under interventions.

The related work of Ke et al. [2019], which also support unknown targets, bears similarity to DCDI but addresses a different setting in which interventions are obtained sequentially in an online fashion. One important difference is that their method attempts to identify the *single node* that has been intervened upon (as a hard prediction), whereas DCDI learns a distribution over all potential interventional families via the continuous parameters $\sigma(\beta_{kj})$, which typically becomes deterministic at convergence. Ke et al. [2019] also use random masks to encode the graph structure but estimates the gradient w.r.t. their distribution parameters using the log-trick which is known to have high variance Rezende et al. [2014] compared to reparameterized gradient Maddison et al. [2017].

4.3.4. DCDI with normalizing flows

In this section, we describe how the scores presented in Sections 4.3.2 & 4.3.3 can accommodate powerful density approximators. In the purely observational setting, very expressive models usually hinder identifiability, but this problem vanishes when enough interventions are available. There are many possibilities when it comes to the choice of the density function $\tilde{f}$. In this paper, we experimented with simple Gaussian distributions as well as *normalizing flows* [Rezende and Mohamed, 2015] which can represent complex causal relationships, e.g., multi-modal distributions that can occur in the presence of latent variables that are parent of only one variable.

A normalizing flow $\boldsymbol{\tau}(\cdot; \boldsymbol{\omega})$ is an invertible function (e.g., a neural network) parameterized by $\boldsymbol{\omega}$ with a tractable Jacobian, which can be used to model complex densities by transforming a simple

random variable via the change of variable formula:

$$\tilde{f}(\boldsymbol{z};\boldsymbol{\omega}) := \left|\det\left(\frac{\partial \boldsymbol{\tau}(\boldsymbol{z};\boldsymbol{\omega})}{\partial \boldsymbol{z}}\right)\right| p(\boldsymbol{\tau}(\boldsymbol{z};\boldsymbol{\omega}))\,, \tag{4.12}$$

where $\frac{\partial \boldsymbol{\tau}(\boldsymbol{z};\boldsymbol{\omega})}{\partial \boldsymbol{z}}$ is the Jacobian matrix of $\boldsymbol{\tau}(\cdot;\boldsymbol{\omega})$ and $p(\cdot)$ is a simple density function, e.g., a Gaussian. The function $\tilde{f}(\cdot;\boldsymbol{\omega})$ can be plugged directly into the scores presented earlier by letting the neural networks $\text{NN}(\cdot;\boldsymbol{\phi}_j^{(k)})$ output the parameter $\boldsymbol{\omega}_j$ of the normalizing flow $\boldsymbol{\tau}_j$ for each variable $\boldsymbol{x}_j$. In our implementation, we use *deep sigmoidal flows* (DSF), a specific instantiation of normalizing flows which is a universal density approximator Huang et al. [2018b]. Details about DSF are relayed to Appendix B.2.

4.4. Experiments

We tested DCDI with Gaussian densities (DCDI-G) and with normalizing flows (DCDI-DSF) on a real-world data set and several synthetic data sets. The real-world task is a flow cytometry data set from Sachs et al. [2005]. Our results, reported in Appendix C.1, show that our approach performs comparably to state-of-the-art methods. In this section, we focus on synthetic data sets, since these allow for a more systematic comparison of methods against various factors of variation (type of interventions, graph size, density, type of mechanisms).

We consider synthetic data sets with three interventional settings: perfect/known, imperfect/-known, and perfect/unknown. Each data set has one of the three different types of causal mechanisms: i) linear Squires et al. [2020], ii) nonlinear additive noise model (ANM) Bühlmann et al. [2014], and iii) nonlinear with non-additive noise using neural networks (NN) Kalainathan et al. [2018]. For each data set type, graphs vary in size (d = 10 or 20) and density ($e = 1$ or 4 where $e \cdot d$ is the average number of edges). For conciseness, we present results for 20-node graphs in the main text and report results on 10-node graphs in Appendix C.7; conclusions are similar for all sizes. For each condition, ten graphs are sampled with their causal mechanisms and then observational and interventional data are generated. Each data set has 10 000 samples uniformly distributed in the different interventional settings. A total of d interventions were performed, each by sampling up to $0.1d$ target nodes. For more details on the generation process, see Appendix B.1.

Most methods have an hyperparameter controlling DAG sparsity. Although performance is sensitive to this hyperparameter, many papers do not specify how it was selected. For score-based methods (GIES, CAM and DCDI), we select it by maximizing the held-out likelihood as explained in Appendix B.5 (without using the ground truth DAG). In contrast, since constraint-based methods (IGSP, UT-IGSP, JCI-PC) do not yield a likelihood model to evaluate on held-out data, we use a fixed cutoff parameter ($\alpha = 1\text{e}{-3}$) that leads to good results. We report additional results with different cutoff values in Appendix C.7. For IGSP and UT-IGSP, we always use the

independence test well tailored to the data set type: partial correlation test for Gaussian linear data and KCI-test [Zhang et al., 2011] for nonlinear data.

The performance of each method is assessed by two metrics comparing the estimated graph to the ground truth graph: i) the *structural Hamming distance* (SHD) which is simply the number of edges that differ between two DAGs (either reversed, missing or superfluous) and ii) the *structural interventional distance* (SID) which assesses how two DAGs differ with respect to their causal inference statements [Peters and Bühlmann, 2015]. In Appendix C.6, we also report how well the graph can be used to predict the effect of unseen interventions Gentzel et al. [2019]. Our implementation is available here and additional information about the baseline methods is provided in Appendix B.4.

4.4.1. Results for different intervention types

Perfect interventions. We compare our methods to GIES [Hauser and Bühlmann, 2012], a modified version of CAM [Bühlmann et al., 2014] that support interventions and IGSP [Wang et al., 2017]. The conditionals of targeted nodes were replaced by the marginal $\mathcal{N}(2, 1)$ similarly to Hauser and Bühlmann [2012], Squires et al. [2020]. Boxplots for SHD and SID over 10 graphs are shown in Figure 4.2. For all conditions, DCDI-G and DCDI-DSF shows competitive results in term of SHD and SID. For graphs with a higher number of average edges, DCDI-G and DCDI-DSF outperform all methods. GIES often shows the best performance for the linear data set, which is not surprising given that it makes the right assumptions, i.e., linear functions with Gaussian noise.

Imperfect interventions. Our conclusions are similar to the perfect intervention setting. As shown in Figure 4.3, DCDI-G and DCDI-DSF show competitive results and outperform other methods for graphs with a higher connectivity. The nature of the imperfect interventions are explained in Appendix B.1.

Perfect unknown interventions. We compare to UT-IGSP [Squires et al., 2020], an extension of IGSP that deal with unknown interventions. The data used are the same as in the perfect intervention setting, but the intervention targets are hidden. Results are shown in Figure 4.4. Except for linear data sets with sparse graphs, DCDI-G and DCDI-DSF show an overall better performance than UT-IGSP.

Summary. For all intervention settings, DCDI has overall the best performance. In Appendix C.5, we show similar results for different types of perfect/imperfect interventions. While the advantage of DCDI-DSF over DCDI-G is marginal, it might be explained by the fact that the densities can be sufficiently well modeled by DCDI-G. In Appendix C.2, we show cases where DCDI-G fails to detect the right causal direction due to its lack of capacity, whereas DCDI-DSF systematically succeeds. In Appendix C.4, we present an ablation study confirming the advantage of neural networks against linear models and the ability of our score to leverage interventional data.

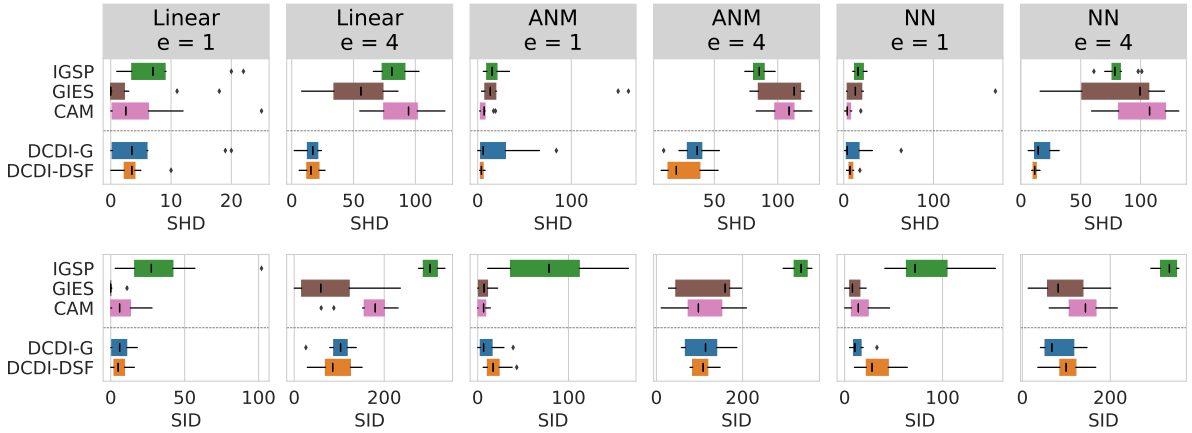


Figure 4.2. Perfect interventions. SHD and SID (lower is better) for 20-node graphs

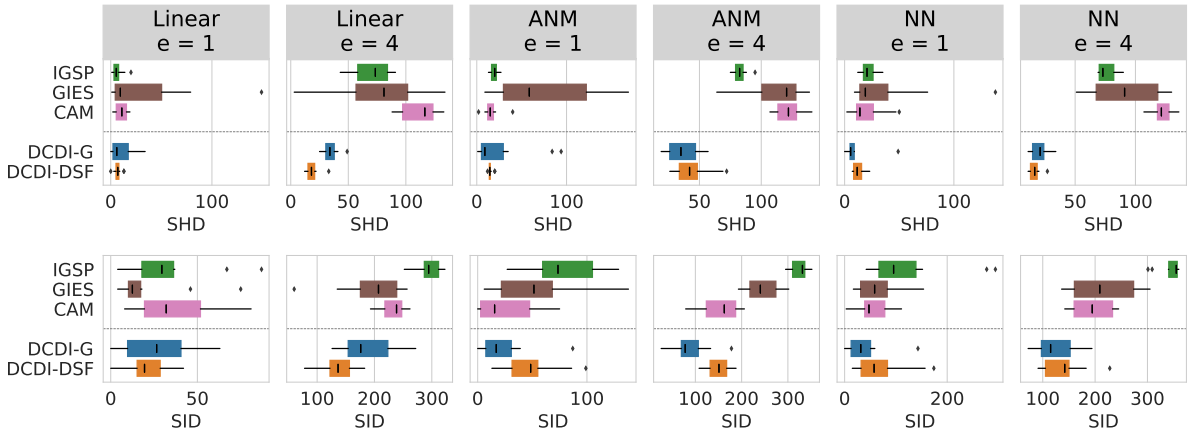


Figure 4.3. Imperfect interventions. SHD and SID for 20-node graphs

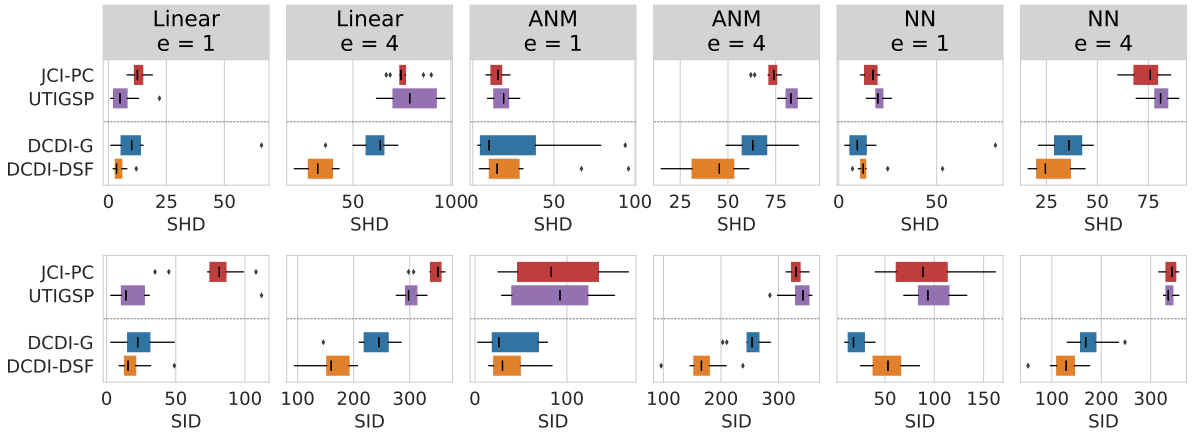


Figure 4.4. Unknown interventions. SHD and SID for 20-node graphs

4.4.2. Scalability experiments

So far the experiments focused on moderate size data sets, both in terms of number of variables (10 or 20) and number of examples ($\approx 10^4$). In Appendix C.3, we compare the running times of

DCDI to those of other methods on graphs of up to 100 nodes and on data sets of up to 1 million examples.

The augmented Lagrangian procedure on which DCDI relies requires the computation of the matrix exponential at each gradient step, which costs $O(d^3)$. We found this does not prevent DCDI from being applied to 100 nodes graphs. Several constraint-based methods use kernel-based conditional independence tests [Zhang et al., 2011, Fukumizu et al., 2008], which scale poorly with the number of examples. For example, KCI-test scales in $O(n^3)$ [Strobl et al., 2019] and HSIC in $O(n^2)$ [Zhang et al., 2018]. On the other hand, DCDI is not greatly affected by the sample size since it relies on stochastic gradient descent which is known to scale well with the data set size Bottou [2010]. Our comparison shows that, among all considered methods, DCDI is the only one supporting nonlinear relationships that can scale to as much as one million examples. We believe that this can open the way to new applications of causal discovery where data is abundant.

4.5. Conclusion

We proposed a general continuous-constrained method for causal discovery which can leverage various types of interventional data as well as expressive neural architectures, such as normalizing flows. This approach is rooted in a sound theoretical framework and is competitive with other state-of-the-art algorithms on real and simulated data sets, both in terms of graph recovery and scalability. This work opens interesting opportunities for future research. One direction is to extend DCDI to time-series data, where non-stationarities can be modeled as unknown interventions Pfister et al. [2019]. Another exciting direction is to learn representations of variables across multiple systems that could serve as prior knowledge for causal discovery in low data settings.

Broader impact

Causal structure learning algorithms are general tools that address two high-level tasks: *understanding* and *acting*. That is, they can help a user understand a complex system and, once such an understanding is achieved, they can help in recommending actions. We envision positive impacts of our work in fields such as scientific investigation (e.g., interpreting and anticipating the outcome of experiments), policy making for decision-makers (e.g., identifying actions that could stimulate economic growth), and improving policies in autonomous agents (e.g., learning causal relationships in the world via interaction). As a concrete example, consider the case of gene knockouts/knockdowns experiments in the field of genomics, which aim to understand how specific genes and diseases interact Zimmer et al. [2019]. Learning causal models using interventions performed in this setting could help gain precious insight into gene pathways, which may catalyze the development of better pharmaceutic targets and broaden our understanding of complex diseases

such as cancer. Of course, applications are likely to extend beyond these examples which seem natural from our current position.

Like any methodological contribution, our work is not immune to undesirable applications that could have negative impacts. For instance, it would be possible, yet unethical for a policy-maker to use our algorithm to understand how specific human-rights violations can reduce crime and recommend their enforcement. The burden of using our work within ethical and benevolent boundaries would rely on the user. Furthermore, even when used in a positive application, our method could have unintended consequences if used without understanding its assumptions.

In order to use our method correctly, it is crucial to understand the assumptions that it makes about the data. When such assumptions are not met, the results may still be valid, but should be used as a support to decision rather than be considered as the absolute truth. These assumptions are:

- Causal sufficiency: there are no hidden confounding variables
- The samples for a given interventional distribution are independent and identically distributed
- The causal relationships form an acyclic graph (no feedback loops)
- Our theoretical results are valid in the infinite-data regime

We encourage users to be mindful of this and to carefully analyze their results before making decisions that could have a significant downstream impact.

Appendices of Chapter 4

A. Theory

A.1. Theoretical Foundations for Causal Discovery with Imperfect Interventions

Before showing results about our regularized maximum likelihood score from Section 4.3.1, we start by briefly presenting useful definitions and results from Yang et al. [2018]. We refer the reader to the original paper for a more comprehensive introduction to these notions, examples, and proofs. Throughout the appendix, we assume that the reader is comfortable with the concept of d-separation and immorality in directed graphs. These notions are presented in any standard book on probabilistic graphical models, e.g. Koller and Friedman [2009]. Recall that $\mathcal{I} := (I_1^*, ..., I_K)$ and that we always assume $I_1 := \emptyset$. Following the approach of Yang et al. [2018] and to simplify the presentation, we consider only densities which are strictly positive everywhere throught this appendix. We also note that while we present proofs for the cases where the distributions have densities with respect to the Lebesgue measure, all our results also hold for discrete distributions by simply replacing the Lebesgue measure with the counting measure in the integrals. We use the notation $i \to j \in \mathcal{G}$ to indicate that the edge (i, j) is in the edge set of $\mathcal{G}$. Given disjoint $A, B, C \subseteq V$, when C d-separates A from B in graph $\mathcal{G}$, we write $A \perp\!\!\!\perp_{\mathcal{G}} B \mid C$ and when random variables $\boldsymbol{x}_A$ and $\boldsymbol{x}_B$ are independent given $\boldsymbol{x}_C$ in distribution f, we write $\boldsymbol{x}_A \perp\!\!\!\perp_f \boldsymbol{x}_B \mid \boldsymbol{x}_C$.

Definition 4.1. *For a DAG $\mathcal{G}$, let $\mathcal{M}(\mathcal{G})$ be the set of strictly positive densities $f : \mathbb{R}^d \to \mathbb{R}$ such that*

$$f(\boldsymbol{x}_1, \cdots, \boldsymbol{x}_d) = \prod_j f_j(\boldsymbol{x}_j \mid \boldsymbol{x}_{\pi_j^{\mathcal{G}}}), \tag{4.13}$$

where $\int_{\mathbb{R}} f_j(\boldsymbol{x}_j \mid \boldsymbol{x}_{\pi_j^{\mathcal{G}}}) dm(\boldsymbol{x}_j) = 1$ for all $\boldsymbol{x}_{\pi_j^{\mathcal{G}}} \in \mathbb{R}^{|\pi_j^{\mathcal{G}}|}$ and all $j \in [d]$, where m is the Lebesgue measure on $\mathbb{R}$.

Next proposition is adapted from Lauritzen [1996, Theorem 3.27]. It relates the factorization of (4.13) to d-separation statements.

Proposition 4.1. *For a DAG $\mathcal{G}$ and a strictly positive density f,*[3] *we have $f \in \mathcal{M}(\mathcal{G})$ if and only if for any disjoint sets $A, B, C \subseteq V$ we have*

$$A \perp\!\!\!\perp_{\mathcal{G}} B \mid C \implies \boldsymbol{x}_A \perp\!\!\!\perp_f \boldsymbol{x}_B \mid \boldsymbol{x}_C \,.$$

Definition 4.2. *For a DAG $\mathcal{G}$ and an interventional family $\mathcal{I}$, let*

$$\mathcal{M}_{\mathcal{I}}(\mathcal{G}) := \{(f^{(k)})_{k\in[K]} \mid \forall k \in [K],\ f^{(k)} \in \mathcal{M}(\mathcal{G}) \text{ and } \forall j \notin I_k, f_j^{(k)}(\boldsymbol{x}_j \mid \boldsymbol{x}_{\pi_j^{\mathcal{G}}}) = f_j^{(1)}(\boldsymbol{x}_j \mid \boldsymbol{x}_{\pi_j^{\mathcal{G}}})\}\,.$$

Definition 4.2 defines a set $\mathcal{M}_{\mathcal{I}}(\mathcal{G})$ which contains all the sets of distributions $(f^{(k)})_{k\in[K]}$ which are coherent with the definition of interventions provided at Equation (4.2).[4] Note that the assumption of causal sufficiency is implicit to this definition of interventions. Analogously to the observational case, two different DAGs $\mathcal{G}_1$ and $\mathcal{G}_2$ can induce the same interventional distributions.

Definition 4.3 ($\mathcal{I}$-Markov Equivalence Class)**.** *Two DAGs $\mathcal{G}_1$ and $\mathcal{G}_2$ are $\mathcal{I}$-Markov equivalent iff $\mathcal{M}_{\mathcal{I}}(\mathcal{G}_1) = \mathcal{M}_{\mathcal{I}}(\mathcal{G}_2)$. We denote by $\mathcal{I}$-MEC$(\mathcal{G}_1)$ the set of all DAGs which are $\mathcal{I}$-Markov equivalent to $\mathcal{G}_1$, this is the $\mathcal{I}$-Markov equivalence class of $\mathcal{G}_1$.*

We now define an augmented graph containing exactly one node for each intervention k.

Definition 4.4. *Given a DAG $\mathcal{G}$ and an interventional family $\mathcal{I}$, the associated $\mathcal{I}$-DAG, denoted by $\mathcal{G}^{\mathcal{I}}$, is the graph $\mathcal{G}$ augmented with nodes ζ_k and edges $\zeta_k \to i$ for all $k \in [K] \setminus \{1\}$ and all $i \in I_k$.*

In the observational case, we say that a distribution f has the Markov property w.r.t. a graph $\mathcal{G}$ if whenever some d-separation holds in the graph, the corresponding conditional independence holds in f. We now define the $\mathcal{I}$-Markov property, which generalizes this idea to interventions. This property is important since it holds in causal graphical models, as Proposition 4.2 states.

Definition 4.5 ($\mathcal{I}$-Markov property)**.** *Let $\mathcal{I}$ be interventional family such that $I_1 := \emptyset$ and $(f^{(k)})_{k\in[K]}$ be a family of strictly positive densities over $\boldsymbol{x}$. We say that $(f^{(k)})_{k\in[K]}$ satisfies the $\mathcal{I}$-Markov property w.r.t. the $\mathcal{I}$-DAG $\mathcal{G}^{\mathcal{I}}$ iff*

(1) For any disjoint $A, B, C \subseteq V$, $A \perp\!\!\!\perp_{\mathcal{G}} B|C$ implies $\boldsymbol{x}_A \perp\!\!\!\perp_{f^{(k)}} \boldsymbol{x}_B|\boldsymbol{x}_C$ for all $k \in [K]$.

(2) For any disjoint $A, C \subseteq V$ and $k \in [K] \setminus \{1\}$,

$$A \perp\!\!\!\perp_{\mathcal{G}^{\mathcal{I}}} \zeta_k \mid C \cup \zeta_{-k} \text{ implies } f^{(k)}(\boldsymbol{x}_A|\boldsymbol{x}_C) = f^{(1)}(\boldsymbol{x}_A|\boldsymbol{x}_C), \text{ where } \zeta_{-k} := \zeta_{[K]\setminus\{1,k\}}.$$

The next proposition relates the definition of interventions with the $\mathcal{I}$-Markov property that we just defined.

Proposition 4.2. *(Yang et al. [2018]) Suppose the interventional family $\mathcal{I}$ is such that $I_1 := \emptyset$. Then $(f^{(k)})_{k\in[K]} \in \mathcal{M}_{\mathcal{I}}(\mathcal{G})$ iff $(f^{(k)})_{k\in[K]}$ is $\mathcal{I}$-Markov to $\mathcal{G}^{\mathcal{I}}$.*

The next theorem gives a graphical characterization of $\mathcal{I}$-Markov equivalence classes, which will be crucial in the proof of Theorem 4.1.

[3] Note that Proposition 4.1 holds even for distributions with densities which are not strictly positive.

[4] Yang et al. [2018] defines $\mathcal{M}_{\mathcal{I}}(\mathcal{G})$ slightly differently, but show their definition to be equivalent to the one used here. See Lemma A.1 in Yang et al. [2018]

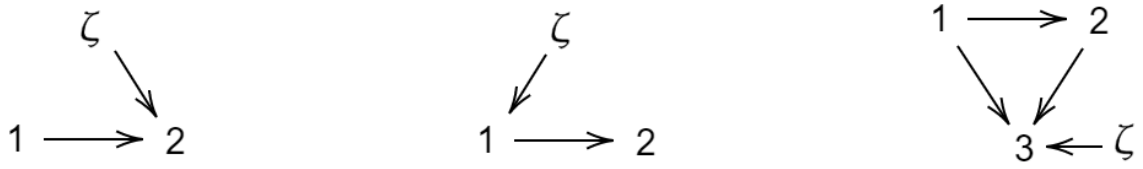


Figure 4.5. Different $\mathcal{I}$-DAGs with a single intervention. The first graph is alone in its $\mathcal{I}$-Markov equivalence class since reversing the $1 \to 2$ edge would break the immorality $1 \to 2 \leftarrow \zeta$. The second graph is also alone in its equivalence class since reversing $1 \to 2$ would create a new immorality $\zeta \to 1 \leftarrow 2$. The third DAG is not alone in its equivalence class since reversing $1 \to 2$ would preserve the skeleton without adding or removing an immorality. It should become apparent that adding more interventions will likely reduce the size of the $\mathcal{I}$-Markov equivalence class by introducing more immoralities.

Theorem 4.3. *(Yang et al. [2018]) Suppose the interventional family $\mathcal{I}$ is such that $I_1 := \emptyset$. Two DAGs $\mathcal{G}_1$ and $\mathcal{G}_2$ are $\mathcal{I}$-Markov equivalent iff their $\mathcal{I}$-DAGs $\mathcal{G}_1^{\mathcal{I}}$ and $\mathcal{G}_2^{\mathcal{I}}$ share the same skeleton and immoralities.*

See Figure 4.5 for a simple illustration of this concept.

We now present a very simple corollary which gives a situation where the $\mathcal{I}$-Markov equivalence class contains a unique graph.

Corollary 4.1. *Let $\mathcal{G}$ be a DAG and let $\mathcal{I} = (\emptyset, \{1\}, \cdots, \{d\})$. Then $\mathcal{G}$ is alone in its $\mathcal{I}$-Markov equivalence class.*

Proof. By Theorem 4.3, all $\mathcal{I}$-Markov equivalent graphs will share its skeleton with $\mathcal{G}$, so we consider only graphs obtained by reversing edges in $\mathcal{G}$.

Consider any edge $i \to j$ in $\mathcal{G}$. We note that $i \to j \leftarrow \zeta_{j+1}$ forms an immorality in the $\mathcal{I}$-DAG $\mathcal{G}^{\mathcal{I}}$. Reversing $i \to j$ would break this immorality which would imply that the resulting DAG is not $\mathcal{I}$-Markov equivalent to $\mathcal{G}$, by Theorem 4.3. Hence, $\mathcal{G}$ is alone in its equivalence class.■

A.2. Proof of Theorem 4.1

We are now ready to present the main result of this section. We recall the score function introduced in Section 4.3.1:

$$\mathcal{S}_{\mathcal{I}^*}(\mathcal{G}) := \sup_{\boldsymbol{\phi}} \sum_{k=1}^{K} \mathbb{E}_{\boldsymbol{x} \sim p^{(k)}} \log f^{(k)}(\boldsymbol{x}; \boldsymbol{M}^{\mathcal{G}}, \boldsymbol{R}^{\mathcal{I}^*}, \boldsymbol{\phi}) - \lambda |\mathcal{G}| \,, \tag{4.14}$$

where

$$f^{(k)}(\boldsymbol{x}; \boldsymbol{M}^{\mathcal{G}}, \boldsymbol{R}^{\mathcal{I}}, \boldsymbol{\phi}) := \prod_{j=1}^{d} \tilde{f}(\boldsymbol{x}_j; \mathrm{NN}(\boldsymbol{M}_j^{\mathcal{G}} \odot \boldsymbol{x}; \boldsymbol{\phi}_j^{(1)}))^{1-\boldsymbol{R}_{kj}^{\mathcal{I}}} \tilde{f}(\boldsymbol{x}_j; \mathrm{NN}(\boldsymbol{M}_j^{\mathcal{G}} \odot \boldsymbol{x}; \boldsymbol{\phi}_j^{(k)}))^{\boldsymbol{R}_{kj}^{\mathcal{I}}} . \tag{4.15}$$

Recall that $(p^{(k)})_{k\in[K]}$ are the ground truth interventional distributions with ground truth graph $\mathcal{G}^*$ and ground truth interventional family $\mathcal{I}^*$. We will sometimes use the notation $f^{(k)}_{\mathcal{GI}\phi}(\boldsymbol{x})$ to refer to $f^{(k)}(\boldsymbol{x}; \boldsymbol{M}^{\mathcal{G}}, \boldsymbol{R}^{\mathcal{I}}, \boldsymbol{\phi})$. We define $\mathcal{F}_{\mathcal{I}}(\mathcal{G})$ to be the set of all $(f^{(k)})_{k\in[K]}$ which are expressible by the model specified in Equation (4.15). More precisely,

$$\mathcal{F}_{\mathcal{I}}(\mathcal{G}) := \{(f^{(k)})_{k\in[K]} \mid \exists\, \boldsymbol{\phi} \text{ s.t. } \forall\, k \in [K]\ f^{(k)} = f^{(k)}_{\mathcal{GI}\phi}\}\,. \tag{4.16}$$

Theorem 4.1 relies on four assumptions. The first one requires that the model is expressive enough to represent the ground truth distributions exactly.

Assumption 4.1 (Sufficient capacity)**.** *The ground truth interventional distributions $\mathbb{P}^{(k)}$ all have a density $p^{(k)}$ w.r.t. the Lebesgue measure on $\mathbb{R}^n$ such that $(p^{(k)})_{k\in[K]} \in \mathcal{F}_{\mathcal{I}^*}(\mathcal{G}^*)$, i.e. the model specified in Equation (4.15) is expressive enough to represent the ground truth distributions.*

The second assumption is a generalization of faithfulness to interventions.

Assumption 4.2 ($\mathcal{I}^*$-Faithfulness)**.**

(1) For any disjoint $A, B, C \subseteq V$,

$$A \not\perp\!\!\!\perp_{\mathcal{G}^*} B|C \ \text{ implies } \ \boldsymbol{x}_A \not\perp\!\!\!\perp_{p^{(1)}} \boldsymbol{x}_B|\boldsymbol{x}_C\,.$$

(2) For any disjoint $A, C \subseteq V$ and $k \in [K]$,

$$A \not\perp\!\!\!\perp_{\mathcal{G}^{*\mathcal{I}^*}} \zeta_k \mid C \cup \zeta_{-k} \text{ implies } p^{(k)}\left(\boldsymbol{x}_A|\boldsymbol{x}_C\right) \neq p^{(1)}\left(\boldsymbol{x}_A|\boldsymbol{x}_C\right).$$

The first condition of Assumption 4.2 is exactly the standard faithfulness assumption for the ground truth observational distribution. The second condition is simply the converse of the second condition in the $\mathcal{I}$-Markov property (Definition 4.5) and can be understood as avoiding pathological interventions to make sure that every variables that can be potentially affected by the intervention are indeed affected. The simplest case is when $I_k := \{j\}$, $A := \{j\}$ and $C := \pi_j^{\mathcal{G}^*}$. In this case the condition requires that the intervention actually change something. Another simple case is when $C := \emptyset$. In this case, the condition requires that all descendants are affected, in the sense that their marginals change.

As we just saw, a trivial violation of $\mathcal{I}^*$-faithfulness would be when the intervention is not changing anything, not even the targeted conditional. We now present a non-trivial violation of $\mathcal{I}^*$-faithfulness.

Example 4.1 ($\mathcal{I}^*$-Faithfulness violation)**.** *Suppose $\mathcal{G}^*$ is $\boldsymbol{x}_1 \to \boldsymbol{x}_2$ where both variables are binary. Assume $p^{(1)}(\boldsymbol{x}_1 = 1) = \frac{1}{2}$, $p^{(1)}(\boldsymbol{x}_2 = 1 \mid \boldsymbol{x}_1 = 0) = \frac{1}{4}$ and $p^{(1)}(\boldsymbol{x}_2 = 1 \mid \boldsymbol{x}_1 = 1) = \frac{3}{4}$. From this, we can compute $p^{(1)}(\boldsymbol{x}_2 = 1) = \frac{1}{2}$. Consider the intervention targeting only $\boldsymbol{x}_2$ which changes its conditional to $p^{(2)}(\boldsymbol{x}_2 = 1 \mid \boldsymbol{x}_1 = 0) = \frac{3}{4}$ and $p^{(2)}(\boldsymbol{x}_2 = 1 \mid \boldsymbol{x}_1 = 1) = \frac{1}{4}$. So the interventional family is $\mathcal{I}^* = (\emptyset, \{2\})$. A simple computation shows the new marginal on $\boldsymbol{x}_2$ has not changed, i.e. $p^{(2)}(\boldsymbol{x}_2) = p^{(1)}(\boldsymbol{x}_2)$. This is a violation of $\mathcal{I}^*$-faithfulness since clearly $\boldsymbol{x}_2$ is not d-separated from the interventional node ζ_2 in $\mathcal{G}^{*\mathcal{I}^*}$.*

The third assumption is a technicality to simplify the presentation of the proofs and to follow the presentation of Yang et al. [2018]: we require the density model to be strictly positive.

Assumption 4.3 (Strict positivity)**.** *For all $k \in [K]$, the model density $f^{(k)}(\boldsymbol{x}; \boldsymbol{M}^{\mathcal{G}}, \boldsymbol{R}^{\mathcal{I}}, \boldsymbol{\phi})$ is strictly positive for all $\boldsymbol{\phi}$, DAG $\mathcal{G}$ and interventional family $\mathcal{I}$.*

Note that Assumption 4.3 is satisfied for example when for all $\boldsymbol{\theta}$ in the image of NN, the density $\tilde{f}(\cdot; \boldsymbol{\theta})$ is strictly positive. This happens when using a Gaussian density with variance strictly positive or a deep sigmoidal flow.

From Equation (4.16) and Assumption 4.3, it should be clear that $\mathcal{F}_{\mathcal{I}}(\mathcal{G}) \subseteq \mathcal{M}_{\mathcal{I}}(\mathcal{G})$ (recall $\mathcal{M}_{\mathcal{I}}(\mathcal{G})$ contains only strictly positive densities). Thus, from Proposition 4.2 we see that the $\mathcal{I}$-Markov property holds for all $(f^{(k)})_{k\in[K]} \in \mathcal{F}_{\mathcal{I}}(\mathcal{G})$. This fact will be useful in the proof of Theorem 4.1.

The fourth assumption is purely technical. It requires the differential entropy of the densities $p^{(k)}$ to be finite, which, as we will see in Lemma 4.1, ensures that the score of the ground truth graph $\mathcal{S}_{\mathcal{I}^*}(\mathcal{G}^*)$ is finite. This will be important to ensure that the score of any other graphs can be compared to it. In particular, this is avoiding the hypothetical situation where $\mathcal{S}_{\mathcal{I}^*}(\mathcal{G}^*)$ and $\mathcal{S}_{\mathcal{I}^*}(\mathcal{G})$ are both equal to infinity, which means they cannot be easily compared without defining a specific limiting process.

Assumption 4.4 (Finite differential entropies)**.** *For all $k \in [K]$,*

$$|\mathbb{E}_{p^{(k)}} \log p^{(k)}(\boldsymbol{x})| < \infty \,.$$

Lemma 4.1 (Finite scores)**.** *Under Assumptions 4.1 & 4.4, $|\mathcal{S}_{\mathcal{I}^*}(\mathcal{G}^*)| < \infty$.*

Proof. Consider the Kullback-Leibler divergence between $p^{(k)}$ and $f^{(k)}_{\mathcal{G}^*\mathcal{I}^*\boldsymbol{\phi}}$ for an arbitrary $\boldsymbol{\phi}$.

$$0 \leq D_{KL}(p^{(k)} || f^{(k)}_{\mathcal{G}^*\mathcal{I}^*\boldsymbol{\phi}}) = \mathbb{E}_{p^{(k)}} \log p^{(k)}(\boldsymbol{x}) - \mathbb{E}_{p^{(k)}} \log f^{(k)}_{\mathcal{G}^*\mathcal{I}^*\boldsymbol{\phi}}(\boldsymbol{x}) \,, \tag{4.17}$$

where we applied the linearity of the expectation (which holds because $|\mathbb{E}_{p^{(k)}} \log p^{(k)}(\boldsymbol{x})| < \infty$). We thus have that

$$\mathbb{E}_{p^{(k)}} \log f^{(k)}_{\mathcal{G}^*\mathcal{I}^*\boldsymbol{\phi}}(\boldsymbol{x}) \leq \mathbb{E}_{p^{(k)}} \log p^{(k)}(\boldsymbol{x}) < \infty \,. \tag{4.18}$$

Thus, $\sup_{\boldsymbol{\phi}} \mathbb{E}_{p^{(k)}} \log f^{(k)}_{\mathcal{G}^*\mathcal{I}^*\boldsymbol{\phi}}(\boldsymbol{x}) < \infty$, which implies $\mathcal{S}_{\mathcal{I}^*}(\mathcal{G}^*) < \infty$.

By the assumption of sufficient capacity, there exists some $\boldsymbol{\phi}^*$ such that $f^{(k)}_{\mathcal{G}^*\boldsymbol{\phi}^*} = p^{(k)}$ for all k, hence $\sup_{\boldsymbol{\phi}} \sum_{k=1}^{K} \mathbb{E}_{p^{(k)}} \log f^{(k)}_{\mathcal{G}^*\mathcal{I}^*\boldsymbol{\phi}}(\boldsymbol{x}) \geq \sum_{k=1}^{K} \mathbb{E}_{p^{(k)}} \log f^{(k)}_{\mathcal{G}^*\boldsymbol{\phi}^*}(\boldsymbol{x}) = \sum_{k=1}^{K} \mathbb{E}_{p^{(k)}} \log p^{(k)}(\boldsymbol{x}) > -\infty$. This implies that $\mathcal{S}_{\mathcal{I}^*}(\mathcal{G}^*) > -\infty$. ■

The next lemma shows that the difference $\mathcal{S}_{\mathcal{I}^*}(\mathcal{G}^*) - \mathcal{S}_{\mathcal{I}^*}(\mathcal{G})$ can be rewritten as a minimization of a sum of KL divergences plus the difference in regularizing terms.

Lemma 4.2 (Rewriting of score differences)**.** *Under Assumption 4.1 & 4.4, we have*

$$\mathcal{S}_{\mathcal{I}^*}(\mathcal{G}^*) - \mathcal{S}_{\mathcal{I}^*}(\mathcal{G}) = \inf_{\boldsymbol{\phi}} \sum_{k\in[K]} D_{KL}(p^{(k)}||f^{(k)}_{\mathcal{G}\mathcal{I}^*\boldsymbol{\phi}}) + \lambda(|\mathcal{G}| - |\mathcal{G}^*|)\,. \tag{4.19}$$

Proof. By Lemma 4.1, we have that $|\mathcal{S}_{\mathcal{I}^*}(\mathcal{G}^*)| < \infty$, which ensures the difference $\mathcal{S}_{\mathcal{I}^*}(\mathcal{G}^*) - \mathcal{S}_{\mathcal{I}^*}(\mathcal{G})$ is well defined.

$$\mathcal{S}_{\mathcal{I}^*}(\mathcal{G}^*) - \mathcal{S}_{\mathcal{I}^*}(\mathcal{G}) \tag{4.20}$$

$$= \mathcal{S}_{\mathcal{I}^*}(\mathcal{G}^*) - \sum_{k\in[K]} \mathbb{E}_{p^{(k)}} \log p^{(k)}(\boldsymbol{x}) - \mathcal{S}_{\mathcal{I}^*}(\mathcal{G}) + \sum_{k\in[K]} \mathbb{E}_{p^{(k)}} \log p^{(k)}(\boldsymbol{x}) \tag{4.21}$$

$$\begin{aligned} = &\sup_{\boldsymbol{\phi}} \sum_{k\in[K]} \mathbb{E}_{p^{(k)}} \log f^{(k)}_{\mathcal{G}^*\mathcal{I}^*\boldsymbol{\phi}}(\boldsymbol{x}) - \sum_{k\in[K]} \mathbb{E}_{p^{(k)}} \log p^{(k)}(\boldsymbol{x}) \\ &- \sup_{\boldsymbol{\phi}} \sum_{k\in[K]} \mathbb{E}_{p^{(k)}} \log f^{(k)}_{\mathcal{G}\mathcal{I}^*\boldsymbol{\phi}}(\boldsymbol{x}) + \sum_{k\in[K]} \mathbb{E}_{p^{(k)}} \log p^{(k)}(\boldsymbol{x}) \\ &+ \lambda(|\mathcal{G}| - |\mathcal{G}^*|) \end{aligned} \tag{4.22}$$

$$\begin{aligned} = &\inf_{\boldsymbol{\phi}} - \sum_{k\in[K]} \mathbb{E}_{p^{(k)}} \log f^{(k)}_{\mathcal{G}\mathcal{I}^*\boldsymbol{\phi}}(\boldsymbol{x}) + \sum_{k\in[K]} \mathbb{E}_{p^{(k)}} \log p^{(k)}(\boldsymbol{x}) \\ &- \inf_{\boldsymbol{\phi}} - \sum_{k\in[K]} \mathbb{E}_{p^{(k)}} \log f^{(k)}_{\mathcal{G}^*\mathcal{I}^*\boldsymbol{\phi}}(\boldsymbol{x}) - \sum_{k\in[K]} \mathbb{E}_{p^{(k)}} \log p^{(k)}(\boldsymbol{x}) \\ &+ \lambda(|\mathcal{G}| - |\mathcal{G}^*|) \end{aligned} \tag{4.23}$$

$$\begin{aligned} = &\inf_{\boldsymbol{\phi}} \sum_{k\in[K]} D_{KL}(p^{(k)}||f^{(k)}_{\mathcal{G}\mathcal{I}^*\boldsymbol{\phi}}) - \inf_{\boldsymbol{\phi}} \sum_{k\in[K]} D_{KL}(p^{(k)}||f^{(k)}_{\mathcal{G}^*\mathcal{I}^*\boldsymbol{\phi}}) \\ &+ \lambda(|\mathcal{G}| - |\mathcal{G}^*|) \end{aligned} \tag{4.24}$$

The first equality holds since by Assumption 4.4 the differential entropy of $p^{(k)}$ is finite for all k. In (4.24), we use the linearity of the expectation, which holds because the entropy term is finite. By Assumption 4.1, $(p^{(k)})_{k\in[K]} \in \mathcal{F}_{\mathcal{I}^*}(\mathcal{G}^*)$ which implies that $\inf_{\boldsymbol{\phi}} \sum_{k\in[K]} D_{KL}(p^{(k)}||f^{(k)}_{\mathcal{G}^*\mathcal{I}^*\boldsymbol{\phi}}) = 0$.■

We will now prove three technical lemmas (Lemma 4.3, 4.4 & 4.5). Their proof can be safely skipped during a first reading.

Lemma 4.3 is adapted from Koller and Friedman [2009, Theorem 8.7] to handle cases where infinite differential entropies might arise.

Lemma 4.3. *Let $\mathcal{G}$ be a DAG. If $p \notin \mathcal{M}(\mathcal{G})$ and $p(x) > 0$ for all $\boldsymbol{x} \in \mathbb{R}^d$, then*

$$\inf_{f\in\mathcal{M}(\mathcal{G})} D_{KL}(p||f) > 0\,.$$

Proof. We consider a new density function defined as

$$\hat{f}(\boldsymbol{x}) := \prod_{j=1}^{d} p(\boldsymbol{x}_j \mid \boldsymbol{x}_{\pi_j^{\mathcal{G}}}) \,, \tag{4.25}$$

where

$$p(\boldsymbol{x}_j|\boldsymbol{x}_{\pi_j^{\mathcal{G}}}) := \frac{p(\boldsymbol{x}_j, \boldsymbol{x}_{\pi_j^{\mathcal{G}}})}{p(\boldsymbol{x}_{\pi_j^{\mathcal{G}}})} \,, \tag{4.26}$$

i.e. it is the conditional density. This should not be conflated with $p(\boldsymbol{x}_j|\boldsymbol{x}_{\pi_j^{\mathcal{G}^*}})$. It should be clear from (4.25) and the fact that p is strictly positive that $\hat{f} \in \mathcal{M}(\mathcal{G})$ hence $p \neq \hat{f}$. We will show that $\hat{f} \in \arg\min_{f \in \mathcal{M}(\mathcal{G})} D_{KL}(p||f)$.

Pick an arbitrary $f \in \mathcal{M}(\mathcal{G})$. We first show that $\mathbb{E}_p \log \frac{\hat{f}(\boldsymbol{x})}{f(\boldsymbol{x})}$ can be written as a sum of KL divergences.

$$\mathbb{E}_p \log \frac{\hat{f}(\boldsymbol{x})}{f(\boldsymbol{x})} = \mathbb{E}_p \sum_{j=1}^{d} \log \frac{p(\boldsymbol{x}_j|\boldsymbol{x}_{\pi_j^{\mathcal{G}}})}{f(\boldsymbol{x}_j|\boldsymbol{x}_{\pi_j^{\mathcal{G}}})} \tag{4.27}$$

$$= \sum_{j=1}^{d} \mathbb{E}_p \log \frac{p(\boldsymbol{x}_j|\boldsymbol{x}_{\pi_j^{\mathcal{G}}})}{f(\boldsymbol{x}_j|\boldsymbol{x}_{\pi_j^{\mathcal{G}}})} \tag{4.28}$$

In Equation (4.28), we apply the linearity of the Lebesgue integral, which holds as long as we are not summing infinities of opposite signs (in which case the sum is undefined).[5] We now show that it is not the case since each term is an expectation of a KL divergence, which is in $[0, +\infty]$:

$$\mathbb{E}_p \log \frac{p(\boldsymbol{x}_j|\boldsymbol{x}_{\pi_j^{\mathcal{G}}})}{f(\boldsymbol{x}_j|\boldsymbol{x}_{\pi_j^{\mathcal{G}}})} = \int p(\boldsymbol{x}_{\pi_j^{\mathcal{G}}}) \int p(\boldsymbol{x}_j \mid \boldsymbol{x}_{\pi_j^{\mathcal{G}}}) \log \frac{p(\boldsymbol{x}_j|\boldsymbol{x}_{\pi_j^{\mathcal{G}}})}{f(\boldsymbol{x}_j|\boldsymbol{x}_{\pi_j^{\mathcal{G}}})} d\boldsymbol{x}_j d\boldsymbol{x}_{\pi_j^{\mathcal{G}}} \tag{4.29}$$

$$= \int p(\boldsymbol{x}_{\pi_j^{\mathcal{G}}}) D_{KL}(p(\cdot_j \mid \boldsymbol{x}_{\pi_j^{\mathcal{G}}})||f(\cdot_j \mid \boldsymbol{x}_{\pi_j^{\mathcal{G}}}))) \,. \tag{4.30}$$

This implies that $\mathbb{E}_p \log \frac{\hat{f}(\boldsymbol{x})}{f(\boldsymbol{x})} \in [0, +\infty]$. We can now show that $\hat{f} \in \arg\min_{f \in \mathcal{M}(\mathcal{G})} D_{KL}(p||f)$:

[5] The linearity of the Lebesgue integral is typically stated for Lebesgue integrable functions f and g, i.e. $\int |f|, \int |g| < \infty$. See for example Billingsley [1995, Theorem 16.1]. However, it can be extended to cases where f and g are not integrable, as long as $\int f$ and $\int g$ are well-defined and are not infinities of opposite sign (which would yield the undefined expression $\infty - \infty$). The proof is a simple adaptation of Theorem 16.1 which makes use of Theorem 15.1 in Billingsley [1995].

$$D_{KL}(p||f) = \mathbb{E}_p \log \frac{p(\boldsymbol{x})}{\hat{f}(\boldsymbol{x})} \frac{\hat{f}(\boldsymbol{x})}{f(\boldsymbol{x})} \tag{4.31}$$

$$= \mathbb{E}_p \log \frac{p(\boldsymbol{x})}{\hat{f}(\boldsymbol{x})} + \mathbb{E}_p \log \frac{\hat{f}(\boldsymbol{x})}{f(\boldsymbol{x})} \tag{4.32}$$

$$= D_{KL}(p||\hat{f}) + \mathbb{E}_p \log \frac{\hat{f}(\boldsymbol{x})}{f(\boldsymbol{x})} \tag{4.33}$$

$$\geq D_{KL}(p||\hat{f}) > 0\,. \tag{4.34}$$

Equation (4.32) holds as long as we do not have $\infty - \infty$. It is not the case here since (i) the first term is a KL divergence, so it is in $[0, +\infty]$, and (ii) the second term was already shown to be in $[0, +\infty]$. The very last inequality holds because $p \neq \hat{f}$.

We conclude by noting that $\inf_{f \in \mathcal{M}(\mathcal{G})} D_{KL}(p||f) = D_{KL}(p||\hat{f}) > 0$. ■

The following lemma will make use of the following definition:

$$\mathcal{Z}(j, A) := \{(f^{(1)}, f^{(2)}) \mid f^{(1)}(\boldsymbol{x}_j \mid \boldsymbol{x}_A) = f^{(2)}(\boldsymbol{x}_j \mid \boldsymbol{x}_A) \text{ and } f^{(1)}, f^{(2)} > 0\}\,. \tag{4.35}$$

Lemma 4.4. *Let $j \in V$ and $A \subseteq V \setminus \{j\}$. If $(p^{(1)}, p^{(2)}) \notin \mathcal{Z}(j, A)$ and both $p^{(1)}$ and $p^{(2)}$ are strictly positive, then*

$$\inf_{(f^{(1)}, f^{(2)}) \in \mathcal{Z}(j,A)} D_{KL}(p^{(1)}||f^{(1)}) + D_{KL}(p^{(2)}||f^{(2)}) > 0\,.$$

Proof. The proof is very similar in spirit to the proof of Lemma 4.3.

We define new density functions:

$$p^{\text{mid}}(\boldsymbol{x}) := \frac{p^{(1)}(\boldsymbol{x}) + p^{(2)}(\boldsymbol{x})}{2} \tag{4.36}$$

$$\hat{f}^{(k)}(\boldsymbol{x}) := p^{(k)}(\boldsymbol{x}_A) p^{\text{mid}}(\boldsymbol{x}_j \mid \boldsymbol{x}_A) p^{(k)}(\boldsymbol{x}_{V \setminus A \setminus j} \mid \boldsymbol{x}_{A \cup j}) \;\; \forall k \in \{1, 2\}\,. \tag{4.37}$$

We note that p^{mid}, $\hat{f}^{(1)}$ and $\hat{f}^{(2)}$ are strictly positive since $p^{(1)}$ and $p^{(2)}$ are strictly positive. By construction, we have $\hat{f}^{(1)}(\boldsymbol{x}_j|\boldsymbol{x}_A) = \hat{f}^{(2)}(\boldsymbol{x}_j|\boldsymbol{x}_A)$, and thus $(\hat{f}^{(1)}, \hat{f}^{(2)}) \in \mathcal{Z}(i, A)$. This means that $\hat{f}^{(1)} \neq p^{(1)}$ or $\hat{f}^{(2)} \neq p^{(2)}$.

Pick an arbitrary $(f^{(1)}, f^{(2)}) \in \mathcal{Z}(j, A)$. We start by showing that the integral $\int p^{(1)}(\boldsymbol{x}) \log \frac{\hat{f}^{(1)}(\boldsymbol{x})}{f^{(1)}(\boldsymbol{x})} + p^{(2)}(\boldsymbol{x}) \log \frac{\hat{f}^{(2)}(\boldsymbol{x})}{f^{(2)}(\boldsymbol{x})} d\boldsymbol{x}$ is in $[0, +\infty]$.

$$\int p^{(1)}(\boldsymbol{x}) \log \frac{\hat{f}^{(1)}(\boldsymbol{x})}{f^{(1)}(\boldsymbol{x})} + p^{(2)}(\boldsymbol{x}) \log \frac{\hat{f}^{(2)}(\boldsymbol{x})}{f^{(2)}(\boldsymbol{x})} d\boldsymbol{x} \tag{4.38}$$

$$= \int p^{(1)}(\boldsymbol{x}) \left[\log \frac{p^{(1)}(\boldsymbol{x}_A)}{f^{(1)}(\boldsymbol{x}_A)} + \log \frac{p^{\text{mid}}(\boldsymbol{x}_j \mid \boldsymbol{x}_A)}{f^{(1)}(\boldsymbol{x}_j \mid \boldsymbol{x}_A)} + \log \frac{p^{(1)}(\boldsymbol{x}_{V \setminus A \setminus j} \mid \boldsymbol{x}_{A \cup j})}{f^{(1)}(\boldsymbol{x}_{V \setminus A \setminus j} \mid \boldsymbol{x}_{A \cup j})} \right]$$
$$+ p^{(2)}(\boldsymbol{x}) \left[\log \frac{p^{(2)}(\boldsymbol{x}_A)}{f^{(2)}(\boldsymbol{x}_A)} + \log \frac{p^{\text{mid}}(\boldsymbol{x}_j \mid \boldsymbol{x}_A)}{f^{(1)}(\boldsymbol{x}_j \mid \boldsymbol{x}_A)} + \log \frac{p^{(2)}(\boldsymbol{x}_{V \setminus A \setminus j} \mid \boldsymbol{x}_{A \cup j})}{f^{(2)}(\boldsymbol{x}_{V \setminus A \setminus j} \mid \boldsymbol{x}_{A \cup j})} \right] d\boldsymbol{x} \tag{4.39}$$

$$= D_{KL}(p^{(1)}(\cdot_A) || f^{(1)}(\cdot_A)) + \mathbb{E}_{p^{(1)}} D_{KL}(p^{(1)}(\cdot_{V \setminus A \setminus j} \mid \boldsymbol{x}_{A \cup j}) || f^{(1)}(\cdot_{V \setminus A \setminus j} \mid \boldsymbol{x}_{A \cup j}))$$
$$+ D_{KL}(p^{(2)}(\cdot_A) || f^{(2)}(\cdot_A)) + \mathbb{E}_{p^{(2)}} D_{KL}(p^{(2)}(\cdot_{V \setminus A \setminus j} \mid \boldsymbol{x}_{A \cup j}) || f^{(2)}(\cdot_{V \setminus A \setminus j} \mid \boldsymbol{x}_{A \cup j}))$$
$$+ 2 \underbrace{\int \frac{p^{(1)}(\boldsymbol{x}) + p^{(2)}(\boldsymbol{x})}{2} \log \frac{p^{\text{mid}}(\boldsymbol{x}_j \mid \boldsymbol{x}_A)}{f^{(1)}(\boldsymbol{x}_j \mid \boldsymbol{x}_A)} d\boldsymbol{x}}_{= \mathbb{E}_{p^{\text{mid}}} D_{KL}(p^{\text{mid}}(\cdot_j | \boldsymbol{x}_A) || f^{(1)}(\cdot_j | \boldsymbol{x}_A))} . \tag{4.40}$$

In (4.39), we used the fact that $f^{(1)}(\boldsymbol{x}_j \mid \boldsymbol{x}_A) = f^{(2)}(\boldsymbol{x}_j \mid \boldsymbol{x}_A)$. In (4.40), we use the linearity of the integral (which can be safely apply because each resulting "piece" is in $[0, +\infty]$). Since each term in (4.40) is in $[0, +\infty]$, their sum is in $[0, +\infty]$ as well.

We can now look at the sum of KL-divergences we are interested in.

$$D_{KL}(p^{(1)} || f^{(1)}) + D_{KL}(p^{(2)} || f^{(2)})$$
$$= \int p^{(1)}(\boldsymbol{x}) \log \frac{p^{(1)}}{f^{(1)}} d\boldsymbol{x} + \int p^{(2)}(\boldsymbol{x}) \log \frac{p^{(2)}}{f^{(2)}} d\boldsymbol{x} \tag{4.41}$$
$$= \int p^{(1)}(\boldsymbol{x}) \log \frac{p^{(1)}}{f^{(1)}} + p^{(2)}(\boldsymbol{x}) \log \frac{p^{(2)}}{f^{(2)}} d\boldsymbol{x} \tag{4.42}$$
$$= \int p^{(1)}(\boldsymbol{x}) \log \frac{p^{(1)}(\boldsymbol{x})}{\hat{f}^{(1)}(\boldsymbol{x})} + p^{(1)}(\boldsymbol{x}) \log \frac{\hat{f}^{(1)}(\boldsymbol{x})}{f^{(1)}(\boldsymbol{x})} + p^{(2)}(\boldsymbol{x}) \log \frac{p^{(2)}(\boldsymbol{x})}{\hat{f}^{(2)}(\boldsymbol{x})} + p^{(2)}(\boldsymbol{x}) \log \frac{\hat{f}^{(2)}(\boldsymbol{x})}{f^{(2)}(\boldsymbol{x})} d\boldsymbol{x} \tag{4.43}$$
$$= D_{KL}(p^{(1)} || \hat{f}^{(1)}) + D_{KL}(p^{(2)} || \hat{f}^{(2)}) + \int p^{(1)}(\boldsymbol{x}) \log \frac{\hat{f}^{(1)}(\boldsymbol{x})}{f^{(1)}(\boldsymbol{x})} + p^{(2)}(\boldsymbol{x}) \log \frac{\hat{f}^{(2)}(\boldsymbol{x})}{f^{(2)}(\boldsymbol{x})} d\boldsymbol{x} \tag{4.44}$$
$$\geq D_{KL}(p^{(1)} || \hat{f}^{(1)}) + D_{KL}(p^{(2)} || \hat{f}^{(2)}) > 0 . \tag{4.45}$$

In (4.42), we use the linearity of the integral (which can be safely applied given the initial integrals were in $[0, +\infty]$). In (4.44), we again use the linearity of the integral (which is, again, possible because each resulting piece are in $[0, +\infty]$). In (4.45), we use the fact that $\int p^{(1)}(\boldsymbol{x}) \log \frac{\hat{f}^{(1)}(\boldsymbol{x})}{f^{(1)}(\boldsymbol{x})} + p^{(2)}(\boldsymbol{x}) \log \frac{\hat{f}^{(2)}(\boldsymbol{x})}{f^{(2)}(\boldsymbol{x})} d\boldsymbol{x} \in [0, +\infty]$ to get the $\geq$ while the strict inequality holds because either $\hat{f}^{(1)} \neq p^{(1)}$ or $\hat{f}^{(k)} \neq p^{(k)}$.

This implies that

$$\inf_{(f^{(1)}, f^{(2)}) \in \mathcal{Z}(j, A)} D_{KL}(p^{(1)} || f^{(1)}) + D_{KL}(p^{(2)} || f^{(2)}) = D_{KL}(p^{(1)} || \hat{f}^{(1)}) + D_{KL}(p^{(2)} || \hat{f}^{(2)}) > 0 . \blacksquare$$

The following definition will be useful for the next lemma.

Definition 4.6. *Given a DAG $\mathcal{G}$ with node set V and two nodes $i, j \in V$, we define the following sets:*

$$T_{ij}^{\mathcal{G}} := \{\ell \in V \mid \textit{the immorality } i \to \ell \leftarrow j \textit{ is in } \mathcal{G}\} \tag{4.46}$$

$$L_{ij}^{\mathcal{G}} := \mathbf{DE}_{\mathcal{G}}(T_{ij}^{\mathcal{G}}) \cup \{i, j\}\,, \tag{4.47}$$

where $\mathbf{DE}_{\mathcal{G}}(S)$ is the set of descendants of S in $\mathcal{G}$, including S itself.

Lemma 4.5. *Let $\mathcal{G}$ be a DAG with node set V. When $i \to j \notin \mathcal{G}$ and $i \leftarrow j \notin \mathcal{G}$ we have*

$$i \perp\!\!\!\perp_{\mathcal{G}} j \mid V \setminus L_{ij}^{\mathcal{G}}\,. \tag{4.48}$$

Proof: By contradiction. Suppose there is a path from $(i = a_0, a_1, ..., a_p = j)$ with $p > 1$ which is not d-blocked by $V \setminus L_{ij}^{\mathcal{G}}$ in $\mathcal{G}$. We first consider the case where the path contains no colliders.

If the path contains no colliders, then $a_0 \leftarrow a_1$ or $a_{p-1} \to a_p$. Moreover, since the path is not d-blocked and both a_1 and a_{p-1} are not colliders, $a_1, a_{p-1} \in L_{ij}^{\mathcal{G}}$. But this implies that there is a directed path from $i = a_0$ to a_1 and a directed path from $j = a_p$ to a_{p-1}. This creates a directed cycle: either $a_0 \to \cdots \to a_1 \to a_0$ or $a_p \to \cdots \to a_{p-1} \to a_p$. This is a contradiction since $\mathcal{G}$ is acyclic.

Suppose there is a collider a_k, i.e. $a_{k-1} \to a_k \leftarrow a_{k+1}$. Since the path is not d-blocked, there must exists a node $z \in \mathbf{DE}_{\mathcal{G}}(a_k) \cup \{a_k\}$ such that $z \notin L_{ij}^{\mathcal{G}}$. If $i = a_{k-1}$ and $j = a_{k+1}$, then clearly $z \in L_{ij}^{\mathcal{G}}$, which is a contradiction. Otherwise, $i \neq a_{k-1}$ or $j \neq a_{k+1}$. Without loss of generality, assume $i \neq a_{k-1}$. Clearly, a_{k-1} is not a collider and since the path is not d-blocked, $a_{k-1} \in L_{ij}^{\mathcal{G}}$. But by definition, $L_{ij}^{\mathcal{G}}$ also contains all the descendants of a_{k-1} including z. Again, this is a contradiction with $z \notin L_{ij}^{\mathcal{G}}$. ■

We recall Theorem 1 from Section 4.3.1 and present its proof.

Theorem 4.1 (Identification via score maximization)**.** *Suppose the interventional family $\mathcal{I}^*$ is such that $I_1^* := \emptyset$. Let $\mathcal{G}^*$ be the ground truth DAG and $\hat{\mathcal{G}} \in \arg\max_{\mathcal{G} \in DAG} \mathcal{S}_{\mathcal{I}^*}(\mathcal{G})$. Assume that the density model has enough capacity to represent the ground truth distributions, that $\mathcal{I}^*$-faithfulness holds, that the density model is strictly positive and that the ground truth densities $p^{(k)}$ have finite differential entropy, respectively Assumptions 4.1, 4.2, 4.3 & 4.4. Then for $\lambda > 0$ small enough, we have that $\hat{\mathcal{G}}$ is $\mathcal{I}^*$-Markov equivalent to $\mathcal{G}^*$.*

Proof. It is sufficient to prove that, for all $\mathcal{G} \notin \mathcal{I}^*\text{-MEC}(\mathcal{G}^*)$, $\mathcal{S}_{\mathcal{I}^*}(\mathcal{G}^*) > \mathcal{S}_{\mathcal{I}^*}(\mathcal{G})$. We use Theorem 4.3 which states that $\hat{\mathcal{G}}$ is not $\mathcal{I}^*$-Markov equivalent to $\mathcal{G}^*$ if and only if $\hat{\mathcal{G}}^{\mathcal{I}^*}$ does not share its skeleton or its immoralities with $\mathcal{G}^{*\mathcal{I}^*}$. The proof is organized in six cases. Cases 1-2 treat when $\mathcal{G}$ and $\mathcal{G}^*$ do not share the same skeleton, cases 3 & 4 when their immoralities differ and cases 5 & 6 when their immoralities implying interventional nodes ζ_k differ. In almost every cases, the idea is the same:

(1) Use Lemma 4.5 to find a d-separation which holds in $\mathcal{G}^{\mathcal{I}^*}$ and show it does not hold in $\mathcal{G}^{*\mathcal{I}^*}$;

(2) Use the fact that $\mathcal{F}_{\mathcal{I}}(\mathcal{G}) \subseteq \mathcal{M}_{\mathcal{I}}(\mathcal{G})$ (by strict positivity), Proposition 4.2 and the $\mathcal{I}^*$-faithfulness assumption to obtain an invariance which holds for all $(f^{(k)})_{k\in[K]} \in \mathcal{F}_{\mathcal{I}^*}(\mathcal{G})$ but not in $(p^{(k)})_{k\in[K]}$;

(3) Use the fact that the invariance forces $\inf_\phi \sum_{k\in[K]} D_{KL}(p^{(k)}||f^{(k)}_{\mathcal{G}\mathcal{I}^*\phi})$ to be greater than zero (by Lemma 4.3 or 4.4) and;

(4) Conclude that $\mathcal{S}_{\mathcal{I}^*}(\mathcal{G}^*) > \mathcal{S}_{\mathcal{I}^*}(\mathcal{G})$ via Lemma 4.2.

In this proof, we are exclusively referring to $\mathcal{I}^*$. Thus for notational convenience, we set $\mathcal{I} := \mathcal{I}^*$.

Case 1: We consider the graphs $\mathcal{G}$ such that there exists $i \to j \in \mathcal{G}^*$ but $i \to j \notin \mathcal{G}$ and $i \leftarrow j \notin \mathcal{G}$. Let $\mathbb{G}$ be the set of all such $\mathcal{G}$. By Lemma 4.5, $i \perp\!\!\!\perp_{\mathcal{G}} j \mid V \setminus L^{\mathcal{G}}_{ij}$ but clearly $i \not\perp\!\!\!\perp_{\mathcal{G}^*} j \mid V \setminus L^{\mathcal{G}}_{ij}$. Hence, by $\mathcal{I}$-faithfulness (Assumption 4.2) we have $\boldsymbol{x}_i \not\perp\!\!\!\perp_{p^{(1)}} \boldsymbol{x}_j | \boldsymbol{x}_{V\setminus L^{\mathcal{G}}_{ij}}$. It implies that $p^{(1)} \notin \mathcal{M}(\mathcal{G})$, by Proposition 4.1.

For notation convenience, let us define

$$\eta(\mathcal{G}) := \inf_{\phi} \sum_{k\in[K]} D_{KL}(p^{(k)}||f^{(k)}_{\mathcal{G}\mathcal{I}\phi}) \,. \tag{4.49}$$

Note that

$$\eta(\mathcal{G}) \geq \inf_{\phi} D_{KL}(p^{(1)}||f^{(1)}_{\mathcal{G}\mathcal{I}\phi}) \geq \inf_{f\in\mathcal{M}(\mathcal{G})} D_{KL}(p^{(1)}||f) > 0 \,, \tag{4.50}$$

where the first inequality holds by non-negativity of the KL divergence, the second holds because, for all ϕ, $f^{(1)}_{\mathcal{G}\mathcal{I}\phi} \in \mathcal{M}(\mathcal{G})$ and the third holds by Lemma 4.3 (which applies here because $p^{(1)} \notin \mathcal{M}(\mathcal{G})$). Using Lemma 4.2, we can write

$$\mathcal{S}_{\mathcal{I}}(\mathcal{G}^*) - \mathcal{S}_{\mathcal{I}}(\mathcal{G}) = \eta(\mathcal{G}) + \lambda(|\mathcal{G}| - |\mathcal{G}^*|) \,. \tag{4.51}$$

If $|\mathcal{G}| \geq |\mathcal{G}^*|$ then clearly $\mathcal{S}_{\mathcal{I}}(\mathcal{G}^*) - \mathcal{S}_{\mathcal{I}}(\mathcal{G}) > 0$. Let $\mathbb{G}^+ := \{\mathcal{G} \in \mathbb{G} \mid |\mathcal{G}| < |\mathcal{G}^*|\}$. To make sure we have $\mathcal{S}_{\mathcal{I}}(\mathcal{G}^*) - \mathcal{S}_{\mathcal{I}}(\mathcal{G}) > 0$ for all $\mathcal{G} \in \mathbb{G}^+$, we need to pick λ sufficiently small. Choosing $0 < \lambda < \min_{\mathcal{G}\in\mathbb{G}^+} \frac{\eta(\mathcal{G})}{|\mathcal{G}^*|-|\mathcal{G}|}$ is sufficient since (and note that minimum exists because the set $\mathbb{G}^+$ is finite and is strictly positive by (4.50)):

$$\lambda < \min_{\mathcal{G}\in\mathbb{G}^+} \frac{\eta(\mathcal{G})}{|\mathcal{G}^*| - |\mathcal{G}|} \tag{4.52}$$

$$\iff \lambda < \frac{\eta(\mathcal{G})}{|\mathcal{G}^*| - |\mathcal{G}|} \;\; \forall \mathcal{G} \in \mathbb{G}^+ \tag{4.53}$$

$$\iff \lambda(|\mathcal{G}^*| - |\mathcal{G}|) < \eta(\mathcal{G}) \;\; \forall \mathcal{G} \in \mathbb{G}^+ \tag{4.54}$$

$$\iff 0 < \eta(\mathcal{G}) + \lambda(|\mathcal{G}| - |\mathcal{G}^*|) = \mathcal{S}_{\mathcal{I}}(\mathcal{G}^*) - \mathcal{S}_{\mathcal{I}}(\mathcal{G}) \;\; \forall \mathcal{G} \in \mathbb{G}^+ \,. \tag{4.55}$$

Case 2: We consider the graphs $\mathcal{G}$ such that there exists $i \to j \in \mathcal{G}$ but $i \to j \notin \mathcal{G}^*$ and $i \leftarrow j \notin \mathcal{G}^*$. We can assume $k \to \ell \in \mathcal{G}^*$ implies $k \to \ell \in \mathcal{G}$ or $k \leftarrow \ell \in \mathcal{G}$, since otherwise we are in Case 1. Hence, it means $|\mathcal{G}| > |\mathcal{G}^*|$ which in turn implies that $\mathcal{S}_{\mathcal{I}}(\mathcal{G}^*) > \mathcal{S}_{\mathcal{I}}(\mathcal{G})$.

Cases 1 and 2 completely cover the situations where $\mathcal{G}^{\mathcal{I}}$ and $\mathcal{G}^{*\mathcal{I}}$ do not share the same skeleton. Next, we assume that $\mathcal{G}^{\mathcal{I}}$ and $\mathcal{G}^{*\mathcal{I}}$ do have the same skeleton (which implies that $|\mathcal{G}| = |\mathcal{G}^*|$). The remaining cases treat the differences in immoralities.

Case 3: Suppose $\mathcal{G}^*$ contains an immorality $i \to \ell \leftarrow j$ which is not present in $\mathcal{G}$. We first show that $\ell \notin L_{ij}^{\mathcal{G}}$. Suppose the opposite. This means ℓ is a descendant of both i and j in $\mathcal{G}$. Since $\mathcal{G}$ and $\mathcal{G}^*$ share skeleton and because $i \to \ell \leftarrow j$ is not an immorality in $\mathcal{G}$, we have that $i \leftarrow \ell \in \mathcal{G}$ or $\ell \to j \in \mathcal{G}$, which in both cases creates a cycle. This is a contradiction.

The path (i, ℓ, j) is not d-blocked by $V \setminus L_{ij}^{\mathcal{G}}$ in $\mathcal{G}^*$ since $\ell \in V \setminus L_{ij}^{\mathcal{G}}$. By $\mathcal{I}$-faithfulness (Assumption 4.2), this means that $\boldsymbol{x}_i \not\perp\!\!\!\perp_{p^{(1)}} \boldsymbol{x}_j \mid \boldsymbol{x}_{V \setminus L_{ij}^{\mathcal{G}}}$. Since $\mathcal{G}^*$ and $\mathcal{G}$ share the same skeleton, we know $i \to j$ and $i \leftarrow j$ are not in $\mathcal{G}$. Using Lemma 4.5, we have that $i \perp\!\!\!\perp_{\mathcal{G}} j \mid V \setminus L_{ij}^{\mathcal{G}}$. Hence by Proposition 4.1, $p^{(1)} \notin \mathcal{M}(\mathcal{G})$. Similarly to Case 1, this implies that $\eta(\mathcal{G}) > 0$ which in turn implies that $\mathcal{S}_{\mathcal{I}}(\mathcal{G}^*) - \mathcal{S}_{\mathcal{I}}(\mathcal{G}) > 0$ (using the fact $|\mathcal{G}^*| = |\mathcal{G}|$).

Case 4: Suppose $\mathcal{G}$ contains an immorality $i \to \ell \leftarrow j$ which is not present in $\mathcal{G}^*$. Since $\mathcal{G}$ and $\mathcal{G}^*$ share the same skeleton and $\ell \notin V \setminus L_{ij}^{\mathcal{G}}$, we know there is a (potentially undirected) path (i, ℓ, j) which is not d-blocked by $V \setminus L_{ij}^{\mathcal{G}}$ in $\mathcal{G}^*$. By $\mathcal{I}$-faithfulness (Assumption 4.2), we know that $\boldsymbol{x}_i \not\perp\!\!\!\perp_{p^{(1)}} \boldsymbol{x}_j \mid \boldsymbol{x}_{V \setminus L_{ij}^{\mathcal{G}}}$. However by Lemma 4.5, we have that $i \perp\!\!\!\perp_{\mathcal{G}} j \mid V \setminus L_{ij}^{\mathcal{G}}$, which implies, again by Proposition 4.1, that $p^{(1)} \notin \mathcal{M}(\mathcal{G})$. Thus, again by the same argument as Case 3, $\mathcal{S}_{\mathcal{I}}(\mathcal{G}^*) - \mathcal{S}_{\mathcal{I}}(\mathcal{G}) > 0$.

So far, all cases did not require interventional nodes ζ_k. Cases 5 and 6 treat the difference in immoralities implying interventional nodes ζ_k. Note that the arguments are analog to cases 3 and 4.

Case 5: Suppose that there is an immorality $i \to \ell \leftarrow \zeta_j$ in $\mathcal{G}^{*\mathcal{I}}$ which does not appear in $\mathcal{G}^{\mathcal{I}}$. The path (i, ℓ, ζ_j) is not d-blocked by $\zeta_{-j} \cup V \setminus L_{i\zeta_j}^{\mathcal{G}^{\mathcal{I}}}$ in $\mathcal{G}^{*\mathcal{I}}$ since $\ell \in \zeta_{-j} \cup V \setminus L_{i\zeta_j}^{\mathcal{G}^{\mathcal{I}}}$ (by same argument as presented in Case 3). By $\mathcal{I}$-faithfulness (Assumption 4.2), this means that

$$p^{(1)}(\boldsymbol{x}_i \mid \boldsymbol{x}_{V \setminus L_{i\zeta_j}^{\mathcal{G}^{\mathcal{I}}}}) \neq p^{(j)}(\boldsymbol{x}_i \mid \boldsymbol{x}_{V \setminus L_{i\zeta_j}^{\mathcal{G}^{\mathcal{I}}}}) \,. \tag{4.56}$$

Thus, $(p^{(1)}, p^{(j)}) \notin \mathcal{Z}(i, V \setminus L_{i\zeta_j}^{\mathcal{G}^{\mathcal{I}}})$ (defined in Equation (4.35)).

On the other hand, Lemma 4.5 implies that $i \perp\!\!\!\perp_{\mathcal{G}^{\mathcal{I}}} \zeta_j \mid \zeta_{-j} \cup V \setminus L_{i\zeta_j}^{\mathcal{G}^{\mathcal{I}}}$. Thus by Proposition 4.2 and since $\mathcal{F}_{\mathcal{I}}(\mathcal{G}) \subseteq \mathcal{M}_{\mathcal{I}}(\mathcal{G})$, we have that for all $\boldsymbol{\phi}$,

$$f_{\mathcal{G}\mathcal{I}\boldsymbol{\phi}}^{(1)}(\boldsymbol{x}_i \mid \boldsymbol{x}_{V \setminus L_{i\zeta_j}^{\mathcal{G}^{\mathcal{I}}}}) = f_{\mathcal{G}\mathcal{I}\boldsymbol{\phi}}^{(j)}(\boldsymbol{x}_i \mid \boldsymbol{x}_{V \setminus L_{i\zeta_j}^{\mathcal{G}^{\mathcal{I}}}}) \text{ i.e. } (f_{\mathcal{G}\mathcal{I}\boldsymbol{\phi}}^{(1)}, f_{\mathcal{G}\mathcal{I}\boldsymbol{\phi}}^{(j)}) \in \mathcal{Z}(i, V \setminus L_{i\zeta_j}^{\mathcal{G}^{\mathcal{I}}}) \,. \tag{4.57}$$

This means that $\mathcal{S}_{\mathcal{I}}(\mathcal{G}^*) > \mathcal{S}_{\mathcal{I}}(\mathcal{G})$ since

$$\mathcal{S}_{\mathcal{I}}(\mathcal{G}^*) - \mathcal{S}_{\mathcal{I}}(\mathcal{G}) = \inf_{\boldsymbol{\phi}} \sum_{k\in[K]} D_{KL}(p^{(k)}||f^{(k)}_{\mathcal{GI}\boldsymbol{\phi}}) \tag{4.58}$$

$$\geq \inf_{\boldsymbol{\phi}} D_{KL}(p^{(1)}||f^{(1)}_{\mathcal{GI}\boldsymbol{\phi}}) + D_{KL}(p^{(j)}||f^{(j)}_{\mathcal{GI}\boldsymbol{\phi}}) \tag{4.59}$$

$$\geq \inf_{(f^{(1)},f^{(j)})\in\mathcal{Z}(i,V\setminus L^{\mathcal{G}^{\mathcal{I}}}_{i\zeta_j})} D_{KL}(p^{(1)}||f^{(1)}) + D_{KL}(p^{(j)}||f^{(j)}) \tag{4.60}$$

$$> 0\,. \tag{4.61}$$

In (4.60), we use the fact that, for all $\boldsymbol{\phi}$, $(f^{(1)}_{\mathcal{GI}\boldsymbol{\phi}}, f^{(j)}_{\mathcal{GI}\boldsymbol{\phi}}) \in \mathcal{Z}(i, V \setminus L^{\mathcal{G}^{\mathcal{I}}}_{i\zeta_j})$. The very last strict inequality holds by Lemma 4.4, which applies here because $(p^{(1)}, p^{(j)}) \notin \mathcal{Z}(i, V \setminus L^{\mathcal{G}^{\mathcal{I}}}_{i\zeta_j})$.

Case 6: Suppose that there is an immorality $i \to \ell \leftarrow \zeta_j$ in $\mathcal{G}^{\mathcal{I}}$ which does not appear in $\mathcal{G}^{*\mathcal{I}}$. The path (i, ℓ, ζ_j) is not d-blocked by $\zeta_{-j} \cup V \setminus L^{\mathcal{G}^{\mathcal{I}}}_{i\zeta_j}$ in $\mathcal{G}^{*\mathcal{I}}$, since $\ell \notin \zeta_{-j} \cup V \setminus L^{\mathcal{G}^{\mathcal{I}}}_{i\zeta_j}$ and both $\mathcal{I}$-DAGs share the same skeleton. It follows by $\mathcal{I}$-faithfulness (Assumption 4.2) that

$$p^{(1)}(\boldsymbol{x}_i \mid \boldsymbol{x}_{V\setminus L^{\mathcal{G}^{\mathcal{I}}}_{i\zeta_j}}) \neq p^{(j)}(\boldsymbol{x}_i \mid \boldsymbol{x}_{V\setminus L^{\mathcal{G}^{\mathcal{I}}}_{i\zeta_j}})\,. \tag{4.62}$$

On the other hand, Lemma 4.5 implies that $i \perp\!\!\!\perp_{\mathcal{G}^{\mathcal{I}}} \zeta_j \mid \zeta_{-j} \cup V \setminus L^{\mathcal{G}^{\mathcal{I}}}_{i\zeta_j}$. Again by the $\mathcal{I}$-Markov property (Proposition 4.2), it means that, for all $\boldsymbol{\phi}$,

$$f^{(1)}_{\mathcal{GI}\boldsymbol{\phi}}(\boldsymbol{x}_i \mid \boldsymbol{x}_{V\setminus L^{\mathcal{G}^{\mathcal{I}}}_{i\zeta_j}}) = f^{(j)}_{\mathcal{GI}\boldsymbol{\phi}}(\boldsymbol{x}_i \mid \boldsymbol{x}_{V\setminus L^{\mathcal{G}^{\mathcal{I}}}_{i\zeta_j}})\,. \tag{4.63}$$

By an argument identical to that of Case 5, it follows that $\mathcal{S}_{\mathcal{I}}(\mathcal{G}^*) > \mathcal{S}_{\mathcal{I}}(\mathcal{G})$.

The proof is complete since there is no other way in which $\mathcal{G}^{\mathcal{I}}$ and $\mathcal{G}^{*\mathcal{I}}$ can differ in terms of skeleton and immoralities. ■

A.3. Theory for unknown targets

Theorem 4.1 assumes implicitly that, for each intervention k, the ground truth interventional target I^*_k is known. What if we do not have access to this information? We now present an extension of Theorem 4.1 to unknown targets. In this setting, the interventional family $\mathcal{I}$ is learned similarly to $\mathcal{G}$. We denote the ground truth interventional family by $\mathcal{I}^* := (I^*_1, \cdots, I^*_K)$ and assume that $I^*_1 := \emptyset$. We first recall score introduced in Section 4.3.3:

$$\mathcal{S}(\mathcal{G}, \mathcal{I}) := \sup_{\boldsymbol{\phi}} \sum_{k=1}^{K} \mathbb{E}_{\boldsymbol{x}\sim p^{(k)}} \log f^{(k)}(\boldsymbol{x}; \boldsymbol{M}^{\mathcal{G}}, \boldsymbol{R}^{\mathcal{I}}, \boldsymbol{\phi}) - \lambda|\mathcal{G}| - \lambda_{\boldsymbol{R}}|\mathcal{I}|\,, \tag{4.64}$$

where $f^{(k)}(\boldsymbol{x}; \boldsymbol{M}^{\mathcal{G}}, \boldsymbol{R}^{\mathcal{I}}, \boldsymbol{\phi})$ was defined in (4.15) and $|\mathcal{I}| = \sum_{k=1}^{K} |I_k|$. Notice that the assumption that $I^*_1 = \emptyset$ is integrated in the joint density of (4.15) with $k = 1$ (the row vector $\boldsymbol{R}^{\mathcal{I}}_{1:}$ has no effect).

The only difference between $\mathcal{S}_{\mathcal{I}^*}(\mathcal{G})$ and $\mathcal{S}(\mathcal{G},\mathcal{I})$ is that, in the latter, $\mathcal{I}$ is considered a variable and the extra regularizing term $-\lambda_{\boldsymbol{R}}|\mathcal{I}|$.

The result of this section relies on the exact same assumptions as those of Theorem 4.1, namely Assumptions 4.1, 4.2, 4.3 & 4.4.

The next Lemma is an adaptation of Lemma 4.2 to this new setting.

Lemma 4.6 (Rewriting of score differences)**.** *Under Assumption 4.1 & 4.4, we have*

$$\mathcal{S}(\mathcal{G}^*,\mathcal{I}^*) - \mathcal{S}(\mathcal{G},\mathcal{I}) = \inf_{\boldsymbol{\phi}} \sum_{k\in[K]} D_{KL}(p^{(k)}||f^{(k)}_{\mathcal{GI}\boldsymbol{\phi}}) + \lambda(|\mathcal{G}| - |\mathcal{G}^*|) + \lambda_{\boldsymbol{R}}(|\mathcal{I}| - |\mathcal{I}^*|)\,. \tag{4.65}$$

Proof. We note that $|\mathcal{S}(\mathcal{G}^*,\mathcal{I}^*)| = |\mathcal{S}_{\mathcal{I}^*}(\mathcal{G}^*) - \lambda_{\boldsymbol{R}}|\mathcal{I}^*|| < \infty$, by Lemma 4.1. This implies that the difference $\mathcal{S}(\mathcal{G}^*,\mathcal{I}^*) - \mathcal{S}(\mathcal{G},\mathcal{I})$ is always well defined.

The rest of the proof is identical to Lemma 4.2.■

We are now ready to state and prove our identifiability result for unknown targets.

Theorem 4.2 (Unknown targets identification)**.** *Suppose $\mathcal{I}^*$ is such that $I^*_1 := \emptyset$. Let $\mathcal{G}^*$ be the ground truth DAG and $(\hat{\mathcal{G}},\hat{\mathcal{I}}) \in \arg\max_{\mathcal{G}\in DAG,\mathcal{I}} \mathcal{S}(\mathcal{G},\mathcal{I})$. Under the same assumptions as Theorem 4.1 and for $\lambda, \lambda_{\boldsymbol{R}} > 0$ small enough, $\hat{\mathcal{G}}$ is $\mathcal{I}^*$-Markov equivalent to $\mathcal{G}^*$ and $\hat{\mathcal{I}} = \mathcal{I}^*$.*

Proof: We simply add two cases at the beginning of the proof of Theorem 4.1 to handle cases where $\mathcal{I} \neq \mathcal{I}^*$ (we will denote them by Case 0.1 and Case 0.2). Similarly to Theorem 4.1, it is sufficient to prove that, whenever $\mathcal{G} \notin \mathcal{I}^*\text{-MEC}(\mathcal{G}^*)$ or $\mathcal{I} \neq \mathcal{I}^*$, we have that $\mathcal{S}(\mathcal{G}^*,\mathcal{I}^*) > \mathcal{S}(\mathcal{G},\mathcal{I})$. For convenience, let us define

$$\eta(\mathcal{G},\mathcal{I}) := \inf_{\boldsymbol{\phi}} \sum_{k\in[K]} D_{KL}(p^{(k)}||f^{(k)}_{\mathcal{GI}\boldsymbol{\phi}})\,. \tag{4.66}$$

Case 0.1: Let $\mathbb{I}$ be the set of all $\mathcal{I}$ such that there exists $k_0 \in [K]$ and $j \in [d]$ such that $j \in I^*_{k_0}$ but $j \notin I_{k_0}$. Let $\mathcal{I} \in \mathbb{I}$ and let $\mathcal{G}$ be an arbitrary DAG.

Since the edge $\zeta_{k_0} \to j$ is in $\mathcal{G}^{*\mathcal{I}^*}$, we have that ζ_{k_0} and j are never d-separated. By $\mathcal{I}^*$-faithfulness (Assumption 4.2), we have that

$$p^{(1)}(\boldsymbol{x}_j|\boldsymbol{x}_{\pi^{\mathcal{G}}_j}) \neq p^{(k_0)}(\boldsymbol{x}_j|\boldsymbol{x}_{\pi^{\mathcal{G}}_j})\,. \tag{4.67}$$

Note that this is true for any conditioning set. It means $(p^{(1)}, p^{(k_0)}) \notin \mathcal{Z}(j,\pi^{\mathcal{G}}_j)$ (defined in (4.35)).

Since $j \notin I_k$, we have by definition from (4.15) that, for all $\boldsymbol{\phi}$,

$$f^{(1)}_{\mathcal{GI}\boldsymbol{\phi}}(\boldsymbol{x}_j|\boldsymbol{x}_{\pi^{\mathcal{G}}_j}) = f^{(k_0)}_{\mathcal{GI}\boldsymbol{\phi}}(\boldsymbol{x}_j|\boldsymbol{x}_{\pi^{\mathcal{G}}_j}) \text{ i.e. } (f^{(1)}_{\mathcal{GI}\boldsymbol{\phi}}, f^{(k_0)}_{\mathcal{GI}\boldsymbol{\phi}}) \in \mathcal{Z}(j,\pi^{\mathcal{G}}_j). \tag{4.68}$$

This implies that

$$\eta(\mathcal{G},\mathcal{I}) \geq \inf_{\boldsymbol{\phi}} D_{KL}(p^{(1)}||f^{(1)}_{\mathcal{G}\mathcal{I}\boldsymbol{\phi}}) + D_{KL}(p^{(k_0)}||f^{(k_0)}_{\mathcal{G}\mathcal{I}\boldsymbol{\phi}}) \tag{4.69}$$

$$\geq \inf_{(f^{(1)},f^{(k_0)})\in\mathcal{Z}(j,\pi_j^{\mathcal{G}})} D_{KL}(p^{(1)}||f^{(1)}) + D_{KL}(p^{(k_0)}||f^{(k_0)}) \tag{4.70}$$

$$> 0\,, \tag{4.71}$$

where (4.70) holds because, for all $\boldsymbol{\phi}$, $(f^{(1)}_{\mathcal{G}\mathcal{I}\boldsymbol{\phi}}, f^{(k_0)}_{\mathcal{G}\mathcal{I}\boldsymbol{\phi}}) \in \mathcal{Z}(j,\pi_j^{\mathcal{G}})$ and (4.71) holds by Lemma 4.4.

If $\min\{|\mathcal{G}| - |\mathcal{G}^*|, |\mathcal{I}| - |\mathcal{I}^*|\} \geq 0$, then clearly $\mathcal{S}(\mathcal{G}^*,\mathcal{I}^*) - \mathcal{S}(\mathcal{G},\mathcal{I}) > 0$. Let $\mathbb{S} := \{(\mathcal{G},\mathcal{I}) \in \text{DAG} \times \mathbb{I} \mid \min\{|\mathcal{G}| - |\mathcal{G}^*|, |\mathcal{I}| - |\mathcal{I}^*|\} < 0\}$. To make sure we have $\mathcal{S}(\mathcal{G}^*,\mathcal{I}^*) - \mathcal{S}(\mathcal{G},\mathcal{I}) > 0$ for all $(\mathcal{G},\mathcal{I}) \in \mathbb{S}$, we need to pick λ and $\lambda_{\boldsymbol{R}}$ sufficiently small. Choosing $\lambda + \lambda_{\boldsymbol{R}} < \min_{(\mathcal{G},\mathcal{I})\in\mathbb{S}} \frac{\eta(\mathcal{G},\mathcal{I})}{-\min\{|\mathcal{G}|-|\mathcal{G}^*|,|\mathcal{I}|-|\mathcal{I}^*|\}}$ is sufficient since (and note that the minimum exists because the set $\mathbb{S}$ is finite, and is strictly positive by (4.71)):

$$\lambda + \lambda_{\boldsymbol{R}} < \min_{(\mathcal{G},\mathcal{I})\in\mathbb{S}} \frac{\eta(\mathcal{G},\mathcal{I})}{-\min\{|\mathcal{G}| - |\mathcal{G}^*|, |\mathcal{I}| - |\mathcal{I}^*|\}} \tag{4.72}$$

$$\iff \lambda + \lambda_{\boldsymbol{R}} < \frac{\eta(\mathcal{G},\mathcal{I})}{-\min\{|\mathcal{G}| - |\mathcal{G}^*|, |\mathcal{I}| - |\mathcal{I}^*|\}} \;\; \forall (\mathcal{G},\mathcal{I}) \in \mathbb{S} \tag{4.73}$$

$$\iff -(\lambda + \lambda_{\boldsymbol{R}}) \min\{|\mathcal{G}| - |\mathcal{G}^*|, |\mathcal{I}| - |\mathcal{I}^*|\} < \eta(\mathcal{G},\mathcal{I}) \;\; \forall (\mathcal{G},\mathcal{I}) \in \mathbb{S} \tag{4.74}$$

$$\iff 0 < \eta(\mathcal{G},\mathcal{I}) + (\lambda + \lambda_{\boldsymbol{R}}) \min\{|\mathcal{G}| - |\mathcal{G}^*|, |\mathcal{I}| - |\mathcal{I}^*|\} \;\; \forall (\mathcal{G},\mathcal{I}) \in \mathbb{S} \tag{4.75}$$

$$\leq \eta(\mathcal{G},\mathcal{I}) + \lambda(|\mathcal{G}| - |\mathcal{G}^*|) + \lambda_{\boldsymbol{R}}(|\mathcal{I}| - |\mathcal{I}^*|) \tag{4.76}$$

$$= \mathcal{S}(\mathcal{G}^*,\mathcal{I}^*) - \mathcal{S}(\mathcal{G},\mathcal{I})\,. \tag{4.77}$$

From now on, we can assume $I_k^* \subseteq I_k$ for all $k \in [K]$, since otherwise we are in Case 0.1.

Case 0.2: Let $\bar{\mathbb{I}} := \{\mathcal{I} \mid [I_k^* \subseteq I_k \; \forall k]$ and $[\exists\, k_0, j$ s.t. $j \in I_{k_0}$ and $j \notin I_{k_0}^*]\}$. Let $\mathcal{I} \in \bar{\mathbb{I}}$ and let $\mathcal{G}$ be a DAG. We can already notice that $|\mathcal{I}| > |\mathcal{I}^*|$.

If $|\mathcal{G}| \geq |\mathcal{G}^*|$, then $\mathcal{S}(\mathcal{G}^*,\mathcal{I}^*) - \mathcal{S}(\mathcal{G},\mathcal{I}) > 0$ by (4.65). Let $\bar{\mathbb{S}} := \{(\mathcal{G},\mathcal{I}) \in \text{DAG} \times \bar{\mathbb{I}} \mid |\mathcal{G}| < |\mathcal{G}^*|\}$. To make sure $\mathcal{S}(\mathcal{G}^*,\mathcal{I}^*) - \mathcal{S}(\mathcal{G},\mathcal{I}) > 0$ for all $(\mathcal{G},\mathcal{I}) \in \bar{\mathbb{S}}$, we need to pick λ sufficiently small. Choosing $\lambda < \min_{(\mathcal{G},\mathcal{I})\in\bar{\mathbb{S}}} \frac{\eta(\mathcal{G},\mathcal{I}) + \lambda_{\boldsymbol{R}}(|\mathcal{I}| - |\mathcal{I}^*|)}{|\mathcal{G}^*| - |\mathcal{G}|}$ is sufficient since this implies

$$\lambda < \frac{\eta(\mathcal{G},\mathcal{I}) + \lambda_{\boldsymbol{R}}(|\mathcal{I}| - |\mathcal{I}^*|)}{|\mathcal{G}^*| - |\mathcal{G}|} \;\; \forall\, (\mathcal{G},\mathcal{I}) \in \bar{\mathbb{S}} \tag{4.78}$$

$$\iff \lambda(|\mathcal{G}^*| - |\mathcal{G}|) < \eta(\mathcal{G},\mathcal{I}) + \lambda_{\boldsymbol{R}}(|\mathcal{I}| - |\mathcal{I}^*|) \;\; \forall\, (\mathcal{G},\mathcal{I}) \in \bar{\mathbb{S}} \tag{4.79}$$

$$\iff 0 < \eta(\mathcal{G},\mathcal{I}) + \lambda(|\mathcal{G}| - |\mathcal{G}^*|) + \lambda_{\boldsymbol{R}}(|\mathcal{I}| - |\mathcal{I}^*|) \;\; \forall\, (\mathcal{G},\mathcal{I}) \in \bar{\mathbb{S}} \tag{4.80}$$

$$= \mathcal{S}(\mathcal{G}^*,\mathcal{I}^*) - \mathcal{S}(\mathcal{G},\mathcal{I})\,. \tag{4.81}$$

Cases 0.1 & 0.2 cover all situations where $\mathcal{I} \neq \mathcal{I}^*$. This implies that $\hat{\mathcal{I}} = \mathcal{I}^*$. For the rest of the proof, we can assume that $\mathcal{I} = \mathcal{I}^*$. By noting that $\mathcal{S}(\mathcal{G}^*,\mathcal{I}^*) - \mathcal{S}(\mathcal{G},\mathcal{I}) = \mathcal{S}_{\mathcal{I}^*}(\mathcal{G}^*) - \mathcal{S}_{\mathcal{I}^*}(\mathcal{G})$, we can apply exactly the same steps as in Theorem 4.1 to show that $\hat{\mathcal{G}} \in \mathcal{I}^*\text{-MEC}(\mathcal{G}^*)$.

We will end up with multiple conditions on λ and $\lambda_{\boldsymbol{R}}$. We now make sure they can all be satisfied simultaneously. Recall the three conditions we derived:

$$\lambda + \lambda_{\boldsymbol{R}} < \min_{(\mathcal{G},\mathcal{I})\in\mathbb{S}} \frac{\eta(\mathcal{G},\mathcal{I})}{-\min\{|\mathcal{G}| - |\mathcal{G}^*|, |\mathcal{I}| - |\mathcal{I}^*|\}} =: \alpha \tag{4.82}$$

$$\lambda < \min_{(\mathcal{G},\mathcal{I})\in\bar{\mathbb{S}}} \frac{\eta(\mathcal{G},\mathcal{I}) + \lambda_{\boldsymbol{R}}(|\mathcal{I}| - |\mathcal{I}^*|)}{|\mathcal{G}^*| - |\mathcal{G}|} =: \beta(\lambda_{\boldsymbol{R}}) \tag{4.83}$$

$$\lambda < \min_{\mathcal{G}\in\mathbb{G}^+} \frac{\eta(\mathcal{G},\mathcal{I}^*)}{|\mathcal{G}^*| - |\mathcal{G}|} =: \gamma\,, \tag{4.84}$$

where the third condition comes from the steps of Theorem 4.1. We can simply pick $\lambda_{\boldsymbol{R}} \in (0, \alpha)$ and $\lambda \in (0, \min\{\alpha - \lambda_{\boldsymbol{R}}, \beta(\lambda_{\boldsymbol{R}}), \gamma\})$. ■

A.4. Adapting the score to perfect interventions

The score developed in Section 4.3.1 is designed for general imperfect interventions. Since perfect interventions are just a special case of imperfect ones, this score will work on perfect interventions without problems. However, one can leverage the fact that the interventions are perfect to simplify the score a little bit.

$$\max_{\mathcal{G}\in\text{DAG}} \mathcal{S}_{\mathcal{I}^*}(\mathcal{G}) \tag{4.85}$$

$$= \max_{\mathcal{G}\in\text{DAG}} \sup_{\boldsymbol{\phi}} \sum_{k=1}^{K} \mathbb{E}_{\boldsymbol{x}\sim p^{(k)}} \log f^{(k)}(\boldsymbol{x}; \boldsymbol{M}^{\mathcal{G}}, \boldsymbol{R}^{\mathcal{I}^*}, \boldsymbol{\phi}) - \lambda|\mathcal{G}| \tag{4.86}$$

$$= \max_{\mathcal{G}\in\text{DAG}} \sup_{\boldsymbol{\phi}^{(1)}} \left[\sum_{k=1}^{K} \mathbb{E}_{\boldsymbol{x}\sim p^{(k)}} \log \prod_{j\notin I_k^*} \tilde{f}(\boldsymbol{x}_j; \text{NN}(\boldsymbol{M}_j^{\mathcal{G}} \odot \boldsymbol{x}; \boldsymbol{\phi}_j^{(1)})) \right]$$
$$+ \sup_{\boldsymbol{\phi}^{(2)},\dots,\boldsymbol{\phi}^{(K)}} \left[\sum_{k=2}^{K} \mathbb{E}_{\boldsymbol{x}\sim p^{(k)}} \log \prod_{j\in I_k^*} \tilde{f}(\boldsymbol{x}_j; \text{NN}(\boldsymbol{M}_j^{\mathcal{G}} \odot \boldsymbol{x}; \boldsymbol{\phi}_j^{(k)})) \right] - \lambda|\mathcal{G}| \tag{4.87}$$

$$= \max_{\mathcal{G}\in\text{DAG}} \sup_{\boldsymbol{\phi}^{(1)}} \left[\sum_{k=1}^{K} \mathbb{E}_{\boldsymbol{x}\sim p^{(k)}} \log \prod_{j\notin I_k^*} \tilde{f}(\boldsymbol{x}_j; \text{NN}(\boldsymbol{M}_j^{\mathcal{G}} \odot \boldsymbol{x}; \boldsymbol{\phi}_j^{(1)})) \right]$$
$$+ \sup_{\boldsymbol{\phi}^{(2)},\dots,\boldsymbol{\phi}^{(K)}} \left[\sum_{k=2}^{K} \mathbb{E}_{\boldsymbol{x}\sim p^{(k)}} \log \prod_{j\in I_k^*} \tilde{f}(\boldsymbol{x}_j; \text{NN}(0 \odot \boldsymbol{x}; \boldsymbol{\phi}_j^{(k)})) \right] - \lambda|\mathcal{G}|\,, \tag{4.88}$$

where in (4.88) we use the fact that the interventions are perfect. In (4.88), the second sup does not depend on $\mathcal{G}$, so it can be ignored without changing the $\arg\max_{\mathcal{G}\in\text{DAG}}$.

Hence, for perfect intervention we use the score

$$\mathcal{S}^{\text{perf}}_{\mathcal{I}^*}(\mathcal{G}) := \sup_{\boldsymbol{\phi}^{(1)}} \left[\sum_{k=1}^{K} \mathbb{E}_{\boldsymbol{x}\sim p^{(k)}} \log \prod_{j \notin I_k^*} \tilde{f}(\boldsymbol{x}_j; \text{NN}(\boldsymbol{M}_j^{\mathcal{G}} \odot \boldsymbol{x}; \boldsymbol{\phi}_j^{(1)})) \right] - \lambda |\mathcal{G}| \,. \tag{4.89}$$

B. Additional information

B.1. Synthetic data sets

In this section, we describe how the different synthetic data sets were generated. For each type of data set, we first sample a DAG following the *Erdős-Rényi* scheme and then we sample the parameters of the different causal mechanisms as stated below (in the bulleted list). For 10-node graphs, single node interventions are performed on every node. For 20-node graphs, interventions target 1 to 2 nodes chosen uniformly at random. Then, $n/(d+1)$ examples are sampled for each interventional setting (if n is not divisible by $d+1$, some intervention setting may have one extra sample in order to have a total of n samples). The data are then normalized: we subtract the mean and divide by the standard deviation. For all data sets, the source nodes are Gaussian with zero mean and variance sampled from $\mathcal{U}[1,2]$. The noise variables N_j are mutually independent and sampled from $\mathcal{N}(0,\sigma_j^2)\ \ \forall j$, where $\sigma_j^2 \sim \mathcal{U}[1,2]$.

For perfect intervention, the distribution of intervened nodes is replaced by a marginal $\mathcal{N}(2,1)$. This type of intervention, that produce a mean-shift, is similar to those used in Hauser and Bühlmann [2012], Squires et al. [2020]. For imperfect interventions, besides the initial parameters, an extra set of parameters were sampled by perturbing the initial parameters as described below. For nodes without parents, the distribution of intervened nodes is replaced by a marginal $\mathcal{N}(2,1)$. Both for the perfect and imperfect cases, we explore other types of interventions and report the results in Appendix C.5. We now describe the causal mechanisms and the nature of the imperfect intervention for the three different types of data set:

- The *linear* data sets are generated following $\boldsymbol{x}_j := \boldsymbol{w}_j^\top \boldsymbol{x}_{\pi_j^{\mathcal{G}}} + 0.4 \cdot n_j \ \ \forall j$, where $\boldsymbol{w}_j$ is a vector of $|\pi_j^{\mathcal{G}}|$ coefficients each sampled uniformly from $[-1,-0.25] \cup [0.25,1]$ (to make sure there are no $\boldsymbol{w}$ close to 0). Imperfect interventions are obtained by adding a random vector of $\mathcal{U}([-5,-2] \cup [2,5])$ to $\boldsymbol{w}_j$.
- The *additive noise model* (ANM) data sets are generated following $\boldsymbol{x}_j := f_j(\boldsymbol{x}_{\pi_j^{\mathcal{G}}}) + 0.4 \cdot n_j$ $\forall j$, where the functions f_j are fully connected neural networks with one hidden layer of 10 units and *leaky ReLU* with a negative slope of 0.25 as nonlinearities. The weights of each neural network are randomly initialized from $\mathcal{N}(0,1)$. Imperfect interventions are obtained by adding a random vector of $\mathcal{N}(0,1)$ to the last layer.
- The *nonlinear with non-additive noise* (NN) data sets are generated following $\boldsymbol{x}_j := f_j(\boldsymbol{x}_{\pi_j^{\mathcal{G}}}, n_j)$ $\forall j$, where the functions f_j are fully connected neural networks with one hidden

layer of 20 units and *tanh* as nonlinearities. The weights of each neural network are randomly initialized from $\mathcal{N}(0,1)$. Similarly to the additive noise model, imperfect intervention are obtained by adding a random vector of $\mathcal{N}(0,1)$ to the last layer.

B.2. Deep Sigmoidal Flow: Architectural details

A layer of a Deep Sigmoidal Flow is similar to a fully-connected network with one hidden layer, a single input, and a single output, but is defined slightly differently to ensure that the mapping is invertible and that the Jacobian is tractable. Each layer l is defined as follows:

$$h^{(l)}(\boldsymbol{x}) = \sigma^{-1}(\boldsymbol{w}^\top \sigma(\boldsymbol{a} \cdot \boldsymbol{x} + b))\,, \tag{4.90}$$

where $0 < \boldsymbol{w}_i < 1$, $\sum_i \boldsymbol{w}_i = 1$ and $\boldsymbol{a}_i > 0$. In our method, the neural networks $\text{NN}(\cdot; \boldsymbol{\phi}_j^{(k)})$ output the parameters $(\boldsymbol{w}_j, \boldsymbol{a}_j, b_j)$ for each DSF τ_j. To ensure that the determinant of the Jacobian is calculated in a numerically-stable way, we follow the recommendations of Huang et al. [2018b]. While other flows like the Deep Dense Sigmoidal Flow have more capacity, DSF was sufficient for our use.

B.3. Optimization

In this section, we show how the augmented Lagrangian is applied, how the gradient is estimated and, finally, we illustrate the learning dynamics by analyzing an example.

Let us recall the score and the optimization problem from Section 4.3.2:

$$\hat{\mathcal{S}}_{\text{int}}(\boldsymbol{\Lambda}) := \sup_{\boldsymbol{\phi}} \mathop{\mathbb{E}}_{\boldsymbol{M}\sim\sigma(\boldsymbol{\Lambda})} \left[\sum_{k=1}^{K} \mathop{\mathbb{E}}_{\boldsymbol{x}\sim p^{(k)}} \log f^{(k)}(\boldsymbol{x}; \boldsymbol{M}, \boldsymbol{\phi}) - \lambda ||\boldsymbol{M}||_0 \right], \tag{4.91}$$

$$\sup_{\boldsymbol{\Lambda}} \hat{\mathcal{S}}_{\text{int}}(\boldsymbol{\Lambda}) \quad \text{s.t.} \quad \operatorname{Tr} e^{\sigma(\boldsymbol{\Lambda})} - d = 0\,. \tag{4.92}$$

We optimize for $\boldsymbol{\phi}$ and $\boldsymbol{\Lambda}$ jointly, which yields the following optimization problem:

$$\sup_{\boldsymbol{\phi},\boldsymbol{\Lambda}} \mathop{\mathbb{E}}_{\boldsymbol{M}\sim\sigma(\boldsymbol{\Lambda})} \left[\sum_{k=1}^{K} \mathop{\mathbb{E}}_{\boldsymbol{x}\sim p^{(k)}} \log f^{(k)}(\boldsymbol{x}; \boldsymbol{M}, \boldsymbol{\phi}) \right] - \lambda ||\sigma(\boldsymbol{\Lambda})||_1 \quad \text{s.t.} \quad \operatorname{Tr} e^{\sigma(\boldsymbol{\Lambda})} - d = 0\,, \tag{4.93}$$

where we used the fact that $\mathbb{E}_{\boldsymbol{M}\sim\sigma(\boldsymbol{\Lambda})} ||\boldsymbol{M}||_0 = ||\sigma(\boldsymbol{\Lambda})||_1$. Let us use the notation:

$$h(\boldsymbol{\Lambda}) := \operatorname{Tr} e^{\sigma(\boldsymbol{\Lambda})} - d. \tag{4.94}$$

The augmented Lagrangian transforms the constrained problem into a sequence of unconstrained problems of the form

$$\sup_{\boldsymbol{\phi},\boldsymbol{\Lambda}} \mathop{\mathbb{E}}_{\boldsymbol{M}\sim\sigma(\boldsymbol{\Lambda})} \left[\sum_{k=1}^{K} \mathop{\mathbb{E}}_{\boldsymbol{x}\sim p^{(k)}} \log f^{(k)}(\boldsymbol{x}; \boldsymbol{M}, \boldsymbol{\phi}) \right] - \lambda ||\sigma(\boldsymbol{\Lambda})||_1 - \gamma_t h(\boldsymbol{\Lambda}) - \frac{\mu_t}{2} h(\boldsymbol{\Lambda})^2\,, \tag{4.95}$$

where γ_t and μ_t are the Lagrangian multiplier and the penalty coefficient of the tth unconstrained problem, respectively. In all our experiments, we initialize $\gamma_0 = 0$ and $\mu_0 = 10^{-8}$. Each such problem is approximately solved using a stochastic gradient descent algorithm (RMSprop Tieleman and Hinton [2012] in our experiments). We consider that a subproblem has converged when (4.95) evaluated on a held-out data set stops increasing. Let $(\boldsymbol{\phi}_t^*, \boldsymbol{\Lambda}_t^*)$ be the approximate solution to subproblem t. Then, γ_t and μ_t are updated according to the following rule:

$$\begin{aligned} \gamma_{t+1} &\leftarrow \gamma_t + \mu_t \cdot h\left(\boldsymbol{\Lambda}_t^*\right) \\ \mu_{t+1} &\leftarrow \begin{cases} \eta \cdot \mu_t, & \text{if } h\left(\boldsymbol{\Lambda}_t^*\right) > \delta \cdot h\left(\boldsymbol{\Lambda}_{t-1}^*\right) \\ \mu_t, & \text{otherwise} \end{cases} \end{aligned} \tag{4.96}$$

with $\eta = 2$ and $\delta = 0.9$. Each subproblem t is initialized using the previous subproblem's solution $(\boldsymbol{\phi}_{t-1}^*, \boldsymbol{\Lambda}_{t-1}^*)$. The augmented Lagrangian method stops when $h(\boldsymbol{\Lambda}) \leq 10^{-8}$ and the graph formed by adding an edge whenever $\sigma(\boldsymbol{\Lambda}) > 0.5$ is acyclic.

Gradient estimation. The gradient of (4.95) w.r.t. $\boldsymbol{\phi}$ and $\boldsymbol{\Lambda}$ is estimated by

$$\nabla_{\boldsymbol{\phi},\boldsymbol{\Lambda}} \left[\frac{1}{|B|} \sum_{i \in B} \log f^{(k_i)}(\boldsymbol{x}^{(i)}; \boldsymbol{M}^{(i)}, \boldsymbol{\phi}) - \lambda_t h(\boldsymbol{\Lambda}) - \frac{\mu_t}{2} h(\boldsymbol{\Lambda})^2 \right], \tag{4.97}$$

where B is an index set sampled without replacement, $x^{(i)}$ is an example from the training set and k_i is the index of its corresponding intervention. To compute the gradient of the likelihood part w.r.t. $\boldsymbol{\Lambda}$, we use the Straight-Through Gumbel-Softmax estimator, adapted to sigmoids Maddison et al. [2017], Jang et al. [2017]. This approach was already used in the context of causal discovery without interventional data Ng et al. [2019], Kalainathan et al. [2018]. The matrix $\boldsymbol{M}^{(i)}$ is given by

$$\boldsymbol{M}^{(i)} := \mathbb{I}(\sigma(\boldsymbol{\Lambda} + \boldsymbol{L}^{(i)}) > 0.5) + \sigma(\boldsymbol{\Lambda} + \boldsymbol{L}^{(i)}) - \text{grad-block}(\sigma(\boldsymbol{\Lambda} + \boldsymbol{L}^{(i)})), \tag{4.98}$$

where $\boldsymbol{L}^{(i)}$ is a $d \times d$ matrix filled with independent Logistic samples, $\mathbb{I}$ is the indicator function applied element-wise and the function *grad-block* is such that grad-block$(z) = z$ and ∇_zgrad-block$(z) = 0$. This implies that each entry of $\boldsymbol{M}^{(i)}$ evaluates to a discrete Bernoulli sample with probability given by $\sigma(\boldsymbol{\Lambda})$ while the gradient w.r.t. $\boldsymbol{\Lambda}$ is computed using the soft Gumbel-Softmax sample. This yields a biased estimation of the actual gradient of objective (4.95), but its variance is low compared to the popular unbiased REINFORCE estimator (a Monte Carlo estimator relying on the log-trick) Rezende et al. [2014], Maddison et al. [2017]. A temperature term can be added inside the sigmoid, but we found that a temperature of one gave good results.

In addition to this, we experimented with a different relaxation for the discrete variable $\boldsymbol{M}$. We tried treating $\boldsymbol{M}$ directly as a learnable parameter constrained in $[0, 1]$ via gradient projection. However, this approach yielded significantly worse results. We believe that the fact $\boldsymbol{M}$ is continuous in this setting is problematic, since as an entry of $\boldsymbol{M}$ gets closer and closer to zero, the weights of the first neural network layer can compensate, without affecting the likelihood whatsoever. This

cannot happen when using the Straight-Through Gumbel-Softmax estimator because the neural network weights are only exposed to discrete M.

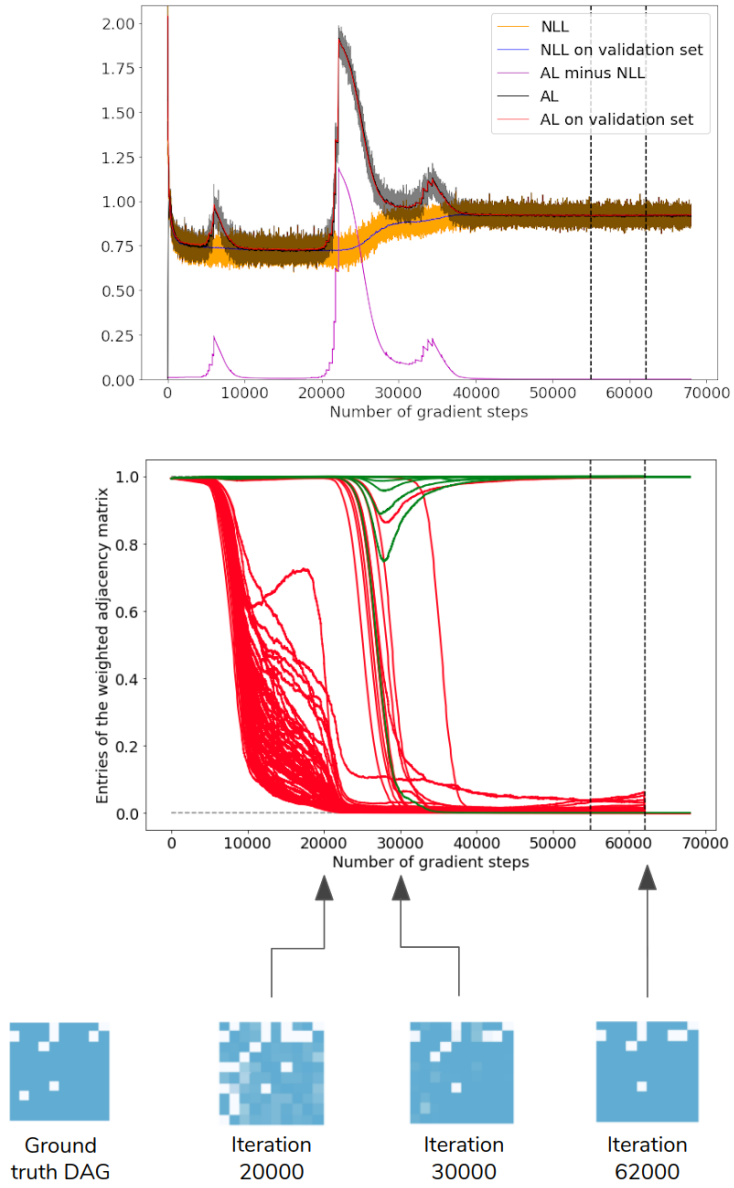


Figure 4.6. Top: Learning curves during training. *NLL* and *NLL on validation* are respectively the (pseudo) negative log-likelihood (NLL) on training and validation sets. *AL minus NLL* can be thought of as the acyclicity constraint violation plus the edge sparsity regularizer. *AL* and *AL on validation set* are the augmented Lagrangian objectives on training and validation set, respectively. **Middle and bottom:** Entries of the matrix $\sigma(\mathbf{\Lambda})$ w.r.t. to the number of iterations (green edges = edge present in the ground truth DAG, red edges = edge not present). The adjacency matrix to the left correspond to the ground truth DAG. The other matrices correspond to $\sigma(\mathbf{\Lambda})$ at $20\,000$, $30\,000$ and $62\,000$ iterations.

Learning dynamics. We present in Figure 4.6 the learning curves (top) and the matrix $\sigma(\mathbf{\Lambda})$ (middle and bottom) as DCDI-DSF is trained on a linear data set with perfect intervention sampled from a sparse 10-node graph (the same phenomenon was observed in a wide range of settings). In the graph at the top, we show the augmented Lagrangian and the (pseudo) negative log-likelihood (NLL) on train and validation set. To be exact, the NLL corresponds to a negative log-likelihood only once acyclicity is achieved. In the graph representing $\sigma(\mathbf{\Lambda})$ (middle), each curve represents a $\sigma(\alpha_{ij})$: green edges are edges present in the ground truth DAG and red edges are edges not present. The same information is presented in matrix form for a few specific iterations and can be easily compared to the adjacency matrix of the ground truth DAG (white = presence of an edge, blue = absence). Recall that when a $\sigma(\alpha_{ij})$ is equal (or close to) 0, it means that the entry ij of the mask M will also be 0. This is equivalent to say that the edge is not present in the learned DAG.

In this section, we review some important steps of the learning dynamics. At first, the NLL on the training and validation sets decrease sharply as the model fits the data. Around iteration 5000, the decrease slows down and the weights of the constraint (namely γ and μ) are increased. This puts pressure on the entries $\sigma(\alpha_{ij})$ to decrease. At iteration $20\,000$, many $\sigma(\alpha_{ij})$ that correspond to red edges have diminished close to 0, meaning that edges are correctly removed. It is noteworthy to mention that the matrix at this stage is close to being symmetric: the algorithm did not yet choose an orientation for the different edges. While this learned graph still has false-positive edges, the skeleton is reminiscent of a Markov Equivalence Class. As the training progresses, the weights of the constraint are greatly increased passed the $20\,000$th iteration leading to the removal of additional edges (leading also to an NLL increase). Around iteration $62\,000$ (the second vertical line), the stopping criterion is met: the acyclicity constraint is below the threshold (i.e. $h(\mathbf{\Lambda}) \leq 10^{-8}$), the learned DAG is acyclic and the augmented Lagrangian on the validation set is not improving anymore. Edges with a $\sigma(\alpha_{ij})$ higher than 0.5 are set to 1 and others set to 0. The learned DAG has a SHD of 1 since it has a reversed edge compared to the ground truth DAG.

Finally, we illustrate the learning of interventional targets in the (perfect) unknown intervention setting by comparing an example of $\sigma(\beta_{kj})$, the learned targets, with the ground truth targets in Figure 4.7. Results are from DCDI-G on 10-node graph with higher connectivity. Each column corresponds to an interventional target I_k and each row corresponds to a node. In the right matrix, a dark grey square in position ij means that the node i was intervened on in the interventional setting I_j. Each entry of the left matrix corresponds to the value of $\sigma(\beta_{kj})$. The binary matrix $\boldsymbol{R}$ (from Equation 4.15) is sampled following these entries.

B.4. Baseline methods

In this section, we provide additional details on the baseline methods and cite the implementations that were used. GIES has been designed for the perfect interventions setting. It assumes linear

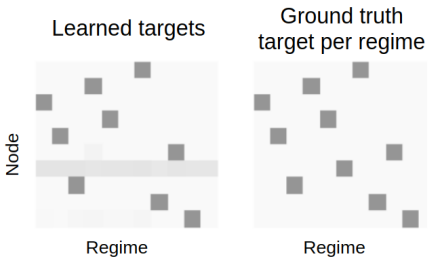


Figure 4.7. Learned targets $\sigma(\beta_{kj})$ compared to the ground truth targets.

relations with Gaussian noise and outputs an $\mathcal{I}$-Markov equivalence classes. In order to obtain the SHD and SID, we compare a DAG randomly sampled from the returned $\mathcal{I}$-Markov equivalence classes to the ground truth DAG. CAM has been modified to support perfect interventions. In particular, we used the loss that was already present in the code (similarly to the loss proposed for DCDI in the perfect intervention setting). Also, the preliminary neighbor search (PNS) and pruning processes were modified to not take into account data where variables are intervened on. Note that, while these two methods yield competitive results in the imperfect intervention setting, they were designed for perfect interventions: the targeted conditional are not fitted by an additional model (in contrast to our proposed score), they are simply removed from the score. Finally, JCI-PC is JCI used with the PC method Mooij et al. [2020]. The graph to learn is augmented with context variables (one per system variable in our case). This modified version of PC can deal with unknown interventions. For the conditional independence test, we only used the gaussian CI test since using KCI-test was too slow for this algorithm.

For GIES, we used the implementation from the R package `pcalg`. For CAM, we modified the implementation from the R package `pcalg`. For IGSP and UT-IGSP, we used the implementation from `https://github.com/uhlerlab/causaldag`. The cutoff values used for `alpha-inv` was always the same as `alpha`. For JCI-PC, we modified the implementation from the R package `pcalg` using code from the JCI repository: `https://github.com/caus-am/jci/tree/master/jci`. The normalizing flows that we used for DCDI-DSF were adapted from the DSF implementation provided by its author Huang et al. [2018b]. We also used several tools from the Causal Discovery Toolbox (`https://github.com/FenTechSolutions/CausalDiscoveryToolbox`) Kalainathan and Goudet [2019] to interface R with Python and to compute the SHD and SID metrics.

B.5. Default hyperparameters and hyperparameter search

For all score-based methods, we performed a hyperparameter search. The models were trained on 80% examples and evaluated on the 20% remaining examples. The hyperparameter combination chosen was the one that induced the lowest negative log-likelihood on the held-out examples. For

DCDI, a grid search was performed over 10 values of the regularization coefficient (see Table 4.1) for known interventions (10 hyperparameter combinations in total) and, in the unknown intervention case, 3 values for the regularization coefficient of the learned targets $\lambda_{\boldsymbol{R}}$ were also explored (30 hyperparameter combinations in total). For GIES and CAM, 50 hyperparameter combinations were considered using a random search following the sampling scheme of Table 4.1.

For IGSP, UT-IGSP and JCI-PC, we could not do a similar hyperparameter search since there is no score available to rank hyperparameter combinations. Thus, all examples were used to fit the model. Despite this, for IGSP and UT-IGSP, we explored a range of cutoff values around 10^{-5} (the value used for all the experiments in Squires et al. [2020]): $\alpha = \{2e-1, 1e-1, 1e-2, 1e-3, 1e-5, 1e-7, 1e-9\}$. In the main text and figures, we report results with $\alpha = 1e-3$, which yielded low SHD and SID. For JCI-PC, we tested the following range of cutoff values: $\alpha = \{2e-1, 1e-1, 1e-2, 1e-3\}$ and report results with $\alpha = 1e-3$. Note that in a realistic setting, we do not have access to the ground truth graphs to choose a good cutoff value.

	Hyperparameter space
DCDI	$\log_{10}(\lambda) \sim \mathcal{U}\{-7,-6,-5,-4,-3,-2,-1,0,1,2\}$ $\log_{10}(\lambda_{\boldsymbol{R}}) \sim \mathcal{U}\{-4,-3,-2\}$ (only for unknown interventions)
CAM	$\log_{10}$(pruning cutoff) $\sim \mathcal{U}[-7,0]$
GIES	$\log_{10}$(regularizer coefficient) $\sim \mathcal{U}[-4,4]$

Table 4.1. Hyperparameter search spaces for each algorithm

Except for the normalizing flows of DCDI-DSF, DCDI-G and DCDI-DSF used exactly the same default hyperparameters that are summarized in Table 4.2. Some of these hyperparameters (μ_0, γ_0), which are related to the optimization process are presented in Appendix B.3. These hyperparameters were used for almost all experiments, except for the real-world data set and the two-node graphs with complex densities, where overfitting was observed. Smaller architectures were tested until no major overfitting was observed. The default hyperparameters were chosen using small-scale experiments on perfect-known interventions data sets in order to have a small SHD. Since we observed that DCDI is not highly sensible to changes in hyperparameter values, only the regularization factors were part of a more thorough hyperparameter search. The neural networks were initialized following the Xavier initialization Glorot and Bengio [2010a]. The neural network activation functions were leaky-ReLU. RMSprop was used as the optimizer Tieleman and Hinton [2012] with minibatches of size 64.

DCDI hyperparameters
μ_0: 10^{-8}, γ_0: 0, η: 2, δ: 0.9
Augmented Lagrangian constraint threshold: 10^{-8}
learning rate: 10^{-3}
hidden units: 16
hidden layers: 2
flow hidden units: 16 (only for DCDI-DSF)
flow hidden layers: 2 (only for DCDI-DSF)

Table 4.2. Default Hyperparameter for DCDI-G and DCDI-DSF

C. Additional experiments

C.1. Real-world data set

We tested the methods that support perfect intervention on the flow cytometry data set of Sachs et al. [2005]. The measurements are the level of expression of phosphoproteins and phospholipids in human cells. Interventions were performed by using reagents to activate or inhibit the measured proteins. As in Wang et al. [2017], we use a subset of the data set, excluding experimental conditions where the perturbations were not directly done on a measured protein. This subset comprises 5 846 measurements: 1 755 measurements are considered observational, while the other 4 091 measurements are from five different single node interventions (with the following proteins as targets: Akt, PKC, PIP2, Mek, PIP3). The consensus graph from Sachs et al. [2005] that we use as the ground truth DAG contains 11 nodes and 17 edges. While the flow cytometry data set is standard in the causal structure learning literature, some concerns have been raised. The "consensus" network proposed by Sachs et al. [2005] has been challenged by some experts Mooij et al. [2016]. Also, several assumptions of the different models may not be respected in this real-world data set (for more details, see Mooij et al. [2016]): i) the causal sufficiency assumption may not hold, ii) the interventions may not be as specific as stated, and iii) the ground truth network is possibly not a DAG since feedback loops are common in cellular signaling networks.

Method	SHD	SID	tp	fn	fp	rev	F_1 score
IGSP	18	54	4	6	5	7	0.42
GIES	38	34	10	0	41	7	0.33
CAM	35	20	12	1	30	4	0.51
DCDI-G	36	43	6	2	25	9	0.31
DCDI-DSF	33	47	6	2	22	9	0.33

Table 4.3. Results for the flow cytometry data sets

In Table 4.3 we report SHD and SID for all methods, along with the number of true positive (tp), false-negative (fn), false positive (fp), reversed (rev) edges, and the F_1 score. There are no measures of central tendencies, since there is only one graph. The modified version of CAM has overall the best performance: the highest F_1 score and a low SID. IGSP has a low SHD, but a high SID, which can be explained by the relatively high number of false negative. DCDI-G and DCDI-DSF have SHDs comparable to GIES and CAM, but higher than IGSP. In terms of SID, they outperform IGSP, but not GIES and CAM. Finally, the DCDI models have F_1 scores similar to that of GIES. Hence, we conclude that DCDI performs comparably to the state of the art on this data set, while none of the methods show great performance across the board.

Hyperparameters. We report the hyperparameters used for Table 4.3. IGSP used the KCI-test with a cutoff value of 10^{-3}. Hyperparameters for CAM and GIES were chosen following the hyperparameter search described in Appendix B.5. For DCDI, since overfitting was observed, we included some hyperparameters related to the architecture in the hyperparameter grid search (number of hidden units: $\{4, 8\}$, number of hidden layers: $\{1, 2\}$ and only for DSF, number of flow hidden units: $\{4, 8\}$, number of flow layers: $\{1, 2\}$), and used the scheme described in Appendix B.5 for choosing the regularization coefficient.

C.2. Learning causal direction from complex distributions

To show that insufficient capacity can hinder learning the right causal direction, we used toy data sets with simple 2-node graphs under perfect and imperfect interventions. We show, in Figure 4.8 and 4.9, the joint densities respectively learned by DCDI-DSF and DCDI-G. We tested two different data sets: X and DNA, which corresponds to the left and right column, respectively. In both data sets, we experimented with perfect and imperfect interventions, on both the cause and the effect, i.e. $\mathcal{I} = (\emptyset, \{1\}, \{2\})$. In both figures, the top row corresponds to the learned densities when no intervention are performed. The bottom row corresponds to the learned densities under an imperfect intervention on the effect variable (changing the conditional).

For the X data set, both under perfect and imperfect interventions, the incapacity of DCDI-G to model this complex distribution properly makes it conclude (falsely) that there is no dependency between the two variables (the μ outputted by DCDI-G is constant). Conversely, for the DNA data set with perfect interventions, it does infer the dependencies between the two variables and learn the correct causal direction, although the distribution is modeled poorly. Notice that, for the DNA data set with imperfect interventions, the lack of capacity of DCDI-G has pushed it to learn the same density with and without interventions (compare the two densities in the second column of Figure 4.9; the learned density functions remain mostly unchanged from top to bottom). This prevented DCDI-G from learning the correct causal direction, while DCDI-DSF had no problem. We believe that if the imperfect interventions were more radical, DCDI-G could have recovered

Figure 4.8. Joint density learned by DCDI-DSF. White dots are data points and the color represents the learned density. The x-axis is cause and the y-axis is the effect. First row is observational while second row is with an imperfect intervention on the effect.

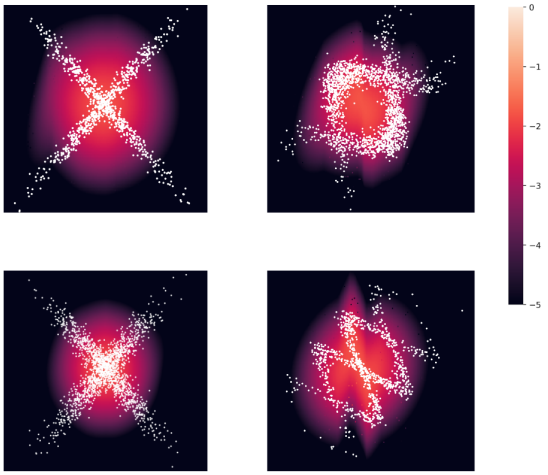


Figure 4.9. Joint density learned by DCDI-G. White dots are data points and the color represents the learned density. The x-axis is cause and the y-axis is the effect. First row is observational while second row is with an imperfect intervention on the effect.

the correct direction even though it lacks capacity. In all cases, DCDI-DSF can easily model these functions and systematically infers the right causal direction.

While the proposed data sets are synthetic, similar multimodal distributions could be observed in real-world data sets due to latent variables that are parent of only one node (i.e., that are not confounders). A hidden variable that act as a selector between two different mechanisms could induce distributions similar to those in Figures 4.8 and 4.9. In fact, this idea was used to produce the synthetic data sets, i.e., a latent variable $z \in \{0, 1\}$ was sampled and, according to its value, examples were generated following one of two mechanisms. The X dataset (second column in the

figures) was generated by two linear mechanisms in the following way:

$$y := \begin{cases} wx + N & z = 0 \\ -wx + N & z = 1, \end{cases}$$

where N is a Gaussian noise and w was randomly sampled from $[-1, -0.25] \cup [0.25, 1]$.

C.3. Scalability experiments

Figure 4.10 presents two experiments which study the scalability of various methods in terms of number of examples (left) and number of variables (right). In these experiments, the runtime was restricted to 12 hours while the RAM memory was restricted to 16GB. All experiments considered perfect interventions. Experiments from Figure 4.10 were run with fixed hyperparameters. **DCDI.** Same as Table 4.2 except $\mu_0 = 10^{-2}$, # hidden units $= 8$ and $\lambda = 10^{-1}$. **CAM.** Pruning cutoff $= 10^{-3}$. Preliminary neighborhood selection was performed in the large graph experiments (otherwise CAM cannot run on 50 nodes in less than 12 hours). **GIES.** Regularizing parameter $= 1$. **IGSP.** The suffixes -G and -K refers to the partial correlation test and the KCI-test, respectively. The α parameter is set to 10^{-3}.
Number of examples. DCDI was the only algorithm supporting nonlinear relationships that could run on as much as 1 million examples without running out of time or memory. We believe different trade-offs between SHD and SID could be achieved with different hyperparameters, especially for GIES and CAM which achieved very good SID but poor SHD.
Number of variables. We see that using a GPU starts to pay off for graphs of 50 nodes or more. For 10-50 nodes data sets, DCDI-GPU outperforms the other methods in terms of SHD and SID, while maintaining a runtime similar to CAM. For the hundred-node data sets, the runtime of DCDI increases significantly with a SHD/SID performance comparable to the much faster GIES. We believe the weaker performance of DCDI in the hundred-node setting is due to the fact that the conditionals are high dimensional functions which are prone to overfitting. Also, we believe this runtime could be significantly reduced by limiting the number of parents via preliminary neighborhood selection similar to CAM Bühlmann et al. [2014]. This would have the effect of reducing the cost of computing the gradient of w.r.t. to the neural network parameters. These adaptions to higher dimensionality are left as future work.

C.4. Ablation study

In this section, by doing ablation studies, we show that i) that interventions are beneficial to our method to recover the DAG, ii) that the proposed losses yield better results than a standard loss ignoring information about interventions, and iii) that the use of high capacity model is relevant for nonlinear data sets.

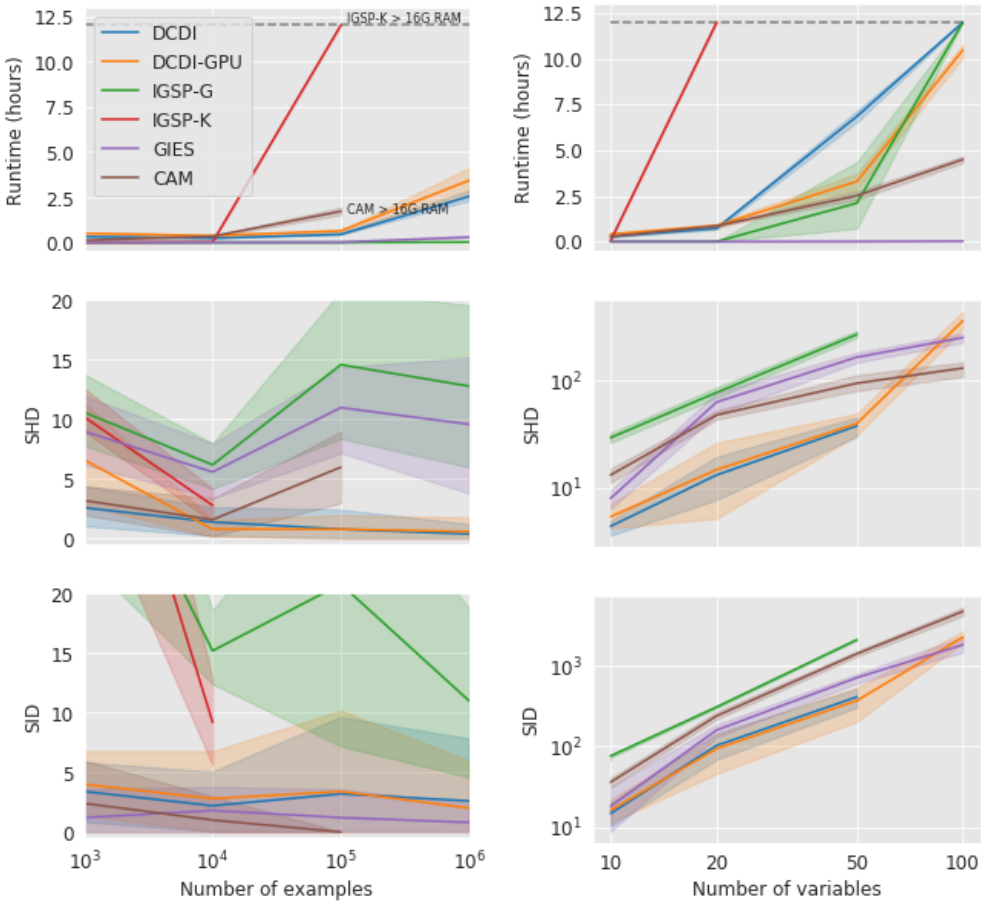


Figure 4.10. We report the runtime (in hours), SHD and SID of multiple methods in multiple settings. The horizontal dashed lines at 12 hours represents the time limit imposed. When a curve reaches this dashed line, it means that the method could not finish within 12 hours. We write $\geq$ 16G when the RAM memory needed by the algorithm exceeded 16GB. All data sets have 10 interventional targets containing $0.1d$ targets. We considered perfect interventions. **Left:** Different data set sizes. Ten nodes ANM data with connectivity $e = 1$. **Right:** Different number of variables. NN data set with connectivity $e = 4$ and 10^4 samples. Each curve is an average over 5 different datasets while the error bars are %95 confidence intervals computed via bootstrap.

Effect of number of interventions. In a small scale experiment, we show in Figure 4.11 the effect of the number of interventions on the performance of DCDI-G. The SHD and SID of DCDI-G and DCD are shown over ten linear data sets (20-node graph with sparse connectivity) with $\{0, 5, 10, 15, 20\}$ perfect interventions. The baseline DCD is equivalent to DCDI-G, but it uses a loss that doesn't take into account the interventions. It can first be noticed that, as the number of interventions increases, the performance of DCDI-G increases. This increase is particularly noticeable from the purely interventional data to data with 5 interventions. While DCD's performance also increases in terms of SHD, it seems to have no clear gain in terms of SID. Also, DCDI-G with interventional data is always better than DCD showing that the proposed loss

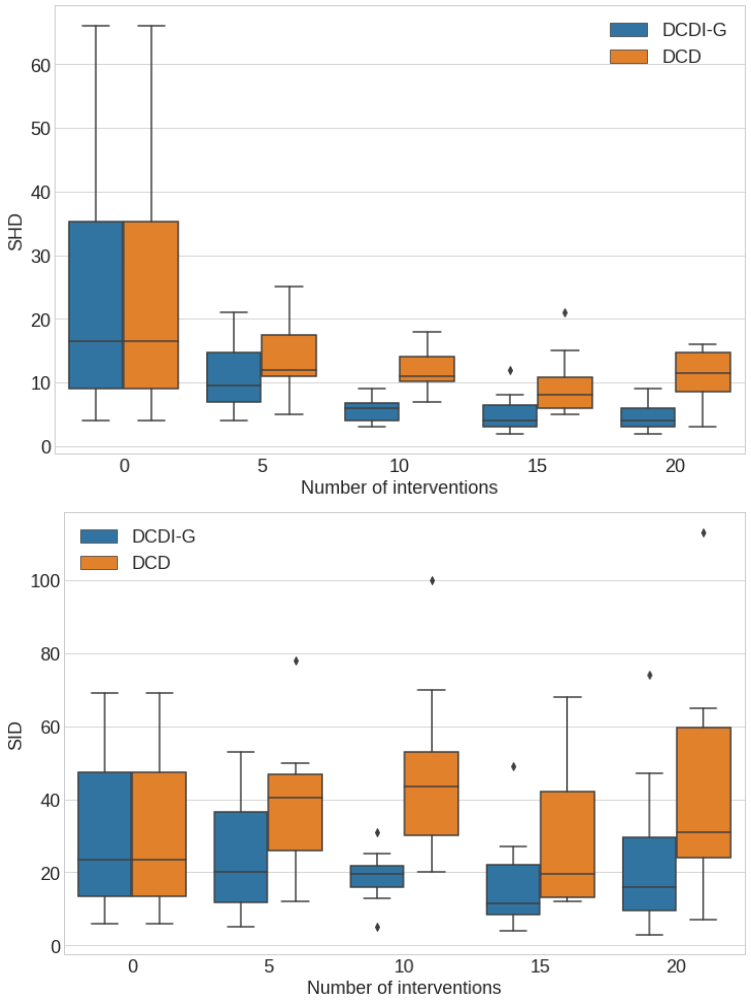


Figure 4.11. SHD and SID for DCDI-G and DCD on data sets with a different number of interventional settings.

for perfect interventions is pertinent. Note that the first two boxes are the same since DCDI-G on observational data is equivalent to DCD (the experiment was done only once).

Relevance of DCDI score to leverage interventional data. In a larger scale experiment, with the same data sets used in the main text (Section 4.4), we compare DCDI-G and DCDI-DSF to DCD and DCD-no-interv for perfect/known, imperfect/known and perfect/unknown interventions (shown respectively in Appendix C.4.1, C.4.2, and C.4.3). The values reported are the mean and the standard deviation of SHD and SID over ten data sets of each condition. DCD-no-interv is DCDI-G applied to purely observational data. These purely observational data sets were generated from the same CGM as the other data set containing interventions and had the same total sample size. For SHD, the advantage of DCDI over DCD and DCD-no-interv is clear over all conditions. For SID, DCDI has no advantage for sparse graphs, but is usually better for graphs with higher connectivity.

As in the first small scale experiment, the beneficial effect of interventions is clear. Also, these results show that the proposed losses for the different type of interventions are pertinent.

Relevance of neural network models. As a sanity check of our proposed method, we trained DCDI-G without hidden layers, i.e. a linear model. In Table 4.4, 4.5 and 4.6, we report the mean and standard deviation of SHD and SID over ten 20-node graphs for DCDI-linear and compare it to results obtained for DCDI-G and DCDI-DSF (both using hidden layers). As expected, this linear version of DCDI has competitive results for the linear data set, but poorer results on nonlinear data sets, showing the interest of using high capacity models.

	20 nodes, $e = 1$		20 nodes, $e = 4$	
Method	SHD	SID	SHD	SID
DCDI-linear	5.9 ± 7.6	7.1 ± 6.9	16.0 ± 6.7	98.3 ± 31.4
DCDI-G	5.4 ± 4.5	13.4 ± 12.0	23.7 ± 5.6	112.8 ± 41.8
DCDI-DSF	3.6 ± 2.7	6.0 ± 5.4	16.6 ± 6.4	92.5 ± 40.1

Table 4.4. Results for the linear data set with perfect intervention

	20 nodes, $e = 1$		20 nodes, $e = 4$	
Method	SHD	SID	SHD	SID
DCDI-linear	29.6 ± 15.4	24.8 ± 18.4	66.2 ± 13.7	219.0 ± 41.7
DCDI-G	21.8 ± 30.1	11.6 ± 13.1	35.2 ± 13.2	109.8 ± 44.6
DCDI-DSF	4.3 ± 1.9	19.7 ± 12.6	26.7 ± 16.9	105.3 ± 22.7

Table 4.5. Results for the additive noise model data set with perfect intervention

	20 nodes, $e = 1$		20 nodes, $e = 4$	
Method	SHD	SID	SHD	SID
DCDI-linear	19.8 ± 12.7	14.2 ± 9.2	45.6 ± 12.0	177.9 ± 27.6
DCDI-G	13.9 ± 20.3	13.7 ± 8.1	16.8 ± 8.7	82.5 ± 38.1
DCDI-DSF	8.3 ± 4.1	32.4 ± 17.3	11.8 ± 2.1	102.3 ± 34.5

Table 4.6. Results for the nonlinear with non-additive noise data set with perfect intervention

	10 nodes, $e = 1$		10 nodes, $e = 4$		20 nodes, $e = 1$		20 nodes, $e = 4$	
Method	SHD	SID	SHD	SID	SHD	SID	SHD	SID
DCD	6.6 ± 3.6	14.1 ± 11.5	24.4 ± 6.0	67.0 ± 9.2	18.2 ± 15.8	30.9 ± 21.7	56.7 ± 10.2	227.0 ± 38.6
DCD-no-interv	8.9 ± 2.8	19.5 ± 10.9	26.7 ± 5.9	69.0 ± 11.2	24.6 ± 20.5	31.2 ± 22.8	64.4 ± 11.4	292.9 ± 28.9
DCDI-G	1.3 ± 1.9	0.8 ± 1.8	3.3 ± 2.1	10.7 ± 12.0	5.4 ± 4.5	13.4 ± 12.0	23.7 ± 5.6	112.8 ± 41.8
DCDI-DSF	0.9 ± 1.3	0.6 ± 1.9	3.7 ± 2.3	18.9 ± 14.1	3.6 ± 2.7	6.0 ± 5.4	16.6 ± 6.4	92.5 ± 40.1

Table 4.7. Results for the linear data set with perfect intervention

	10 nodes, $e = 1$		10 nodes, $e = 4$		20 nodes, $e = 1$		20 nodes, $e = 4$	
Method	SHD	SID	SHD	SID	SHD	SID	SHD	SID
DCD	11.5 ± 6.6	18.2 ± 11.8	30.4 ± 3.8	75.5 ± 4.6	39.3 ± 28.4	39.8 ± 33.3	62.7 ± 14.2	241.0 ± 44.8
DCD-no-interv	11.6 ± 8.8	15.8 ± 12.1	21.3 ± 5.2	63.5 ± 12.3	41.7 ± 44.1	36.2 ± 27.1	43.7 ± 9.2	226.1 ± 42.8
DCDI-G	5.2 ± 7.5	2.4 ± 4.9	4.3 ± 2.4	16.0 ± 11.9	21.8 ± 30.1	11.6 ± 13.1	35.2 ± 13.2	109.8 ± 44.6
DCDI-DSF	4.2 ± 5.6	5.6 ± 5.5	5.5 ± 2.4	23.9 ± 14.3	4.3 ± 1.9	19.7 ± 12.6	26.7 ± 16.9	105.3 ± 22.7

Table 4.8. Results for the additive noise model data set with perfect intervention

	10 nodes, $e = 1$		10 nodes, $e = 4$		20 nodes, $e = 1$		20 nodes, $e = 4$	
Method	SHD	SID	SHD	SID	SHD	SID	SHD	SID
DCD	5.9 ± 6.9	10.9 ± 10.4	15.7 ± 4.9	53.0 ± 9.9	28.7 ± 13.0	29.7 ± 9.3	29.3 ± 8.9	163.1 ± 48.4
DCD-no-interv	11.0 ± 9.3	9.9 ± 11.0	18.4 ± 6.4	56.4 ± 11.0	16.5 ± 22.8	31.9 ± 17.5	31.6 ± 11.3	160.3 ± 46.3
DCDI-G	2.3 ± 3.6	2.7 ± 3.3	2.4 ± 1.6	13.9 ± 8.5	13.9 ± 20.3	13.7 ± 8.1	16.8 ± 8.7	82.5 ± 38.1
DCDI-DSF	7.0 ± 10.7	7.8 ± 5.8	1.6 ± 1.6	7.7 ± 13.8	8.3 ± 4.1	32.4 ± 17.3	11.8 ± 2.1	102.3 ± 34.5

Table 4.9. Results for the nonlinear with non-additive noise data set with perfect intervention

C.4.1. Perfect interventions.

	10 nodes, $e = 1$		10 nodes, $e = 4$		20 nodes, $e = 1$		20 nodes, $e = 4$	
Method	SHD	SID	SHD	SID	SHD	SID	SHD	SID
DCD	10.6 ± 5.4	24.6 ± 18.2	24.0 ± 4.1	67.2 ± 7.6	21.2 ± 11.5	56.0 ± 31.5	56.7 ± 9.0	268.0 ± 25.4
DCD-no-interv	6.8 ± 4.4	19.5 ± 13.2	27.4 ± 4.4	74.0 ± 7.2	19.8 ± 9.2	48.2 ± 30.6	58.2 ± 9.9	288.6 ± 31.6
DCDI-G	2.7 ± 2.8	8.2 ± 8.8	5.2 ± 3.5	25.1 ± 12.9	15.6 ± 14.5	29.1 ± 23.4	34.0 ± 7.7	180.9 ± 44.5
DCDI-DSF	1.3 ± 1.3	4.2 ± 4.0	1.7 ± 2.4	10.2 ± 14.9	6.9 ± 6.3	22.7 ± 21.9	21.7 ± 8.1	137.4 ± 34.3

Table 4.10. Results for the linear data set with imperfect intervention

	10 nodes, $e = 1$		10 nodes, $e = 4$		20 nodes, $e = 1$		20 nodes, $e = 4$	
Method	SHD	SID	SHD	SID	SHD	SID	SHD	SID
DCD	12.0 ± 9.2	14.8 ± 10.4	24.3 ± 3.8	64.5 ± 11.1	51.7 ± 41.7	44.5 ± 20.0	54.1 ± 12.0	196.6 ± 37.2
DCD-no-interv	14.6 ± 4.3	12.1 ± 11.8	24.8 ± 4.8	69.3 ± 8.3	49.5 ± 36.0	32.7 ± 22.7	41.2 ± 8.1	197.7 ± 50.1
DCDI-G	6.2 ± 5.4	7.6 ± 11.0	13.1 ± 2.9	48.1 ± 9.1	30.5 ± 33.0	12.5 ± 8.8	43.1 ± 10.2	96.6 ± 47.1
DCDI-DSF	13.4 ± 8.4	17.9 ± 10.5	14.4 ± 2.4	53.2 ± 8.2	13.1 ± 4.5	43.5 ± 19.2	50.5 ± 11.4	172.1 ± 19.6

Table 4.11. Results for the additive noise model data set with imperfect intervention

	10 nodes, $e = 1$		10 nodes, $e = 4$		20 nodes, $e = 1$		20 nodes, $e = 4$	
Method	SHD	SID	SHD	SID	SHD	SID	SHD	SID
DCD	12.7 ± 8.4	11.8 ± 7.3	15.2 ± 3.7	52.2 ± 9.1	40.4 ± 54.7	45.2 ± 43.9	30.5 ± 8.0	151.2 ± 41.7
DCD-no-interv	13.6 ± 9.7	13.0 ± 8.1	14.8 ± 3.5	51.7 ± 12.5	37.1 ± 40.7	57.1 ± 56.2	31.3 ± 5.5	162.3 ± 40.5
DCDI-G	3.9 ± 3.9	7.5 ± 6.5	7.3 ± 2.2	28.0 ± 10.5	18.2 ± 28.8	36.9 ± 37.0	21.7 ± 8.0	127.3 ± 40.1
DCDI-DSF	5.3 ± 4.2	16.3 ± 10.0	5.9 ± 3.2	35.1 ± 12.3	13.2 ± 5.1	76.5 ± 57.8	16.8 ± 5.3	143.6 ± 48.8

Table 4.12. Results for the nonlinear with non-additive noise data set with imperfect intervention

C.4.2. Imperfect interventions.

	10 nodes, $e = 1$		10 nodes, $e = 4$		20 nodes, $e = 1$		20 nodes, $e = 4$	
Method	SHD	SID	SHD	SID	SHD	SID	SHD	SID
DCD	6.6 ± 3.6	14.1 ± 11.5	24.4 ± 6.0	67.0 ± 9.2	18.2 ± 15.8	30.9 ± 21.7	56.7 ± 10.2	227.0 ± 38.6
DCD-no-interv	8.9 ± 2.8	19.5 ± 10.9	26.7 ± 5.9	69.0 ± 11.2	24.6 ± 20.5	31.2 ± 22.8	64.4 ± 11.4	292.9 ± 28.9
DCDI-G	5.3 ± 3.7	12.9 ± 11.5	5.2 ± 3.0	24.3 ± 15.3	15.4 ± 10.3	30.8 ± 18.6	39.2 ± 8.7	173.7 ± 45.6
DCDI-DSF	3.9 ± 4.3	7.1 ± 7.1	7.1 ± 3.6	35.8 ± 12.5	4.3 ± 2.4	18.4 ± 7.3	29.7 ± 12.6	147.8 ± 42.7

Table 4.13. Results for the linear data set with perfect intervention with unknown targets

	10 nodes, $e = 1$		10 nodes, $e = 4$		20 nodes, $e = 1$		20 nodes, $e = 4$	
Method	SHD	SID	SHD	SID	SHD	SID	SHD	SID
DCD	11.5 ± 6.6	18.2 ± 11.8	30.4 ± 3.8	75.5 ± 4.6	39.3 ± 28.4	39.8 ± 33.3	62.7 ± 14.2	241.0 ± 44.8
DCD-no-interv	11.6 ± 8.8	15.8 ± 12.1	21.3 ± 5.2	63.5 ± 12.3	41.7 ± 44.1	36.2 ± 27.1	43.7 ± 9.2	226.1 ± 42.8
DCDI-G	7.6 ± 10.3	5.0 ± 5.4	9.1 ± 3.8	37.5 ± 14.1	41.3 ± 39.2	22.9 ± 15.5	39.9 ± 18.8	153.7 ± 50.3
DCDI-DSF	11.9 ± 8.8	13.8 ± 7.9	6.6 ± 2.6	32.6 ± 14.1	22.3 ± 31.9	33.1 ± 17.5	42.5 ± 18.7	152.9 ± 53.4

Table 4.14. Results for the additive noise model data set with perfect intervention with unknown targets

	10 nodes, $e = 1$		10 nodes, $e = 4$		20 nodes, $e = 1$		20 nodes, $e = 4$	
Method	SHD	SID	SHD	SID	SHD	SID	SHD	SID
DCD	5.9 ± 6.9	10.9 ± 10.4	15.7 ± 4.9	53.0 ± 9.9	28.7 ± 13.0	29.7 ± 9.3	29.3 ± 8.9	163.1 ± 48.4
DCD-no-interv	11.0 ± 9.3	9.9 ± 11.0	18.4 ± 6.4	56.4 ± 11.0	16.5 ± 22.8	31.9 ± 17.5	31.6 ± 11.3	160.3 ± 46.3
DCDI-G	3.4 ± 4.2	6.9 ± 7.5	3.3 ± 1.3	20.4 ± 10.4	21.8 ± 32.1	20.9 ± 12.3	20.1 ± 8.1	104.6 ± 47.1
DCDI-DSF	7.8 ± 7.9	11.8 ± 5.7	3.3 ± 1.2	23.2 ± 9.1	27.4 ± 30.9	49.3 ± 15.7	22.2 ± 10.4	131.0 ± 41.0

Table 4.15. Results for the nonlinear with non-additive noise data set with perfect intervention with unknown targets

C.4.3. Unknown interventions.

C.5. Different kinds of interventions

In this section, we compare DCDI to IGSP using data sets under different kinds of interventions. We report results in tabular form for 10-node and 20-node graphs. For the perfect interventions, instead of replacing the target conditional distribution by the marginal $\mathcal{N}(2, 1)$ (as in the main results), we used a marginal that doesn't involve a mean-shift: $\mathcal{U}[-1, 1]$. The results reported in Tables 4.16, 4.17, 4.18 of Section C.5.1 are the mean and the standard deviation of SHD and SID over ten data sets of each condition. From these results, we can conclude that DCDI-G still outperforms IGSP and, by comparing to DCD (DCDI-G with a loss that doesn't take into account interventions), that the proposed loss is still beneficial for this kind of interventions. It has competitive results compared to GIES and CAM on the linear data set and it outperforms them on the other data sets.

For imperfect intervention, we tried more modest changes in the parameters. For the linear data set, an imperfect intervention consisted of adding $\mathcal{U}[0.5, 1]$ to w_j if $w_j > 0$ and subtracting if $w_j \leq 0$. It was done this way to ensure that the intervention would not remove dependencies between variables. For the additive noise model and the nonlinear with non-additive noise data sets, $\mathcal{N}(0, 0.1)$ was added to each weight of the neural networks. Results are reported in Tables 4.19, 4.20, 4.21 of Section C.5.2. These smaller changes made the difference between DCD and DCDI imperceptible. For sparse graphs, IGSP has a better or comparable performance to DCDI. For graphs with higher connectivity, DCDI often has a better performance than IGSP.

	10 nodes, $e = 1$		10 nodes, $e = 4$		20 nodes, $e = 1$		20 nodes, $e = 4$	
Method	SHD	SID	SHD	SID	SHD	SID	SHD	SID
IGSP	4.0 ± 4.8	15.7 ± 15.4	28.8 ± 2.0	72.2 ± 5.1	9.7 ± 8.7	45.1 ± 45.4	68.1 ± 13.6	295.4 ± 27.6
GIES	0.3 ± 0.5	0.0 ± 0.0	4.0 ± 6.5	6.7 ± 17.7	1.5 ± 1.2	0.3 ± 0.9	49.4 ± 22.2	111.9 ± 51.4
CAM	0.6 ± 1.0	0.0 ± 0.0	11.8 ± 4.3	32.2 ± 17.2	6.3 ± 7.4	7.6 ± 9.8	91.4 ± 21.3	181.7 ± 60.5
DCD	6.3 ± 3.4	14.8 ± 10.6	26.1 ± 3.3	66.4 ± 11.4	11.1 ± 4.7	45.8 ± 22.8	49.0 ± 12.0	258.6 ± 41.6
DCDI-G	0.4 ± 0.7	1.3 ± 2.1	7.5 ± 1.4	29.7 ± 8.2	3.2 ± 3.2	12.1 ± 11.2	21.0 ± 4.9	147.6 ± 49.5

Table 4.16. Results for the linear data set with perfect intervention

	10 nodes, $e = 1$		10 nodes, $e = 4$		20 nodes, $e = 1$		20 nodes, $e = 4$	
Method	SHD	SID	SHD	SID	SHD	SID	SHD	SID
IGSP	5.7 ± 2.3	23.4 ± 13.6	32.8 ± 2.4	79.3 ± 3.2	14.9 ± 8.1	78.8 ± 64.6	80.5 ± 6.4	337.6 ± 27.3
GIES	7.5 ± 5.1	2.3 ± 2.5	9.2 ± 2.9	27.1 ± 11.5	23.8 ± 18.4	3.1 ± 4.4	89.6 ± 14.7	143.9 ± 53.1
CAM	6.3 ± 6.9	0.0 ± 0.0	6.3 ± 3.8	14.6 ± 20.1	9.2 ± 14.3	13.5 ± 25.1	106.2 ± 14.6	96.2 ± 57.9
DCD	6.4 ± 4.6	22.0 ± 14.7	31.1 ± 3.4	77.4 ± 3.1	18.1 ± 8.0	51.5 ± 41.5	55.7 ± 8.3	261.3 ± 22.5
DCDI-G	0.9 ± 1.2	3.9 ± 6.4	5.2 ± 1.9	24.0 ± 9.3	6.5 ± 5.6	17.9 ± 19.1	26.8 ± 7.0	94.4 ± 41.5

Table 4.17. Results for the additive noise model data set with perfect intervention

	10 nodes, $e = 1$		10 nodes, $e = 4$		20 nodes, $e = 1$		20 nodes, $e = 4$	
Method	SHD	SID	SHD	SID	SHD	SID	SHD	SID
IGSP	6.6 ± 3.9	25.8 ± 17.9	31.1 ± 3.3	77.1 ± 5.7	14.4 ± 4.8	63.8 ± 26.5	79.7 ± 8.1	341.4 ± 18.1
GIES	6.2 ± 3.5	0.9 ± 1.5	9.5 ± 3.6	29.0 ± 17.7	12.2 ± 2.1	3.4 ± 3.2	63.8 ± 11.1	124.9 ± 36.9
CAM	4.1 ± 3.8	2.3 ± 3.4	11.3 ± 4.2	35.4 ± 20.8	4.2 ± 2.3	10.9 ± 10.3	106.6 ± 15.7	144.2 ± 51.8
DCD	6.6 ± 3.5	18.1 ± 8.1	20.6 ± 3.9	65.8 ± 9.9	9.4 ± 4.9	25.6 ± 16.2	28.6 ± 6.8	188.0 ± 28.7
DCDI-G	2.1 ± 1.5	4.6 ± 5.4	5.0 ± 4.3	28.8 ± 17.6	6.4 ± 3.8	15.1 ± 8.0	12.2 ± 2.7	96.1 ± 18.9

Table 4.18. Results for the nonlinear with non-additive noise data set with perfect intervention

C.5.1. Perfect interventions.

	10 nodes, $e = 1$		10 nodes, $e = 4$		20 nodes, $e = 1$		20 nodes, $e = 4$	
Method	SHD	SID	SHD	SID	SHD	SID	SHD	SID
IGSP	1.1 ± 1.1	5.4 ± 5.4	28.7 ± 3.2	72.4 ± 6.7	4.2 ± 3.9	17.7 ± 12.3	86.1 ± 12.3	289.8 ± 26.3
DCD	3.8 ± 3.6	9.4 ± 6.4	27.7 ± 3.4	74.6 ± 3.5	27.2 ± 22.3	39.3 ± 20.5	65.0 ± 8.0	306.8 ± 26.3
DCDI-G	4.7 ± 4.5	11.5 ± 9.5	27.4 ± 4.9	73.8 ± 5.4	29.6 ± 16.5	37.7 ± 14.5	62.8 ± 6.5	303.2 ± 27.6
DCDI-DSF	4.1 ± 2.3	10.3 ± 7.5	24.3 ± 5.3	69.1 ± 8.7	12.2 ± 2.9	42.6 ± 18.3	56.1 ± 9.2	291.4 ± 35.7

Table 4.19. Results for the linear data set with imperfect intervention

	10 nodes, $e = 1$		10 nodes, $e = 4$		20 nodes, $e = 1$		20 nodes, $e = 4$	
Method	SHD	SID	SHD	SID	SHD	SID	SHD	SID
IGSP	5.7 ± 4.0	17.4 ± 13.4	30.3 ± 4.0	73.9 ± 11.3	12.5 ± 6.6	44.9 ± 26.7	85.8 ± 4.4	344.0 ± 9.8
DCD	12.0 ± 10.3	11.3 ± 8.4	23.5 ± 2.1	69.7 ± 2.5	39.5 ± 42.3	28.2 ± 13.9	50.9 ± 7.1	247.8 ± 36.6
DCDI-G	12.7 ± 9.1	11.8 ± 6.5	21.7 ± 4.3	65.2 ± 9.2	16.2 ± 18.0	27.8 ± 13.1	46.2 ± 5.9	240.1 ± 26.3
DCDI-DSF	8.1 ± 8.2	15.8 ± 9.3	23.3 ± 6.3	68.7 ± 8.2	12.3 ± 4.1	39.9 ± 19.5	51.0 ± 7.1	257.7 ± 31.6

Table 4.20. Results for the additive noise model data set with imperfect intervention

	10 nodes, $e = 1$		10 nodes, $e = 4$		20 nodes, $e = 1$		20 nodes, $e = 4$	
Method	SHD	SID	SHD	SID	SHD	SID	SHD	SID
IGSP	7.0 ± 5.7	22.7 ± 19.5	29.4 ± 5.0	74.2 ± 7.3	18.7 ± 7.1	86.3 ± 37.1	81.6 ± 6.9	344.4 ± 20.5
DCD	9.4 ± 8.9	13.3 ± 11.0	15.1 ± 3.7	54.2 ± 9.8	28.5 ± 25.0	25.5 ± 16.8	32.7 ± 9.8	177.1 ± 37.5
DCDI-G	6.7 ± 5.1	13.0 ± 9.7	14.6 ± 3.3	53.9 ± 9.1	28.9 ± 33.7	25.2 ± 15.2	32.3 ± 7.9	177.0 ± 55.8
DCDI-DSF	12.8 ± 9.6	22.9 ± 14.8	14.4 ± 4.8	54.2 ± 10.3	13.3 ± 5.3	54.2 ± 20.9	28.6 ± 8.9	199.5 ± 32.7

Table 4.21. Results for the nonlinear with non-additive noise data set with imperfect intervention

C.5.2. Imperfect interventions.

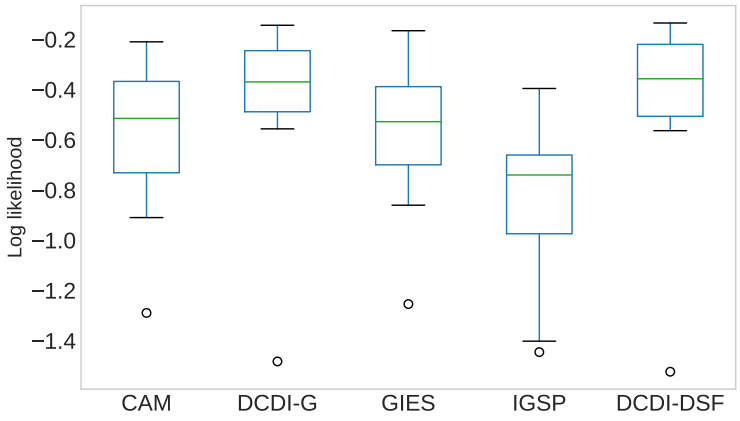


Figure 4.12. Log-likelihood on unseen interventional distributions of the nonlinear with non-additive noise data sets.

C.6. Evaluation on unseen interventional distributions

As advocated by Gentzel et al. [2019], we present *interventional performance measures* for the flow cytometry data set of Sachs et al. [2005] and for the nonlinear with non-additive noise data set. Interventional performance refers to the ability of the causal graph to model the effect of *unseen* interventions. To evaluate this, methods are trained on all the data, except for data coming from one interventional setting. Then, we evaluate the likelihood of the fitted model on the remaining *unseen* interventional distribution. Since some algorithms do not model distributions, for each method, given its estimated causal graph, we fit a distribution using a normalizing flow model, enabling a fair comparison. We report the log-likelihood evaluated on an unseen intervention. Note that when evaluating the likelihood, we ignore the conditional of the targeted node.

For the nonlinear data sets with non-additive noise, we report in Figure 4.12 boxplots over 10 dense graphs ($e = 4$) of 10 nodes. For each graph, one interventional setting was chosen randomly as the unseen intervention. DCDI-G and DCDI-DSF have the best performance, as was the case for the SHD and SID.

For Sachs, the data where intervention were applied on the protein *Akt* were used as the "held-out" distribution. We report in Figure 4.13 the log-likelihood and its standard deviation over these data samples. The ordering of the methods is different from the structural metrics: IGSP has the best performance followed by DCDI-G (whereas CAM seemed to have the best performance with the structural metrics).

C.7. Comprehensive results of the main experiments

In this section, we report the main results presented in Section 4.4 in tabular form for 10-node and 20-node graphs. Recall that the hyperparameters of DCDI, CAM and GIES were selected to yield

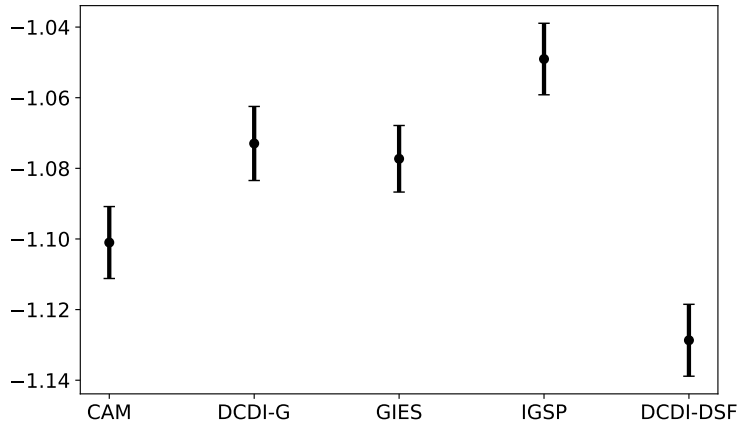


Figure 4.13. Log-likelihood on an unseen interventional distribution of the Sachs data set.

the best likelihood on a held-out data set. However, this is not possible for IGSP, UTIGSP and JCI-PC since they do not have a likelihood model. To make sure these algorithms are represented fairly, we report their performance for different hyperparameter values. For IGSP and UT-IGSP, we report performance for the cutoff hyperparameter $\alpha = \{2e-1, 1e-1, 1e-2, 1e-3, 1e-5, 1e-7, 1e-9\}$. This range was chosen to be around the cutoff values used in Wang et al. [2017] and Squires et al. [2020]. We used the same range for JCI-PC, but since most runs with $\alpha \leq 1e-5$ would not terminate after 12 hours, we only report results with $\alpha = \{2e-1, 1e-1, 1e-2, 1e-3\}$. The overall ranking of the methods does not change for different hyperparameters. To be even fairer to these methods, we also report the performance one obtains by selecting, for every data set, the hyperparameter which yields the lowest SHD. These results are denoted by IGSP*, UTIGSP* and JCI-PC*. Notice that this is unfair to DCDI, CAM and GIES which have not tuned their hyperparameters to minimize SHD or SID. Even in this unfair comparison, DCDI remains very competitive. For IGSP and UTIGSP, we also include results using partial correlation test (indicated with the suffix *-lin*) and KCI-test for every data sets. The reported values in the following tables are the mean and the standard deviation of SHD and SID over ten data sets of each condition. As stated in the main discussion, our conclusions are similar for 10-node graphs: DCDI has competitive performance in almost all conditions and outperforms the other methods for graphs with higher connectivity.

	10 nodes, $e = 1$		10 nodes, $e = 4$		20 nodes, $e = 1$		20 nodes, $e = 4$	
Method	SHD	SID	SHD	SID	SHD	SID	SHD	SID
IGSP*-lin	2.2 ± 2.0	11.5 ± 11.4	23.5 ± 1.8	67.3 ± 3.3	4.7 ± 3.7	19.1 ± 13.4	73.4 ± 7.9	291.6 ± 46.4
IGSP*	1.9 ± 1.8	8.9 ± 9.5	24.6 ± 3.3	69.0 ± 10.3	9.2 ± 4.8	42.5 ± 31.8	78.5 ± 6.8	337.0 ± 16.4
IGSP(α=2e-1)-lin	9.3 ± 4.1	18.5 ± 15.6	26.4 ± 3.9	71.2 ± 3.9	37.7 ± 10.7	42.9 ± 37.1	94.6 ± 8.9	271.8 ± 18.3
IGSP(α=1e-1)-lin	5.8 ± 3.5	17.1 ± 13.4	27.4 ± 2.8	71.6 ± 4.0	18.7 ± 4.4	25.9 ± 12.8	84.4 ± 12.2	264.8 ± 27.4
IGSP(α=1e-2)-lin	2.4 ± 2.1	11.8 ± 11.0	27.6 ± 4.2	70.9 ± 8.2	7.2 ± 5.3	22.8 ± 17.3	78.9 ± 10.6	278.7 ± 19.5
IGSP(α=1e-3)-lin	2.4 ± 2.1	11.8 ± 11.0	26.9 ± 4.0	68.3 ± 6.8	8.5 ± 7.2	33.3 ± 29.4	82.4 ± 12.1	304.3 ± 20.4
IGSP(α=1e-5)-lin	2.4 ± 2.1	11.9 ± 11.1	30.6 ± 3.9	74.8 ± 7.0	9.4 ± 5.4	41.1 ± 36.8	83.9 ± 11.1	327.8 ± 9.0
IGSP(α=1e-7)-lin	2.7 ± 2.5	13.8 ± 14.3	33.7 ± 3.3	78.8 ± 4.8	8.6 ± 5.1	44.2 ± 36.0	81.5 ± 10.6	338.7 ± 8.8
IGSP(α=1e-9)-lin	2.6 ± 2.5	13.4 ± 14.6	29.3 ± 3.4	71.0 ± 9.7	11.6 ± 5.1	65.1 ± 45.5	82.0 ± 6.4	341.5 ± 12.2
IGSP(α=2e-1)	8.1 ± 3.4	10.7 ± 11.2	28.6 ± 5.3	74.0 ± 6.3	51.8 ± 10.4	64.7 ± 46.5	102.4 ± 9.8	311.4 ± 13.8
IGSP(α=1e-1)	5.4 ± 2.8	13.1 ± 11.1	26.7 ± 3.7	69.5 ± 11.1	31.0 ± 8.6	52.0 ± 31.9	93.2 ± 8.2	314.3 ± 21.3
IGSP(α=1e-2)	2.5 ± 2.0	10.5 ± 10.3	31.0 ± 3.8	78.2 ± 4.8	12.1 ± 5.1	40.4 ± 22.6	86.8 ± 9.5	336.4 ± 16.4
IGSP(α=1e-3)	2.8 ± 2.5	13.1 ± 13.8	31.3 ± 2.9	76.0 ± 8.1	12.4 ± 4.7	55.6 ± 30.9	84.7 ± 10.1	346.3 ± 8.5
IGSP(α=1e-5)	2.9 ± 2.7	13.8 ± 14.6	33.3 ± 2.5	78.8 ± 7.1	12.9 ± 5.6	64.9 ± 35.3	84.4 ± 6.1	347.7 ± 14.0
IGSP(α=1e-7)	4.1 ± 3.9	15.6 ± 14.9	33.0 ± 3.3	77.7 ± 5.4	15.2 ± 7.2	75.6 ± 43.6	83.9 ± 6.6	350.1 ± 20.4
IGSP(α=1e-9)	4.0 ± 3.6	16.3 ± 17.9	33.6 ± 3.1	76.2 ± 5.6	16.7 ± 6.3	81.9 ± 35.7	83.0 ± 6.7	339.7 ± 13.8
GIES	0.6 ± 1.3	0.0 ± 0.0	2.9 ± 3.0	0.0 ± 0.0	3.2 ± 6.3	1.1 ± 3.5	53.1 ± 25.8	82.9 ± 84.9
CAM	1.9 ± 2.6	1.7 ± 3.1	10.6 ± 3.1	34.5 ± 11.0	5.4 ± 7.9	8.2 ± 9.6	91.1 ± 21.7	167.8 ± 55.4
DCDI-G	1.3 ± 1.9	0.8 ± 1.8	3.3 ± 2.1	10.7 ± 12.0	5.4 ± 4.5	13.4 ± 12.0	23.7 ± 5.6	112.8 ± 41.8
DCDI-DSF	0.9 ± 1.3	0.6 ± 1.9	3.7 ± 2.3	18.9 ± 14.1	3.6 ± 2.7	6.0 ± 5.4	16.6 ± 6.4	92.5 ± 40.1

Table 4.22. Results for linear data set with perfect intervention

	10 nodes, $e = 1$		10 nodes, $e = 4$		20 nodes, $e = 1$		20 nodes, $e = 4$	
Method	SHD	SID	SHD	SID	SHD	SID	SHD	SID
IGSP*-lin	7.7 ± 2.4	24.1 ± 11.1	22.5 ± 2.0	64.4 ± 6.3	14.2 ± 5.2	58.6 ± 37.5	75.9 ± 3.1	307.1 ± 25.0
IGSP*	5.3 ± 3.0	20.9 ± 13.9	25.8 ± 2.8	68.0 ± 9.4	13.6 ± 6.6	69.6 ± 47.9	76.7 ± 6.5	332.6 ± 18.2
IGSP(α=2e-1)-lin	17.0 ± 5.2	25.0 ± 13.1	27.3 ± 3.3	69.2 ± 7.0	56.3 ± 10.5	78.3 ± 47.5	125.3 ± 7.9	282.9 ± 27.2
IGSP(α=1e-1)-lin	13.2 ± 5.3	21.1 ± 9.8	27.3 ± 4.4	69.4 ± 5.6	42.0 ± 11.9	73.4 ± 37.5	115.8 ± 11.6	286.0 ± 34.6
IGSP(α=1e-2)-lin	11.4 ± 4.6	26.4 ± 13.9	27.8 ± 3.4	72.4 ± 4.2	21.5 ± 9.6	64.7 ± 42.0	101.0 ± 10.1	298.6 ± 20.2
IGSP(α=1e-3)-lin	10.4 ± 3.9	26.6 ± 11.8	26.9 ± 2.9	70.2 ± 7.3	19.0 ± 8.0	58.1 ± 34.2	93.2 ± 4.8	308.5 ± 18.3
IGSP(α=1e-5)-lin	9.7 ± 2.3	27.4 ± 8.8	28.2 ± 3.9	70.2 ± 9.9	20.1 ± 8.6	84.9 ± 49.1	82.9 ± 5.3	312.9 ± 19.6
IGSP(α=1e-7)-lin	9.2 ± 2.3	28.1 ± 10.4	27.9 ± 3.8	72.5 ± 8.2	16.1 ± 5.2	63.5 ± 37.3	84.1 ± 8.6	322.1 ± 22.4
IGSP(α=1e-9)-lin	9.8 ± 2.4	31.5 ± 12.3	30.9 ± 4.7	77.7 ± 5.4	17.2 ± 6.3	73.1 ± 37.3	78.7 ± 5.7	314.8 ± 23.9
IGSP(α=2e-1)	13.3 ± 4.9	23.2 ± 15.9	28.4 ± 3.3	71.5 ± 8.3	43.2 ± 7.6	55.8 ± 30.0	98.0 ± 11.2	302.3 ± 34.7
IGSP(α=1e-1)	9.7 ± 5.3	21.8 ± 14.6	29.0 ± 2.9	73.4 ± 4.9	30.6 ± 6.4	64.7 ± 41.5	88.9 ± 9.2	320.9 ± 16.2
IGSP(α=1e-2)	7.3 ± 3.3	21.9 ± 11.3	31.4 ± 2.5	74.3 ± 9.7	17.2 ± 6.0	74.7 ± 40.2	84.1 ± 10.1	322.8 ± 15.8
IGSP(α=1e-3)	7.8 ± 3.4	24.2 ± 12.1	29.6 ± 3.8	75.1 ± 5.6	16.5 ± 8.9	79.6 ± 53.6	85.1 ± 7.7	334.2 ± 22.0
IGSP(α=1e-5)	8.1 ± 4.0	29.2 ± 15.3	30.5 ± 4.2	77.3 ± 4.7	16.6 ± 6.6	79.7 ± 50.3	81.2 ± 8.2	324.4 ± 26.0
IGSP(α=1e-7)	7.3 ± 2.8	28.5 ± 11.1	33.0 ± 1.8	78.3 ± 4.0	15.3 ± 6.2	75.0 ± 45.4	82.5 ± 6.8	334.3 ± 22.8
IGSP(α=1e-9)	9.4 ± 5.2	34.3 ± 15.6	30.9 ± 3.9	73.7 ± 10.3	15.3 ± 6.7	78.2 ± 50.6	81.6 ± 10.8	333.4 ± 17.2
GIES	9.1 ± 8.5	1.8 ± 3.6	9.0 ± 2.7	23.8 ± 15.6	40.3 ± 61.0	7.5 ± 7.2	103.2 ± 18.6	120.1 ± 68.5
CAM	5.2 ± 3.0	1.0 ± 1.9	8.5 ± 3.7	11.5 ± 13.4	7.5 ± 6.0	5.6 ± 4.9	105.7 ± 13.2	108.7 ± 61.0
DCDI-G	5.2 ± 7.5	2.4 ± 4.9	4.3 ± 2.4	16.0 ± 11.9	21.8 ± 30.1	11.6 ± 13.1	35.2 ± 13.2	109.8 ± 44.6
DCDI-DSF	4.2 ± 5.6	5.6 ± 5.5	5.5 ± 2.4	23.9 ± 14.3	4.3 ± 1.9	19.7 ± 12.6	26.7 ± 16.9	105.3 ± 22.7

Table 4.23. Results for the additive noise model data set with perfect intervention

	10 nodes, $e = 1$		10 nodes, $e = 4$		20 nodes, $e = 1$		20 nodes, $e = 4$	
Method	SHD	SID	SHD	SID	SHD	SID	SHD	SID
IGSP*-lin	4.4 ± 2.6	15.2 ± 11.0	23.3 ± 1.9	66.0 ± 7.9	13.4 ± 3.2	67.4 ± 27.8	67.0 ± 9.6	318.4 ± 19.1
IGSP*	4.1 ± 2.8	13.6 ± 11.9	25.4 ± 3.8	69.4 ± 5.3	15.3 ± 4.5	73.0 ± 28.8	72.7 ± 9.6	329.4 ± 21.5
IGSP(α=2e-1)-lin	12.4 ± 4.5	15.2 ± 9.1	27.6 ± 3.9	70.1 ± 6.3	51.4 ± 9.1	72.5 ± 31.1	102.2 ± 11.5	297.1 ± 27.5
IGSP(α=1e-1)-lin	9.7 ± 4.7	17.5 ± 13.4	26.5 ± 3.1	68.5 ± 8.1	35.8 ± 9.2	83.4 ± 35.1	93.6 ± 10.2	293.9 ± 25.3
IGSP(α=1e-2)-lin	7.1 ± 3.4	16.4 ± 12.5	28.7 ± 2.7	72.7 ± 4.9	19.0 ± 5.1	73.7 ± 33.7	76.0 ± 12.9	315.7 ± 14.2
IGSP(α=1e-3)-lin	5.9 ± 3.5	15.9 ± 10.5	29.6 ± 3.0	75.0 ± 2.8	16.4 ± 4.4	77.1 ± 32.2	75.0 ± 11.0	325.1 ± 17.7
IGSP(α=1e-5)-lin	6.6 ± 3.0	21.1 ± 12.3	27.7 ± 3.4	73.6 ± 4.8	15.9 ± 5.7	79.6 ± 22.5	73.3 ± 12.7	323.2 ± 16.0
IGSP(α=1e-7)-lin	7.2 ± 4.3	24.3 ± 15.9	30.1 ± 4.1	75.4 ± 5.9	17.3 ± 3.9	84.1 ± 22.1	73.2 ± 11.2	325.5 ± 23.1
IGSP(α=1e-9)-lin	5.9 ± 3.5	20.9 ± 16.1	31.3 ± 2.1	76.6 ± 4.0	19.2 ± 4.2	94.4 ± 29.9	77.4 ± 11.3	347.2 ± 15.5
IGSP(α=2e-1)	10.6 ± 2.7	12.4 ± 4.9	27.0 ± 3.0	70.8 ± 4.1	48.2 ± 7.7	97.5 ± 29.8	89.5 ± 15.5	306.3 ± 17.1
IGSP(α=1e-1)	7.7 ± 4.1	12.1 ± 8.8	27.5 ± 5.0	73.0 ± 5.2	32.3 ± 7.1	87.5 ± 39.9	89.4 ± 16.4	325.4 ± 21.6
IGSP(α=1e-2)	5.4 ± 2.5	15.3 ± 6.4	29.5 ± 3.5	74.2 ± 4.9	19.5 ± 5.2	82.5 ± 38.5	83.0 ± 9.5	337.3 ± 15.9
IGSP(α=1e-3)	6.6 ± 4.1	21.7 ± 14.5	31.3 ± 3.8	75.9 ± 7.7	17.3 ± 6.1	83.3 ± 36.2	80.4 ± 11.9	331.0 ± 23.7
IGSP(α=1e-5)	6.3 ± 3.1	19.8 ± 12.1	34.0 ± 4.2	76.8 ± 12.0	19.3 ± 4.6	90.8 ± 32.6	77.0 ± 9.5	345.2 ± 9.8
IGSP(α=1e-7)	6.3 ± 3.3	21.4 ± 13.1	34.1 ± 1.9	78.5 ± 8.4	19.1 ± 4.0	91.6 ± 29.0	75.8 ± 11.1	344.4 ± 16.6
IGSP(α=1e-9)	5.9 ± 3.7	21.7 ± 15.9	34.6 ± 2.6	79.7 ± 6.2	18.8 ± 3.9	94.0 ± 33.8	77.5 ± 9.0	341.4 ± 24.5
GIES	4.4 ± 6.1	1.0 ± 1.6	7.9 ± 4.7	25.5 ± 13.2	26.9 ± 50.5	9.5 ± 7.4	80.1 ± 36.2	96.7 ± 59.1
CAM	1.8 ± 1.5	2.8 ± 4.4	7.9 ± 3.6	26.7 ± 19.0	6.1 ± 5.2	18.1 ± 16.3	101.8 ± 24.5	142.5 ± 49.1
DCDI-G	2.3 ± 3.6	2.7 ± 3.3	2.4 ± 1.6	13.9 ± 8.5	13.9 ± 20.3	13.7 ± 8.1	16.8 ± 8.7	82.5 ± 38.1
DCDI-DSF	7.0 ± 10.7	7.8 ± 5.8	1.6 ± 1.6	7.7 ± 13.8	8.3 ± 4.1	32.4 ± 17.3	11.8 ± 2.1	102.3 ± 34.5

Table 4.24. Results for the nonlinear with non-additive noise data set with perfect intervention

C.7.1. Perfect interventions.

	10 nodes, $e = 1$		10 nodes, $e = 4$		20 nodes, $e = 1$		20 nodes, $e = 4$	
Method	SHD	SID	SHD	SID	SHD	SID	SHD	SID
IGSP*-lin	2.1 ± 0.9	11.7 ± 6.7	20.7 ± 5.8	61.4 ± 11.0	4.0 ± 2.9	17.9 ± 12.9	62.2 ± 12.0	256.8 ± 35.5
IGSP*	3.4 ± 1.8	14.9 ± 12.4	24.1 ± 2.5	68.9 ± 9.3	8.0 ± 5.7	43.8 ± 33.6	75.3 ± 9.2	338.3 ± 22.3
IGSP(α=2e-1)-lin	8.5 ± 2.7	15.5 ± 8.0	23.2 ± 7.3	65.8 ± 11.3	45.3 ± 9.5	48.0 ± 28.4	86.1 ± 15.0	253.7 ± 29.8
IGSP(α=1e-1)-lin	4.5 ± 3.3	15.3 ± 10.8	24.4 ± 6.6	65.4 ± 12.6	23.4 ± 9.9	47.3 ± 31.8	80.5 ± 13.7	259.4 ± 27.2
IGSP(α=1e-2)-lin	2.8 ± 1.9	12.8 ± 6.6	26.1 ± 4.8	69.7 ± 8.8	6.6 ± 4.4	20.2 ± 13.3	68.2 ± 13.7	279.2 ± 22.4
IGSP(α=1e-3)-lin	3.9 ± 2.8	17.2 ± 9.1	26.4 ± 5.6	71.1 ± 9.7	7.0 ± 5.9	33.2 ± 26.3	70.6 ± 16.2	296.3 ± 20.8
IGSP(α=1e-5)-lin	4.3 ± 2.6	21.4 ± 13.4	29.2 ± 5.1	75.3 ± 7.4	8.1 ± 5.0	45.4 ± 39.9	75.5 ± 7.7	325.3 ± 21.3
IGSP(α=1e-7)-lin	3.4 ± 1.3	19.1 ± 10.1	29.1 ± 3.9	74.8 ± 6.6	10.7 ± 5.1	52.8 ± 33.3	77.9 ± 9.2	333.1 ± 16.7
IGSP(α=1e-9)-lin	4.6 ± 3.3	23.7 ± 20.4	31.3 ± 4.1	79.1 ± 5.7	10.5 ± 5.0	61.6 ± 33.9	78.0 ± 8.1	343.4 ± 23.9
IGSP(α=2e-1)	9.5 ± 3.6	21.5 ± 13.6	27.7 ± 5.4	70.9 ± 10.4	46.9 ± 10.3	64.1 ± 34.6	95.5 ± 8.6	306.0 ± 20.0
IGSP(α=1e-1)	5.6 ± 2.2	15.9 ± 16.0	26.8 ± 5.3	68.8 ± 9.8	32.3 ± 9.6	54.3 ± 30.5	89.0 ± 9.7	315.5 ± 20.6
IGSP(α=1e-2)	5.0 ± 2.8	20.2 ± 15.3	32.0 ± 3.2	76.3 ± 5.3	11.8 ± 9.1	48.8 ± 43.6	82.7 ± 12.5	339.2 ± 11.7
IGSP(α=1e-3)	4.0 ± 2.7	19.9 ± 14.3	31.0 ± 4.1	76.4 ± 6.8	10.8 ± 6.0	56.6 ± 32.3	82.6 ± 8.6	347.3 ± 8.3
IGSP(α=1e-5)	5.4 ± 4.4	23.3 ± 19.8	30.9 ± 4.1	80.4 ± 2.9	12.7 ± 6.9	71.2 ± 41.5	80.3 ± 9.6	347.6 ± 12.6
IGSP(α=1e-7)	5.1 ± 2.4	21.6 ± 12.7	31.4 ± 2.7	79.5 ± 3.4	13.8 ± 7.4	80.4 ± 42.1	82.2 ± 7.3	351.0 ± 13.7
IGSP(α=1e-9)	6.5 ± 3.3	28.0 ± 18.4	30.6 ± 3.9	78.3 ± 4.4	15.3 ± 7.7	80.3 ± 45.2	83.0 ± 8.8	351.4 ± 8.6
GIES	13.7 ± 11.9	20.9 ± 19.4	14.2 ± 7.1	47.1 ± 16.8	33.7 ± 48.8	20.8 ± 22.4	78.7 ± 40.4	194.1 ± 61.0
CAM	8.1 ± 6.2	22.6 ± 18.8	19.4 ± 4.7	56.0 ± 10.1	10.5 ± 5.8	36.3 ± 23.6	111.7 ± 16.5	232.5 ± 23.4
DCDI-G	2.7 ± 2.8	8.2 ± 8.8	5.2 ± 3.5	25.1 ± 12.9	10.8 ± 12.0	27.0 ± 21.3	34.7 ± 7.1	188.0 ± 48.8
DCDI-DSF	1.3 ± 1.3	4.2 ± 4.0	1.7 ± 2.4	10.2 ± 14.9	7.0 ± 4.0	21.0 ± 12.5	18.9 ± 5.9	133.6 ± 33.9

Table 4.25. Results for the linear data set with imperfect intervention

	10 nodes, $e = 1$		10 nodes, $e = 4$		20 nodes, $e = 1$		20 nodes, $e = 4$	
Method	SHD	SID	SHD	SID	SHD	SID	SHD	SID
IGSP*-lin	9.1 ± 4.4	23.6 ± 12.7	22.8 ± 4.6	62.0 ± 9.7	18.1 ± 6.0	67.5 ± 26.2	81.2 ± 8.8	322.9 ± 13.8
IGSP*	6.2 ± 3.4	15.8 ± 9.9	27.6 ± 2.3	67.2 ± 8.7	17.0 ± 3.9	79.7 ± 33.9	75.7 ± 7.4	321.0 ± 23.8
IGSP(α=2e-1)-lin	19.7 ± 3.4	29.9 ± 14.9	26.0 ± 5.0	67.0 ± 11.7	59.0 ± 11.9	87.0 ± 40.0	123.7 ± 10.4	279.5 ± 27.6
IGSP(α=1e-1)-lin	17.8 ± 5.5	35.4 ± 14.1	26.1 ± 5.5	68.9 ± 9.5	40.1 ± 12.0	71.4 ± 39.3	119.2 ± 10.8	285.5 ± 21.3
IGSP(α=1e-2)-lin	13.0 ± 4.7	28.1 ± 12.0	27.7 ± 3.0	70.0 ± 5.8	24.9 ± 9.9	67.1 ± 35.0	109.6 ± 11.6	291.6 ± 29.8
IGSP(α=1e-3)-lin	13.1 ± 6.0	30.6 ± 16.0	28.7 ± 3.6	71.8 ± 6.2	24.4 ± 9.0	68.8 ± 24.9	96.5 ± 10.6	303.7 ± 17.3
IGSP(α=1e-5)-lin	11.5 ± 7.3	31.0 ± 17.4	28.8 ± 6.0	69.6 ± 12.8	21.6 ± 5.1	81.3 ± 32.2	90.4 ± 10.8	314.1 ± 15.3
IGSP(α=1e-7)-lin	10.6 ± 5.8	31.0 ± 15.8	29.5 ± 5.0	74.1 ± 8.1	23.3 ± 5.1	93.2 ± 35.9	84.2 ± 8.9	329.3 ± 15.6
IGSP(α=1e-9)-lin	11.0 ± 6.4	34.0 ± 20.7	29.7 ± 2.8	69.7 ± 9.5	21.3 ± 5.7	86.3 ± 29.7	83.4 ± 8.1	328.5 ± 19.2
IGSP(α=2e-1)	11.4 ± 4.2	23.8 ± 16.0	29.0 ± 3.2	72.1 ± 7.5	48.0 ± 8.3	77.8 ± 42.6	97.5 ± 12.8	307.5 ± 23.7
IGSP(α=1e-1)	10.6 ± 5.1	26.2 ± 15.8	31.3 ± 3.3	73.7 ± 7.1	36.9 ± 6.1	86.9 ± 42.6	88.8 ± 11.1	318.5 ± 25.8
IGSP(α=1e-2)	9.1 ± 4.4	24.3 ± 11.5	32.4 ± 4.1	76.9 ± 6.8	20.9 ± 6.2	84.8 ± 39.9	86.1 ± 8.4	334.3 ± 14.2
IGSP(α=1e-3)	8.2 ± 4.5	24.5 ± 13.5	32.7 ± 2.2	78.2 ± 8.3	19.3 ± 4.4	78.8 ± 32.2	82.9 ± 5.7	325.1 ± 19.7
IGSP(α=1e-5)	8.0 ± 3.8	25.8 ± 14.2	33.8 ± 2.4	79.4 ± 4.1	21.4 ± 5.4	91.8 ± 40.5	83.1 ± 7.8	343.4 ± 14.3
IGSP(α=1e-7)	8.4 ± 4.3	27.6 ± 15.3	33.2 ± 1.9	78.1 ± 5.9	20.3 ± 4.7	87.2 ± 39.6	85.6 ± 7.4	334.9 ± 25.2
IGSP(α=1e-9)	8.4 ± 4.5	28.3 ± 16.3	34.4 ± 3.4	79.9 ± 4.4	19.6 ± 3.1	90.1 ± 33.1	79.1 ± 7.4	332.5 ± 20.9
GIES	19.9 ± 10.4	23.0 ± 10.1	18.9 ± 6.0	59.5 ± 11.2	74.4 ± 59.8	56.4 ± 43.1	112.2 ± 23.8	245.2 ± 36.1
CAM	11.2 ± 9.3	7.8 ± 8.7	9.6 ± 3.0	25.2 ± 10.8	16.3 ± 9.9	26.7 ± 27.2	121.9 ± 11.6	155.4 ± 41.5
DCDI-G	6.2 ± 5.4	7.6 ± 11.0	13.1 ± 2.9	48.1 ± 9.1	26.0 ± 34.6	23.3 ± 25.7	36.4 ± 13.4	88.5 ± 43.8
DCDI-DSF	13.4 ± 8.4	17.9 ± 10.5	14.4 ± 2.4	53.2 ± 8.2	15.2 ± 2.7	49.4 ± 26.7	44.6 ± 15.4	149.8 ± 26.0

Table 4.26. Results for the additive noise model data set with imperfect intervention

	10 nodes, $e = 1$		10 nodes, $e = 4$		20 nodes, $e = 1$		20 nodes, $e = 4$	
Method	SHD	SID	SHD	SID	SHD	SID	SHD	SID
IGSP*-lin	5.6 ± 3.6	23.0 ± 19.6	22.5 ± 2.9	63.4 ± 6.7	13.8 ± 6.9	86.0 ± 71.7	65.1 ± 12.0	315.4 ± 46.2
IGSP*	5.1 ± 4.3	20.8 ± 16.5	24.3 ± 2.9	69.1 ± 5.5	18.2 ± 7.9	100.3 ± 74.7	71.7 ± 5.2	331.7 ± 35.9
IGSP(α=2e-1)-lin	14.1 ± 6.1	30.8 ± 21.8	26.9 ± 4.1	70.1 ± 5.8	49.7 ± 13.2	89.7 ± 64.3	100.2 ± 8.8	297.2 ± 13.9
IGSP(α=1e-1)-lin	9.8 ± 4.8	24.9 ± 23.0	25.5 ± 4.6	68.1 ± 7.1	39.7 ± 12.3	104.9 ± 62.7	90.2 ± 13.0	289.0 ± 32.7
IGSP(α=1e-2)-lin	8.0 ± 4.4	29.6 ± 22.6	26.4 ± 3.8	69.9 ± 4.0	18.1 ± 8.2	88.6 ± 58.7	70.6 ± 13.0	301.0 ± 40.9
IGSP(α=1e-3)-lin	7.6 ± 4.8	26.9 ± 22.4	28.4 ± 2.3	73.7 ± 3.7	16.3 ± 8.8	88.5 ± 72.6	72.9 ± 8.7	326.0 ± 18.1
IGSP(α=1e-5)-lin	7.7 ± 5.3	29.2 ± 24.7	27.2 ± 4.0	69.3 ± 8.6	18.9 ± 6.9	112.2 ± 64.6	70.7 ± 9.8	320.2 ± 27.6
IGSP(α=1e-7)-lin	6.7 ± 4.6	26.3 ± 19.9	28.8 ± 3.9	73.1 ± 5.8	16.8 ± 7.2	106.1 ± 63.8	72.6 ± 9.9	338.0 ± 17.2
IGSP(α=1e-9)-lin	7.7 ± 4.3	29.2 ± 17.9	30.0 ± 3.2	74.4 ± 7.4	17.7 ± 6.8	119.8 ± 77.9	72.3 ± 9.6	337.1 ± 23.8
IGSP(α=2e-1)	12.5 ± 5.5	27.9 ± 21.0	26.7 ± 4.4	71.7 ± 4.2	52.9 ± 6.6	113.0 ± 64.2	91.7 ± 7.6	311.0 ± 15.9
IGSP(α=1e-1)	9.5 ± 5.4	26.7 ± 24.0	26.2 ± 4.7	70.6 ± 6.4	37.1 ± 10.1	113.0 ± 79.7	79.5 ± 9.0	318.2 ± 30.3
IGSP(α=1e-2)	7.3 ± 4.5	26.9 ± 19.4	28.4 ± 3.3	73.9 ± 4.3	20.9 ± 7.7	100.1 ± 71.9	77.5 ± 7.5	324.7 ± 28.7
IGSP(α=1e-3)	7.4 ± 5.2	29.8 ± 21.8	29.6 ± 2.9	76.0 ± 3.0	22.4 ± 7.8	125.9 ± 89.4	76.2 ± 7.6	343.4 ± 21.3
IGSP(α=1e-5)	6.6 ± 5.1	24.9 ± 20.4	31.0 ± 2.4	76.5 ± 4.7	19.6 ± 8.4	114.6 ± 79.9	74.4 ± 5.4	335.7 ± 24.3
IGSP(α=1e-7)	6.8 ± 5.2	25.5 ± 20.2	32.6 ± 3.3	77.7 ± 7.2	21.3 ± 10.0	129.2 ± 92.4	76.4 ± 5.6	341.0 ± 26.0
IGSP(α=1e-9)	6.8 ± 4.4	25.7 ± 18.9	33.0 ± 2.4	77.2 ± 6.7	21.3 ± 9.1	127.6 ± 92.8	76.8 ± 6.5	348.4 ± 18.5
GIES	13.2 ± 11.2	16.7 ± 13.9	18.1 ± 5.6	53.7 ± 15.0	36.8 ± 41.1	67.0 ± 46.3	92.7 ± 29.4	215.8 ± 63.9
CAM	4.3 ± 3.3	9.3 ± 6.8	14.7 ± 5.1	45.7 ± 14.9	20.7 ± 16.2	53.9 ± 32.9	121.5 ± 9.3	194.1 ± 40.3
DCDI-G	3.9 ± 3.9	7.5 ± 6.5	7.4 ± 2.7	29.8 ± 11.0	10.0 ± 14.0	39.2 ± 41.5	20.9 ± 7.2	124.0 ± 39.0
DCDI-DSF	5.3 ± 4.2	16.3 ± 10.0	5.6 ± 3.1	32.4 ± 14.6	12.4 ± 5.3	70.3 ± 55.2	16.4 ± 4.9	139.7 ± 42.6

Table 4.27. Results for the nonlinear with non-additive noise data set with imperfect intervention

C.7.2. Imperfect interventions.

	10 nodes, $e = 1$		10 nodes, $e = 4$		20 nodes, $e = 1$		20 nodes, $e = 4$	
Method	SHD	SID	SHD	SID	SHD	SID	SHD	SID
UTIGSP*-lin	0.7 ± 1.6	3.4 ± 8.4	21.1 ± 3.6	62.9 ± 6.0	3.9 ± 3.6	14.6 ± 9.1	67.9 ± 10.8	271.6 ± 38.6
UTIGSP*	1.7 ± 2.0	7.4 ± 10.7	25.8 ± 2.5	67.4 ± 8.7	14.3 ± 4.8	65.5 ± 32.2	77.9 ± 5.5	332.2 ± 19.7
UTIGSP(α=2e-1)-lin	7.7 ± 3.7	15.1 ± 15.4	24.5 ± 6.1	67.6 ± 8.0	37.6 ± 10.2	44.4 ± 32.6	95.9 ± 9.7	265.6 ± 24.5
UTIGSP(α=1e-1)-lin	3.7 ± 3.2	10.2 ± 12.6	26.4 ± 2.9	68.9 ± 6.5	18.4 ± 5.1	16.8 ± 7.4	83.4 ± 13.1	255.8 ± 20.3
UTIGSP(α=1e-2)-lin	1.7 ± 2.1	7.0 ± 9.3	27.2 ± 5.8	70.1 ± 9.8	4.6 ± 4.0	13.9 ± 11.1	70.1 ± 12.0	271.2 ± 19.9
UTIGSP(α=1e-3)-lin	1.6 ± 2.2	7.2 ± 10.1	29.6 ± 5.5	73.1 ± 9.4	6.9 ± 6.5	25.6 ± 31.6	81.0 ± 12.7	301.1 ± 17.6
UTIGSP(α=1e-5)-lin	1.2 ± 1.9	5.1 ± 8.7	29.4 ± 4.2	73.2 ± 7.1	8.8 ± 6.0	36.7 ± 29.9	81.5 ± 11.7	323.1 ± 14.1
UTIGSP(α=1e-7)-lin	1.8 ± 2.6	7.6 ± 13.4	29.4 ± 3.4	72.3 ± 9.6	8.8 ± 5.5	43.3 ± 40.1	84.8 ± 9.7	339.6 ± 11.8
UTIGSP(α=1e-9)-lin	1.8 ± 2.4	7.8 ± 13.5	29.2 ± 3.8	70.2 ± 7.5	11.6 ± 7.3	57.3 ± 48.4	81.2 ± 5.7	339.4 ± 13.7
UTIGSP(α=2e-1)	8.5 ± 3.0	9.6 ± 8.6	27.8 ± 4.7	70.7 ± 10.4	50.3 ± 15.2	65.1 ± 49.2	106.7 ± 9.7	315.7 ± 24.0
UTIGSP(α=1e-1)	6.2 ± 3.2	13.0 ± 10.9	30.5 ± 2.4	74.3 ± 6.7	32.5 ± 7.0	57.5 ± 35.9	97.4 ± 9.8	317.5 ± 22.1
UTIGSP(α=1e-2)	2.6 ± 2.7	8.6 ± 9.7	30.4 ± 4.0	74.6 ± 7.3	17.9 ± 5.6	60.5 ± 27.1	85.9 ± 8.1	328.2 ± 20.1
UTIGSP(α=1e-3)	2.7 ± 2.2	9.3 ± 10.2	32.1 ± 3.0	78.1 ± 4.6	16.9 ± 6.5	70.2 ± 34.1	83.2 ± 8.6	341.4 ± 8.0
UTIGSP(α=1e-5)	4.3 ± 2.6	15.2 ± 11.5	31.5 ± 2.2	78.4 ± 8.0	17.0 ± 6.6	82.8 ± 37.4	82.2 ± 5.2	344.2 ± 14.1
UTIGSP(α=1e-7)	5.0 ± 3.9	18.2 ± 16.6	32.0 ± 2.8	77.1 ± 5.9	19.5 ± 6.9	89.7 ± 37.7	82.8 ± 4.9	346.0 ± 17.4
UTIGSP(α=1e-9)	6.0 ± 3.7	22.2 ± 18.0	31.7 ± 3.8	73.6 ± 7.1	18.8 ± 6.7	87.4 ± 41.2	81.4 ± 5.7	345.8 ± 15.4
JCI-PC*	5.7 ± 2.6	23.6 ± 13.2	35.9 ± 1.7	83.0 ± 6.5	13.1 ± 3.5	77.4 ± 22.2	76.2 ± 7.0	341.9 ± 22.5
JCI-PC(α=2e-1)	7.4 ± 2.1	28.4 ± 13.8	36.1 ± 1.8	83.2 ± 6.7	17.6 ± 4.2	84.9 ± 26.2	76.2 ± 7.0	341.9 ± 22.5
JCI-PC(α=1e-1)	6.9 ± 2.0	26.2 ± 13.0	36.1 ± 1.8	83.2 ± 6.7	15.2 ± 3.7	83.1 ± 25.3	76.2 ± 7.0	341.9 ± 22.5
JCI-PC(α=1e-2)	5.9 ± 2.3	23.6 ± 13.2	36.1 ± 1.8	83.2 ± 6.7	13.4 ± 3.4	79.0 ± 23.1	76.2 ± 7.0	341.9 ± 22.5
JCI-PC(α=1e-3)	5.7 ± 2.6	23.6 ± 13.2	36.1 ± 1.8	83.2 ± 6.7	13.1 ± 3.5	77.4 ± 22.2	76.2 ± 7.0	341.9 ± 22.5
DCDI-G	10.1 ± 4.2	12.4 ± 8.6	16.4 ± 5.3	52.3 ± 15.2	14.3 ± 18.8	23.3 ± 13.6	59.9 ± 10.5	237.6 ± 40.8
DCDI-DSF	4.4 ± 5.3	9.4 ± 9.4	9.3 ± 4.0	36.9 ± 11.9	4.9 ± 3.1	20.0 ± 12.0	32.5 ± 7.8	161.3 ± 37.1

Table 4.28. Results for the linear data set with perfect intervention with unknown targets

	10 nodes, $e = 1$		10 nodes, $e = 4$		20 nodes, $e = 1$		20 nodes, $e = 4$	
Method	SHD	SID	SHD	SID	SHD	SID	SHD	SID
UTIGSP*-lin	7.1 ± 2.3	20.5 ± 12.5	22.6 ± 3.0	59.2 ± 12.6	14.1 ± 4.8	56.8 ± 32.0	76.4 ± 5.7	312.5 ± 24.3
UTIGSP*	7.0 ± 4.3	20.6 ± 13.7	24.9 ± 2.3	70.8 ± 5.9	16.8 ± 7.0	87.1 ± 52.7	77.9 ± 6.6	333.4 ± 18.7
UTIGSP(α=2e-1)-lin	16.9 ± 4.1	24.2 ± 12.5	25.9 ± 5.0	66.5 ± 9.3	58.0 ± 10.8	73.7 ± 31.9	125.5 ± 11.0	275.8 ± 23.0
UTIGSP(α=1e-1)-lin	13.8 ± 6.0	20.8 ± 15.0	26.9 ± 4.1	67.1 ± 11.8	40.0 ± 11.7	67.0 ± 50.1	117.9 ± 6.3	290.7 ± 16.1
UTIGSP(α=1e-2)-lin	11.2 ± 4.3	25.2 ± 13.1	26.4 ± 4.6	66.5 ± 13.4	20.5 ± 10.5	54.8 ± 41.6	101.5 ± 7.6	298.6 ± 19.3
UTIGSP(α=1e-3)-lin	10.3 ± 3.7	28.1 ± 13.2	26.2 ± 3.6	64.6 ± 7.5	17.3 ± 7.2	47.6 ± 24.4	94.5 ± 7.9	306.8 ± 20.1
UTIGSP(α=1e-5)-lin	9.3 ± 2.5	27.4 ± 9.8	29.0 ± 3.7	73.0 ± 5.4	18.3 ± 6.9	73.0 ± 42.4	87.9 ± 7.8	325.2 ± 14.9
UTIGSP(α=1e-7)-lin	8.1 ± 2.1	24.9 ± 11.6	28.2 ± 3.7	72.4 ± 8.6	16.6 ± 5.7	65.8 ± 40.3	80.2 ± 8.4	316.4 ± 22.1
UTIGSP(α=1e-9)-lin	8.2 ± 2.8	27.5 ± 10.7	30.7 ± 3.9	76.7 ± 5.3	16.7 ± 5.9	70.2 ± 42.0	78.3 ± 4.0	318.9 ± 20.7
UTIGSP(α=2e-1)	13.5 ± 3.9	22.2 ± 17.2	27.6 ± 3.7	73.7 ± 3.5	45.6 ± 9.3	66.2 ± 43.7	98.6 ± 10.0	297.3 ± 36.4
UTIGSP(α=1e-1)	10.6 ± 6.1	20.1 ± 12.8	26.7 ± 2.9	71.9 ± 6.7	31.3 ± 5.3	68.3 ± 45.8	87.8 ± 10.0	301.0 ± 35.3
UTIGSP(α=1e-2)	9.1 ± 4.2	25.3 ± 10.3	29.0 ± 2.6	73.1 ± 3.1	20.8 ± 7.6	97.6 ± 53.0	84.4 ± 9.6	328.2 ± 17.4
UTIGSP(α=1e-3)	10.4 ± 4.1	28.1 ± 12.9	30.5 ± 4.7	77.8 ± 5.4	18.6 ± 7.0	84.5 ± 45.4	83.6 ± 5.3	335.0 ± 25.3
UTIGSP(α=1e-5)	9.9 ± 4.3	33.6 ± 12.0	32.1 ± 3.9	77.4 ± 6.7	19.5 ± 6.6	95.6 ± 50.9	81.9 ± 7.1	341.3 ± 12.1
UTIGSP(α=1e-7)	9.4 ± 4.9	33.3 ± 14.4	33.7 ± 3.9	76.8 ± 9.4	18.5 ± 6.9	92.3 ± 49.0	83.3 ± 8.1	337.5 ± 21.5
UTIGSP(α=1e-9)	9.4 ± 5.2	32.1 ± 15.2	33.0 ± 4.2	77.7 ± 8.7	18.7 ± 6.8	93.8 ± 52.0	82.9 ± 7.0	329.4 ± 28.2
JCI-PC*	8.5 ± 2.7	33.6 ± 12.0	35.5 ± 3.0	76.5 ± 8.7	15.2 ± 5.0	90.8 ± 52.1	72.4 ± 5.4	330.6 ± 12.8
JCI-PC(α=2e-1)	10.2 ± 3.3	35.8 ± 13.1	35.5 ± 3.0	75.6 ± 8.0	21.0 ± 3.6	92.0 ± 49.6	72.9 ± 5.4	328.7 ± 13.8
JCI-PC(α=1e-1)	9.5 ± 3.0	35.2 ± 12.9	35.5 ± 3.0	75.6 ± 8.0	17.5 ± 3.8	91.2 ± 51.2	72.9 ± 5.4	328.7 ± 13.8
JCI-PC(α=1e-2)	9.1 ± 3.0	35.4 ± 13.8	35.5 ± 3.0	75.6 ± 8.0	15.2 ± 5.0	90.8 ± 52.1	72.5 ± 5.4	330.5 ± 12.9
JCI-PC(α=1e-3)	8.6 ± 2.8	33.7 ± 12.1	35.5 ± 3.0	75.6 ± 8.0	15.2 ± 5.0	90.8 ± 52.1	72.4 ± 5.4	330.6 ± 12.8
DCDI-G	18.2 ± 10.1	16.4 ± 5.8	20.4 ± 6.8	64.8 ± 10.4	28.0 ± 33.5	39.1 ± 29.5	65.5 ± 11.6	249.8 ± 26.1
DCDI-DSF	10.6 ± 7.0	15.3 ± 10.5	9.1 ± 3.8	42.2 ± 12.4	28.0 ± 29.9	37.8 ± 22.6	42.4 ± 15.6	168.5 ± 37.8

Table 4.29. Results for the additive noise model data set with perfect intervention with unknown targets

	10 nodes, $e = 1$		10 nodes, $e = 4$		20 nodes, $e = 1$		20 nodes, $e = 4$	
Method	SHD	SID	SHD	SID	SHD	SID	SHD	SID
UTIGSP*-lin	3.6 ± 2.2	14.5 ± 11.1	23.1 ± 3.4	66.3 ± 6.4	13.7 ± 3.6	67.2 ± 28.8	68.0 ± 11.8	323.6 ± 15.7
UTIGSP*	4.1 ± 2.7	13.9 ± 9.5	24.2 ± 3.8	64.2 ± 11.1	17.8 ± 3.7	87.2 ± 25.8	73.4 ± 7.6	328.7 ± 24.9
UTIGSP(α=2e-1)-lin	11.3 ± 2.8	13.7 ± 6.9	24.7 ± 4.7	67.5 ± 7.4	50.2 ± 5.4	66.2 ± 29.4	104.3 ± 13.6	292.4 ± 18.5
UTIGSP(α=1e-1)-lin	8.5 ± 3.2	13.2 ± 10.3	27.0 ± 4.2	70.5 ± 6.3	36.7 ± 8.5	81.7 ± 38.1	91.7 ± 7.6	288.6 ± 20.4
UTIGSP(α=1e-2)-lin	6.6 ± 2.6	17.0 ± 9.9	27.4 ± 3.4	67.9 ± 8.7	18.3 ± 3.7	71.5 ± 37.0	77.6 ± 12.1	304.2 ± 22.4
UTIGSP(α=1e-3)-lin	4.4 ± 2.5	14.5 ± 10.8	27.8 ± 3.6	72.2 ± 6.0	16.1 ± 5.2	77.2 ± 38.4	72.2 ± 13.4	319.0 ± 16.9
UTIGSP(α=1e-5)-lin	6.3 ± 3.6	20.8 ± 14.8	28.7 ± 3.4	72.6 ± 5.7	15.7 ± 3.7	80.5 ± 19.0	71.5 ± 9.3	323.3 ± 17.0
UTIGSP(α=1e-7)-lin	5.7 ± 2.9	21.6 ± 15.3	29.9 ± 2.8	75.1 ± 5.4	15.7 ± 4.2	77.7 ± 25.7	73.0 ± 12.8	325.7 ± 17.8
UTIGSP(α=1e-9)-lin	5.3 ± 3.3	19.6 ± 15.3	30.2 ± 4.0	74.3 ± 8.6	17.1 ± 4.1	81.3 ± 28.2	76.2 ± 11.3	345.8 ± 17.9
UTIGSP(α=2e-1)	10.4 ± 4.0	12.4 ± 10.0	26.4 ± 4.4	69.3 ± 9.7	48.5 ± 7.8	93.9 ± 37.5	90.3 ± 15.5	306.8 ± 19.9
UTIGSP(α=1e-1)	8.1 ± 3.3	14.0 ± 7.6	26.9 ± 4.1	70.6 ± 6.8	35.6 ± 6.5	103.2 ± 28.8	86.5 ± 13.9	319.6 ± 28.1
UTIGSP(α=1e-2)	6.1 ± 4.1	16.6 ± 12.5	28.1 ± 4.8	68.4 ± 14.3	23.0 ± 5.7	107.1 ± 27.5	84.5 ± 8.9	327.3 ± 20.4
UTIGSP(α=1e-3)	6.4 ± 3.6	19.5 ± 14.5	31.0 ± 3.1	76.8 ± 4.3	20.6 ± 3.5	97.3 ± 20.8	81.1 ± 6.2	338.5 ± 10.8
UTIGSP(α=1e-5)	6.8 ± 3.5	21.1 ± 12.9	35.0 ± 2.2	80.6 ± 4.8	20.5 ± 4.2	95.8 ± 23.2	79.4 ± 8.8	338.1 ± 16.0
UTIGSP(α=1e-7)	6.2 ± 3.5	20.0 ± 11.5	32.5 ± 2.1	75.2 ± 9.9	20.0 ± 4.5	97.4 ± 22.2	78.8 ± 9.3	348.1 ± 12.2
UTIGSP(α=1e-9)	7.6 ± 3.8	22.3 ± 13.4	33.9 ± 2.0	78.6 ± 6.9	19.4 ± 3.9	94.3 ± 27.1	77.9 ± 7.5	342.3 ± 18.7
JCI-PC*	8.1 ± 2.6	26.7 ± 13.4	38.8 ± 1.9	80.8 ± 7.6	16.3 ± 3.5	89.8 ± 34.7	73.7 ± 7.7	335.8 ± 15.1
JCI-PC(α=2e-1)	10.5 ± 2.0	27.3 ± 14.3	39.2 ± 2.2	82.9 ± 6.6	23.4 ± 4.6	99.4 ± 34.8	73.8 ± 7.7	334.4 ± 18.4
JCI-PC(α=1e-1)	9.6 ± 2.0	27.8 ± 14.2	39.2 ± 2.2	82.9 ± 6.6	20.5 ± 3.9	100.0 ± 33.3	73.9 ± 7.7	336.2 ± 15.4
JCI-PC(α=1e-2)	8.2 ± 2.5	26.7 ± 13.4	39.4 ± 2.2	84.8 ± 4.6	16.8 ± 3.5	88.8 ± 36.2	74.0 ± 7.7	340.0 ± 14.3
JCI-PC(α=1e-3)	8.1 ± 2.6	26.7 ± 13.4	39.5 ± 2.1	84.9 ± 4.5	16.4 ± 3.6	90.9 ± 37.0	74.1 ± 7.8	340.1 ± 14.4
DCDI-G	6.6 ± 10.1	9.2 ± 9.4	8.5 ± 4.2	37.1 ± 15.3	16.5 ± 22.8	20.8 ± 10.5	35.4 ± 8.4	177.3 ± 38.8
DCDI-DSF	8.3 ± 11.4	12.1 ± 6.6	4.3 ± 2.6	28.6 ± 14.2	17.0 ± 13.5	52.6 ± 20.2	27.7 ± 10.0	126.9 ± 36.6

Table 4.30. Results for the nonlinear with non-additive noise data set with perfect intervention with unknown targets

Prologue to the Third Contribution

Article Details

Nonparametric Partial Disentanglement via Mechanism Sparsity: Sparse Actions, Interventions and Sparse Temporal Dependencies

by *Sébastien Lachapelle, Pau Rodríguez López, Yash Sharma, Katie Everett, Rémi Le Priol, Alexandre Lacoste and Simon Lacoste-Julien.* This work was submitted to the Journal of Machine Learning Research in 2024.

This work is a significantly extended version of the following articles (excluded from thesis):

Disentanglement via Mechanism Sparsity Regularization: A New Principle for Nonlinear ICA

By *Sébastien Lachapelle, Pau Rodríguez López, Yash Sharma, Katie Everett, Rémi Le Priol, Alexandre Lacoste and Simon Lacoste-Julien.* This work was published at the First Conference on Causal Learning and Reasoning (CLeaR 2022).

Partial Disentanglement via Mechanism Sparsity

By *Sébastien Lachapelle and Simon Lacoste-Julien.* This work was presented at The First Workshop on Causal Representation Learning at UAI 2022 with a **best paper award** and an **oral**.

Contributions of the Authors (JMLR version)

Sébastien Lachapelle developed the idea, the theory and proofs behind mechanism sparsity regularization for disentanglement, wrote the crux of the paper, and developed the regularized VAE-based method and performed most of the experiments. **Rémi Le Priol** provided valuable feedback on the clarity of the manuscript. **Simon Lacoste-Julien** helped with overall paper presentation, clarified the conceptual framework and the motivation and provided supervision.

Contributions of the Authors (CLeaR version)

Sébastien Lachapelle developed the idea, the theory and proofs behind mechanism sparsity regularization for disentanglement, wrote the first draft of the paper, and designed and implemented the regularized VAE-based method. **Pau Rodríguez López** ran all experiments appearing in the

paper, produced associated figures and ran experiments with image data that are still work in progress. **Yash Sharma** contributed to the research process, the experimental design in particular, implemented and ran experiments on image data that did not make it in the final version, and contributed to the writing and the literature review. **Katie Everett** implemented and ran experiments on image data that did not make it in the final version and contributed to the writing and figures. **Rémi Le Priol** reviewed the proofs of main theorems, simplified some arguments and the overall proof presentation and contributed to the writing and figures. **Alexandre Lacoste** produced image datasets that did not make it into the final version and provided supervision. **Simon Lacoste-Julien** helped with overall paper presentation, clarified the conceptual framework and the motivation and provided supervision.

Context and Limitations

The following contribution (of which a first version was presented at the Workshop on the Neglected Assumptions in Causal Inference at ICML 2021) was inspired by numerous talks given by Yoshua Bengio at Mila where he described the idea of learning a "causal graph in latent space" and how it could alleviate some of the limitations of current deep learning approaches [Bengio, 2019, Schölkopf et al., 2021]. However, there were no theoretical guarantees showing this vision was actually achievable despite the apparent lack of identifiability. This state of affairs did not prevent empirical investigations in this space including Goyal et al. [2021b], which proposed an architecture in which a sparsely connected latent dynamical system is learned and shown empirically to improve out-of-distribution generalization, Volodin [2021], which proposed a very similar sparsity principle to ours without identifiability guarantees, and Träuble et al. [2021], which highlighted the failure of disentanglement methods to recover correlated latent variables. On the theoretical side, the difficulty of identifiability in the nonlinear mixing case were already well understood [Hyvärinen and Pajunen, 1999, Locatello et al., 2020a] and seminal works in nonlinear ICA were proving the first identifiability results for nonlinear mixing [Hyvarinen and Morioka, 2016, 2017, Hyvärinen et al., 2019, Khemakhem et al., 2020a], which were crucial for the following contribution. The key novelty in our work is to leverage sparsity of the latent causal graphical model to disentangle with theoretical guarantees. More precisely the framework can leverage auxiliary variables with sparse effects (like actions), sparse interventions (also see [Lippe et al., 2022, 2023b]) and/or sparse temporal dependencies. The latter principle was exploited before in an extreme form where each latent $\boldsymbol{z}_i^t$ can only be influenced by itself, $\boldsymbol{z}_i^{t-1}$ [Tong et al., 1993, Hyvarinen and Morioka, 2017, Klindt et al., 2021]. To the best of our knowledge, our framework is the first to show theoretically that more permissive dependency structures between latent factors can be also leveraged to disentangle, even without an observed auxiliary variable (e.g. no actions nor interventions). Our work was also

the first to prove that interventions on latent variables can be used to disentangled, concurrently with Lippe et al. [2022].

We provide a more thorough review of the literature both predating and postdating the CLeaR 2022 version in Section 5.7, where we discuss many recent works in causal representation learning and disentanglement showing identifiability in various settings. We highlight the fact that Zheng et al. [2022] largely based their proof strategies and assumptions on the CLeaR 2022 version of our work, to show that sparsity in the mixing function $\boldsymbol{f}(\boldsymbol{z})$ can also be leveraged for disentanglement.

In the last few years, the community went from having no theoretical basis for causal representation learning to a plethora of new identifiability results showing causal representation learning is actually possible, at least in the infinite data regime. The next frontier for causal representation learning, and disentanglement more generally, is to go from success in theory to success in practical applications. Lopez et al. [2023] adapted our approach in order to apply it to gene expression data with perturbations and showed that (i) many different perturbations have the same effect on a given latent factors, suggesting these act on the same pathway, which is corroborated by previous studies, and (ii) sparse models with a disentangled representation can transfer more easily to held-out perturbations. Lei et al. [2023] similarly showed that the model we propose (with a sparse latent graph) can also adapt faster to sparse shift in the latent distribution of simple video data. These observations might be instantiations of the phenomenon identified by Bengio et al. [2020] and theoretically analyzed by Le Priol et al. [2021] which showed that causal models can sometimes adapt faster to sparse changes, i.e. with fewer samples. Further investigations are needed to explain this phenomenon in the causal representation learning setting.

At the moment, training models for causal representation learning is challenging since it inherits both difficulties of learning a causal graph (which is discrete) and training deep neural networks. There might also be an inherent trade-off between identifiability and ease of optimization: Overparameterization is known to make optimization easier but also making the model less identifiable. Nevertheless, Lippe et al. [2023a] have shown important progress when it comes to training these models, and showed convincing results on image data simulating robot control. More efforts are needed to demonstrate the applicability of causal representation learning methods to realistic settings.

Chapter 5

Nonparametric Partial Disentanglement via Mechanism Sparsity: Sparse Actions, Interventions and Sparse Temporal Dependencies

Abstract

This work introduces a novel principle for disentanglement we call *mechanism sparsity regularization*, which applies when the latent factors of interest depend sparsely on observed auxiliary variables and/or past latent factors. We propose a representation learning method that induces disentanglement by *simultaneously* learning the latent factors and the sparse causal graphical model that explains them. We develop a nonparametric identifiability theory that formalizes this principle and shows that the latent factors can be recovered by regularizing the learned causal graph to be sparse. More precisely, we show identifiablity up to a novel equivalence relation we call *consistency*, which allows some latent factors to remain entangled (hence the term *partial* disentanglement). To describe the structure of this entanglement, we introduce the notions of *entanglement graphs* and *graph preserving functions*. We further provide a graphical criterion which guarantees *complete* disentanglement, that is identifiability up to permutations and element-wise transformations. We demonstrate the scope of the mechanism sparsity principle as well as the assumptions it relies on with several worked out examples. For instance, the framework shows how one can leverage multi-node interventions with unknown targets on the latent factors to disentangle them. We further draw connections between our nonparametric results and the now popular exponential family assumption. Lastly, we propose an estimation procedure based on variational autoencoders and a sparsity constraint and demonstrate it on various synthetic datasets. This work is meant to be a significantly extended version of Lachapelle et al. [2022].

5.1. Introduction

It has been proposed that causal reasoning will be central to move modern machine learning algorithms beyond their current shortcomings, such as their lack of *robustness*, *transferability* and *interpretability* [Pearl, 2019, Schölkopf, 2019, Goyal and Bengio, 2021]. To achieve this, the field of *causal representation learning* (CRL) [Schölkopf et al., 2021] aims to learn representations of high-dimensional observations, such as images, that are suitable to perform causal reasoning such as predicting the effect of unseen interventions and answering counterfactual queries. A now popular formalism to do so is to assume that the observations $\boldsymbol{x} \in \mathbb{R}^{d_z}$ are sampled from a generative model of the form $\boldsymbol{x} = \boldsymbol{f}(\boldsymbol{z})$ where $\boldsymbol{z} \in \mathbb{R}^{d_z}$ is a random vector of *unobserved* and *semantically meaningful* variables, also called latent factors, distributed according to an unknown *causal graphical model* (CGM) [Pearl, 2009b, Peters et al., 2017] and transformed by a potentially highly nonlinear *decoder*, or *mixing function*, $\boldsymbol{f}$ [Kocaoglu et al., 2018, Volodin, 2021, Lachapelle et al., 2022, Lippe et al., 2023b, Brehmer et al., 2022, Ahuja et al., 2023, Buchholz et al., 2023, von Kügelgen et al., 2023, Zhang et al., 2023, Jiang and Aragam, 2023]. The goal is then to recover the latent factors $\boldsymbol{z}_i$ up to permutation and rescaling as well as the causal relationships explaining them. This is closely related to the problem of *disentanglement* [Bengio et al., 2013, Higgins et al., 2017, Locatello et al., 2020a] which also aims at extracting interpretable variables from high-dimensional observations, but without the emphasis on modelling their causal relations. Such problems are plagued by the difficult question of *identifiability*, which is of crucial importance to the classical settings of *causal discovery* [Pearl, 2009b, Peters et al., 2017], where $\boldsymbol{f}$ is assumed to be the identity, and *independent component analysis* (ICA) [Hyvärinen et al., 2001, 2023], where the causal graph over latents is assumed empty. In the former, one can only identify the Markov equivalence class of the causal graph (assuming faithfulness) thus leaving some edge orientations ambiguous [Pearl, 2009b], while in the latter, identifiability of the ground-truth latent factors is impossible when assuming a general nonlinear $\boldsymbol{f}$, [Hyvärinen and Pajunen, 1999]. The general CRL problem inherits the difficulties from both of these settings, which makes identifiability especially challenging. Various strategies to improve identifiability have been contributed to the literature such as assuming access to *interventional data* in which latent factors are targeted by interventions [Lachapelle et al., 2022, Lippe et al., 2022, 2023b, Ahuja et al., 2023], or access to an *auxiliary variable* $\boldsymbol{a}$ that renders the factors $\boldsymbol{z}_i$ mutually independent when conditioned on [Hyvärinen et al., 2019, Khemakhem et al., 2020a,b]. A valid auxiliary variable $\boldsymbol{a}$ must be observed and could correspond, for instance, to a time or an environment index, an action in an interactive environment, or even a previous observation if the data has temporal structure. See Section 5.7 for a more extensive review of existing approaches for latent variable identification.

The present paper introduces[1] *mechanism sparsity regularization* as a new principle for latent variable identification. We show that if (i) an auxiliary variable $\boldsymbol{a}$ is observed and affects the latent variables *sparsely* and/or (ii) the latent variables present *sparse* temporal dependencies, then the latent variables can be recovered by learning a graphical model for $\boldsymbol{z}$ and $\boldsymbol{a}$ and regularizing it to be sparse (Theorems 5.1, 5.2, 5.3 & 5.5). More specifically, we consider models of the form $\boldsymbol{x}^t = \boldsymbol{f}(\boldsymbol{z}^t) + \boldsymbol{n}^t$, where $\boldsymbol{n}^t$ is independent noise (Assumption 5.1) and the latent factors $\boldsymbol{z}_i^t$ are mutually independent given the past factors and auxiliary variables, i.e. $p(\boldsymbol{z}^t \mid \boldsymbol{z}^{<t}, \boldsymbol{a}^{<t}) = \prod_{i=1}^{d_z} p(\boldsymbol{z}_i^t \mid \boldsymbol{z}^{<t}, \boldsymbol{a}^{<t})$ (Assumption 5.2). Crucially, we leverage the assumption that these mechanisms are sparse in the sense that $p(\boldsymbol{z}^t \mid \boldsymbol{z}^{<t}, \boldsymbol{a}^{<t})$ factorizes according to a sparse causal graph $\boldsymbol{G}$ (Assumption 5.3). Interestingly, if $\boldsymbol{a}$ corresponds to an intervention index, our framework explains how interventions targeting unknown subsets of latent factors can identify them (Section 5.3.3.1). We emphasize that the settings where the data has no temporal dependencies or no auxiliary variable $\boldsymbol{a}$ are special cases of our framework. Our identifiability results are summarized in Table 5.1.

This work is meant to be an extended version of Lachapelle et al. [2022] in which we generalize along two main axes: First, we relax the *exponential family* assumption by providing a fully *nonparameteric* treatment. Secondly, our results drop the graphical criterion of Lachapelle et al. [2022] and, thus, allow for *arbitrary* latent causal graphs. As a consequence of this relaxation, instead of guaranteeing identifiability up to permutation and element-wise transformation, we guarantee identifiability up to what we call $\boldsymbol{a}$*-consistency* or $\boldsymbol{z}$*-consistency* (Definitions 5.13 & 5.14), which might allow certain latent variables to remain entangled. Our results thus have the following flavor: Given a specific ground-truth causal graph $\boldsymbol{G}$ over $\boldsymbol{z}$ and $\boldsymbol{a}$, we describe precisely the structure of the entanglement between latent factors via what we call an *entanglement graph* (Definition 5.3) and *graph preserving functions* (Definition 5.12). See Figure 5.3 for examples. Interestingly, the stronger identifiability up to permutation and element-wise transformation arises as a simple consequence of our theory when the graphical criterion of Lachapelle et al. [2022] is assumed to hold. In addition to these two main axes of generalization, we provide extensive examples illustrating the scope of our framework, our assumptions and the consequences of our results (See Table 5.2 for a list). When it comes to the learning algorithm, we replaced the sparsity *penalty* by a sparsity *constraint*, which improves the learning dynamics and is more interpretable, which results in easier hyperparameter tuning.

The hypothesis that *high-level concepts can be described by a sparse dependency graph* has been described and leveraged for out-of-distribution generalization by Bengio [2019] and Goyal et al. [2021b], which were early sources of inspiration for this work. To the best of our knowledge,

[1] A shorter version of this work originally appeared in Lachapelle et al. [2022].

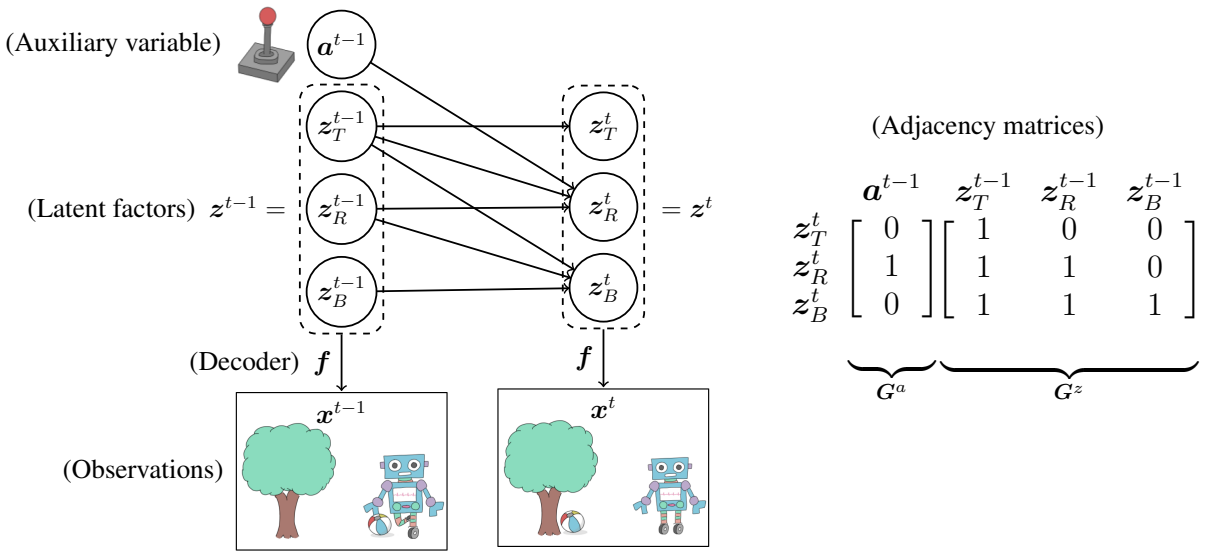


Figure 5.1. A minimal motivating example. The latent factors z_T^t, z_R^t and z_B^t represent the x-positions of the tree, the robot and the ball at time t, respectively. Only the image of the scene x^t and the action a^{t-1} are observed. See end of Section 5.2.1 for details.

			Sparse $\hat{G}^a$		Sparse $\hat{G}^z$		
	Parametric assumption	**Continuous a**	**Discrete a (interventions)**	**Temporal dependencies**	**Sufficient influence**	**Identifiable up to**	**Examples**
Thm. 5.1	None	Required	–	Optional	Ass. 5.6	Def. 5.13	5.2, 5.3, 5.4, 5.8, 5.9
Thm. 5.2	None	–	Required	Optional	Ass. 5.7	Def. 5.13	5.2, 5.3, 5.4, 5.10, 5.11, 5.12
Thm. 5.3	None	Optional	Optional	Required	Ass. 5.8	Def. 5.14	5.5, 5.6, 5.7, 5.13
Thm. 5.4	Exp. fam.	Optional	Optional	Optional	–	Def. 5.17	5.14
Thm. 5.5	Exp. fam.	Optional	Optional	Required	Ass. 5.11	Def. 5.14, 5.17	5.15

Table 5.1. Summary of our identifiability results.

our theory is the first to show formally that this inductive bias can sometimes be enough to recover the latent factors.

Figure 5.1 shows a minimal motivating example in which our approach could be used to extract the high-level variables (such as the x-position of the three objects) and learn their dynamics (how the objects move and affect one another) from a time series of images and agent actions, (x^t, a^t). Theorems 5.1, 5.2, 5.3 & 5.5 show how the sparse dependencies between the objects and the action can be leveraged to estimate the latent variables as well as the graph describing their dynamics. The learned CGM could be used subsequently to simulate interventions on semantic variables [Pearl, 2009b, Peters et al., 2017], such as changing the torque of the robot or the weight of the ball. Moreover, disentanglement could be useful to interpret what caused the actions of an agent [Pearl, 2019]. Following Lachapelle et al. [2022], empirical works demonstrated that disentangled representations with sparse mechanisms can adapt to unseen interventions faster in the context of single-cell biology [Lopez et al., 2023] and synthetic video data [Lei et al., 2023].

Summary of our contributions:

(1) We introduce[1] a new principle for disentanglement based on *mechanism sparsity regularization* motivated by rigorous and novel *identifiability guarantees* (Theorems 5.1, 5.2, 5.3 & 5.5).

(2) We extend Lachapelle et al. [2022] by providing a fully *nonparameteric* treatment and allowing for *arbitrary latent graphs*. Given a latent ground-truth graph, our theory predicts the structure of the entanglement between variables, which we formalize with *entanglement graphs* (Definition 5.3), *graph preserving maps* (Definition 5.12) and novel *equivalence relations* (Definitions 5.13 & 5.14).

(3) We provide several examples to illustrate the generality of our results and get a better understanding of their various assumptions and consequences (summarized in Table 5.2). For instance, we show how multi-node interventions with unknown-targets can yield disentanglement, both with and without temporal dependencies (Examples 5.11 & 5.12).

(4) We introduce an evaluation metric denoted by R_{con} which quantifies how close two representations are to being $\boldsymbol{a}$-consistent or $\boldsymbol{z}$-consistent (Section 5.6).

(5) We implement a learning approach based on variational autoencoders (VAEs) [Kingma and Welling, 2014] which learns the mixing function $\boldsymbol{f}$, the transition distribution $p(\boldsymbol{z}^t \mid \boldsymbol{z}^{<t}, \boldsymbol{a}^{<t})$ and the causal graph $\boldsymbol{G}$. The latter is learned using binary masks and regularized for sparsity via a *constraint* as opposed to a penalty as in Lachapelle et al. [2022].

(6) We perform experiments on synthetic datasets in order to validate the prediction of our theory.

Overview. Section 5.2 introduces the model (Section 5.2.1), entanglement maps and graphs (Section 5.2.2), the notion identifiability (Section 5.2.3), equivalence up to diffeomorphism (Section 5.2.4) and disentanglement formally (Section 5.2.5). Section 5.3 provides mathematical intuition for why mechanism sparsity yields disentanglement (Section 5.3.1); introduces the machinery of graph preserving maps (Section 5.3.2) which are key to establish identifiability up to $\boldsymbol{a}$-consistency (Section 5.3.3) and $\boldsymbol{z}$-consistency (Section 5.3.4), i.e. *partial* disentanglement. Section 5.3 also discusses the relationship to interventions (Section 5.3.3.1), provides a graphical criterion guaranteeing *complete* disentanglement (Section 5.3.6), and introduces and discusses extensively the *sufficient influence assumptions* on which these results critically rely (Sections 5.3.7 & 5.3.8). Section 5.4 draws connections between our *nonparameteric* theory and the *exponential family* assumption sometimes used in the literature. Section 5.5 presents the VAE-based learning algorithm with sparsity constraint. Section 5.6 introduces our novel R_{con} metric. Section 5.7 reviews the literature on identifiability in representation learning. Section 5.8 presents the empirical results.
Notation. Scalars are denoted in lower-case and vectors in lower-case bold, e.g. $x \in \mathbb{R}$ and $\boldsymbol{x} \in \mathbb{R}^n$. Note that these will sometimes denote a random variables, depending on context. We maintain an

Examples	Type of disentanglement	Auxiliary variable	Time dependencies
5.2	Complete	Yes (single target)	Optional
5.3	Partial	Yes (single target)	Optional
5.4	Complete	Yes (multi-target)	Optional
5.5	Complete	Optional	Yes (independent factors)
5.6	Complete	Optional	Yes (dependent factors)
5.7	Partial	Optional	Yes (dependent factors)
5.8	Partial	Yes (single-target continuous)	Yes
5.9	Complete	Yes (multi-target continuous)	No
5.10	Complete	Yes (single-target interventions)	No
5.11	Complete	Yes (multi-target interventions)	Yes
5.12	Complete	Yes (grouped multi-target interventions)	No
5.13	Complete	No	Yes (non-Markovian)
5.15	Complete	No	Yes (Markovian)

Table 5.2. List of examples illustrating the scope of our theory, its assumptions and its consequences.

analogous notation for scalar-valued and vector-valued functions, e.g. f and $\boldsymbol{f}$. The ith coordinate of the vector $\boldsymbol{x}$ is denoted by $\boldsymbol{x}_i$. The set containing the first n integers excluding 0 is denoted by $[n]$. Given a subset of indices $S \subseteq [n]$, $\boldsymbol{x}_S$ denotes the subvector consisting of entries $\boldsymbol{x}_i$ for $i \in S$. Given a sequence of T random vectors $(\boldsymbol{x}^1, \ldots, \boldsymbol{x}^T)$, the subsequence consisting of the first t elements is denoted by $\boldsymbol{x}^{\leq t} := (\boldsymbol{x}^1, ..., \boldsymbol{x}^t)$, and analogously for $\boldsymbol{x}^{<t}$. We will sometimes combine these notations to get $\boldsymbol{x}_S^{\leq t} := (\boldsymbol{x}_S^1, \ldots, \boldsymbol{x}_S^t)$. Given a function $\boldsymbol{f} : \mathbb{R}^n \to \mathbb{R}^m$, its Jacobian matrix evaluated at $\boldsymbol{x} \in \mathbb{R}^n$ is denoted by $D\boldsymbol{f}(\boldsymbol{x}) \in \mathbb{R}^{m \times n}$. See Table 5.5 in appendix for more.

5.2. Problem setting, entanglement graphs & disentanglement

In this section, we introduce the latent variable model under consideration (Section 5.2.1), entanglement graphs (Section 5.2.2), identifiability and observational equivalence (Section 5.2.3), equivalence up to diffeomorphism (Section 5.2.4) as well as permutation equivalence (Section 5.2.5).

5.2.1. An identifiable latent causal model

We now specify the setting under consideration. Assume we observe the realization of a sequence of d_x-dimensional random vectors $\{\boldsymbol{x}^t\}_{t=1}^T$ and a sequence of d_a-dimensional auxiliary vectors $\{\boldsymbol{a}^t\}_{t=0}^{T-1}$. The coordinates of $\boldsymbol{a}^t$ are either discrete or continuous and can potentially represent, for example, an action taken by an agent or the index of an intervention or environment (see Section 5.3.3.1). The observations $\{\boldsymbol{x}^t\}$ are assumed to be explained by a sequence of hidden d_z-dimensional continuous random vectors $\{\boldsymbol{z}^t\}_{t=1}^T$ via a ground-truth decoder function $\boldsymbol{f}$.

Assumption 5.1 (Observation model)**.** *For all $t \in [T]$, the observations $\boldsymbol{x}^t$ are given by*

$$\boldsymbol{x}^t = \boldsymbol{f}(\boldsymbol{z}^t) + \boldsymbol{n}^t \,, \tag{5.1}$$

where $\boldsymbol{n}^t \sim \mathcal{N}(0, \sigma^2 \boldsymbol{I})$ *are mutually independent across time and independent of all* $\boldsymbol{z}^t$ *and* $\boldsymbol{a}^t$ *with* $\sigma^2 \geq 0$*. Moreover,* $d_z \leq d_x$ *and* $\boldsymbol{f} : \mathbb{R}^{d_z} \to \mathbb{R}^{d_x}$ *is a diffeomorphism onto its image*[2]*. Lastly, assume that* $\boldsymbol{f}(\mathbb{R}^{d_z})$ *is closed in* $\mathbb{R}^{d_x}$*.*

Importantly, we suppose that each factor $\boldsymbol{z}_i^t$ contains interpretable information about the observation, e.g. for high-dimensional images, the coordinates $\boldsymbol{z}_i^t$ might be the position of an object, its color, or its orientation in space. This idea that there exists a *ground-truth decoder* $\boldsymbol{f}$ that captures the relationship between the so-called "natural factors of variations" and the observations $\boldsymbol{x}$ is of capital importance, since it is the very basis for a mathematical definition of disentanglement (Definition 5.7). Appendix D.1 discusses the implications of the diffeomorphism assumption (see also Mansouri et al. [2022]). We denote $\boldsymbol{z}^{\leq t} := [\boldsymbol{z}^1 \ \cdots \ \boldsymbol{z}^t] \in \mathbb{R}^{d_z \times t}$ and analogously for $\boldsymbol{z}^{<t}$ and other random vectors.

In a similar spirit to previous works on nonlinear ICA [Hyvärinen et al., 2019, Khemakhem et al., 2020a], we assume the latent factors $\boldsymbol{z}_i^t$ are conditionally independent given the past.

Assumption 5.2 (Conditionally independent latent factors)**.** *The latent factors* $\boldsymbol{z}_i^t$ *are conditionally mutually independent given* $\boldsymbol{z}^{<t}$ *and* $\boldsymbol{a}^{<t}$*:*

$$p(\boldsymbol{z}^t \mid \boldsymbol{z}^{<t}, \boldsymbol{a}^{<t}) = \prod_{i=1}^{d_z} p(\boldsymbol{z}_i^t \mid \boldsymbol{z}^{<t}, \boldsymbol{a}^{<t}) \,, \tag{5.2}$$

where $p(\boldsymbol{z}^t \mid \boldsymbol{z}^{<t}, \boldsymbol{a}^{<t})$ *is a density function w.r.t. the Lebesgue measure on* $\mathbb{R}^{d_z}$*. We assume that the support of* $p(\boldsymbol{z}_i^t \mid \boldsymbol{z}^{<t}, \boldsymbol{a}^{<t})$ *is* $\mathbb{R}$ *for all* $\boldsymbol{z}^{<t}$ *and* $\boldsymbol{a}^{<t}$*. The support of* $p(\boldsymbol{z}^t \mid \boldsymbol{z}^{<t}, \boldsymbol{a}^{<t})$ *is thus given by* $\mathbb{R}^{d_z}$*.*

We will refer to the l.h.s. of (5.2) as the *transition model* and to each factor $p(\boldsymbol{z}_i^t \mid \boldsymbol{z}^{<t}, \boldsymbol{a}^{<t})$ as *mechanisms*. Notice that we do not assume the system is *Markovian*, i.e. the distribution over future states can depend on the whole history of latents and auxiliary variables $(\boldsymbol{z}^{<t}, \boldsymbol{a}^{<t})$. In addition, this model can represent *non-homogeneous* processes by taking the auxiliary variable $\boldsymbol{a}$ to be a time index [Hyvärinen et al., 2019].

We are going to describe the dependency structure of the latent and auxiliary variables through time via a *probabilistic directed graphical model* composed of two bipartite graphs, $\boldsymbol{G}^z \in \{0,1\}^{d_z \times d_z}$, which relates $\boldsymbol{z}^{<t}$ to $\boldsymbol{z}^t$, and $\boldsymbol{G}^a \in \{0,1\}^{d_z \times d_a}$, which relates $\boldsymbol{a}^{<t}$ to $\boldsymbol{z}^t$. A directed edge points from $\boldsymbol{z}_j^{<t}$ to $\boldsymbol{z}_i^t$ if and only if $\boldsymbol{G}^z_{i,j} = 1$. Analogously, a directed edge points from $\boldsymbol{a}_\ell^{<t}$ to $\boldsymbol{z}_i^t$, if and only if $\boldsymbol{G}^a_{i,\ell} = 1$. Figure 5.1 shows an example of such graphs together with

[2]A *diffeomorphism* is a C^1 bijection with a C^1 inverse. Generally, given a map $\boldsymbol{h} : A \to \mathbb{R}^m$ where $A \subseteq \mathbb{R}^n$, saying $\boldsymbol{h}$ is C^k is typically only well defined if A is an open set of $\mathbb{R}^n$. Throughout, if $A \subseteq \mathbb{R}^n$ is arbitrary (not necessarily open), we say $\boldsymbol{h}$ is C^k if there exists a C^k map $\tilde{\boldsymbol{h}} : U \to \mathbb{R}^m$ defined on an open set U of $\mathbb{R}^n$ containing A such that $\boldsymbol{h} = \tilde{\boldsymbol{h}}$ on A. Note that it is then meaningful for $\boldsymbol{f}^{-1} : \boldsymbol{f}(\mathbb{R}^{d_z}) \to \mathbb{R}^{d_z}$ to be C^1 even when $\boldsymbol{f}(\mathbb{R}^{d_z})$ is not open in $\mathbb{R}^{d_x}$. Moreover, it can be shown that $\boldsymbol{f} : \mathbb{R}^{d_z} \to \mathbb{R}^{d_x}$ is a diffeomorphism onto its image if $\boldsymbol{f}$ is an homeomorphism onto its image, i.e. continuous in both directions, and has a full rank Jacobian everywhere on its domain [Munkres, 1991, Sec. 23 & Thm. 24.1].

its adjacency matrix $\boldsymbol{G} := [\boldsymbol{G}^z, \boldsymbol{G}^a]$. The following assumption specifies the relationship between these graphs and the transition model.

Assumption 5.3 (Transition model p is Markov w.r.t. $\boldsymbol{G}$)**.** *For all mechanism $i \in [d_z]$,*

$$p(\boldsymbol{z}_i^t \mid \boldsymbol{z}^{<t}, \boldsymbol{a}^{<t}) = p(\boldsymbol{z}_i^t \mid \boldsymbol{z}^{<t}_{\mathbf{Pa}_i^z}, \boldsymbol{a}^{<t}_{\mathbf{Pa}_i^a}) \,, \tag{5.3}$$

where $\mathbf{Pa}_i^z \subseteq [d_z]$ and $\mathbf{Pa}_i^a \subseteq [d_a]$ are the sets of parents of $\boldsymbol{z}_i^t$ in $\boldsymbol{G}^z$ and $\boldsymbol{G}^a$, respectively.

The graph $\boldsymbol{G}$ thus encodes a set of conditional independence statements about the latent and auxiliary variables. We will say that *mechanisms are sparse* when the graphs $\boldsymbol{G}^a$ and $\boldsymbol{G}^z$ are sparse.

This model has three components that need to be learned: (i) the decoder function $\boldsymbol{f}$, (ii) the transition model over latent variables p, and (iii) the dependency graph $\boldsymbol{G}$. We collect all these components into $\boldsymbol{\theta} := (\boldsymbol{f}, p, \boldsymbol{G})$. Everything else in the model, i.e. d_z and σ^2, is assumed to be known. We assume that σ^2 is known here mainly for simplicity, since, when it is not, it can be identified as shown by Lachapelle et al. [2022, Appendix A.4.1], as long as $d_x > d_z$.

Notice how we have not specified any model for the auxiliary variable $\boldsymbol{a}^t$. We do not intend to do so in this work, as we are solely interested in modelling the *conditional* distribution of $\boldsymbol{x}^{\leq T}$ and $\boldsymbol{z}^{\leq T}$ given $\boldsymbol{a}^{<T}$. We denote by $\mathcal{A} \subseteq \mathbb{R}^{d_a}$ the set of possible values for the auxiliary variable $\boldsymbol{a}^t$. We thus have that, for all values of $\boldsymbol{a}^{<T} \in \mathcal{A}^T$, our model induces a conditional distribution

$$p(\boldsymbol{x}^{\leq T} \mid \boldsymbol{a}^{<T}) = \int \prod_{t=1}^{T} p(\boldsymbol{x}^t \mid \boldsymbol{z}^t) p(\boldsymbol{z}^t \mid \boldsymbol{z}^{<t}, \boldsymbol{a}^{<t}) d\boldsymbol{z}^{\leq T} \,, \tag{5.4}$$

where $p(\boldsymbol{x}^t \mid \boldsymbol{z}^t) = \mathcal{N}(\boldsymbol{x}^t; \boldsymbol{f}(\boldsymbol{z}^t), \sigma^2 \boldsymbol{I})$. We note that if $\sigma^2 = 0$, the conditional distribution of $\boldsymbol{x}^t$ given $\boldsymbol{z}^t$ is a Dirac centered at $\boldsymbol{f}(\boldsymbol{z}^t)$ and thus has no density w.r.t. to the Lebesgue measure. Even if, in that case, the above integral makes no sense, the conditional distribution of $\boldsymbol{x}^{\leq T}$ given $\boldsymbol{a}^{<T}$ is still well-defined and all the results of this work still hold since none of the proofs requires $\sigma^2 > 0$.

A motivating example. Figure 5.1 represents a minimal example where our theory applies. The environment consists of three objects: a tree, a robot and a ball with x-positions $\boldsymbol{z}_T^t$, $\boldsymbol{z}_R^t$ and $\boldsymbol{z}_B^t$, respectively. Together, they form the vector $\boldsymbol{z}^t$ of high-level latent variables, i.e. $\boldsymbol{z}^t = (\boldsymbol{z}_T^t, \boldsymbol{z}_R^t, \boldsymbol{z}_B^t)$. A remote controls the direction in which the wheels of the robot turn. The vector $\boldsymbol{a}^t$ records these actions, which might be taken by a human or an artificial agent trained to accomplish some goal. The only observations are the actions $\boldsymbol{a}^t$ and the images $\boldsymbol{x}^t$ representing the scene which is given by $\boldsymbol{x}^t = \boldsymbol{f}(\boldsymbol{z}^t) + \boldsymbol{n}^t$. The dynamics of the environment is governed by the transition model p, which, e.g., could be given by a Gaussian model of the form $p(\boldsymbol{z}_i^t \mid \boldsymbol{z}^{<t}, \boldsymbol{a}^{<t}) = \mathcal{N}(\boldsymbol{z}_i^t; \mu_i(\boldsymbol{z}^{t-1}, \boldsymbol{a}^{t-1}), \sigma_z^2)$. Plausible connectivity graphs $\boldsymbol{G}^z$ and $\boldsymbol{G}^a$ are given in Figure 5.1 showing how the latent factors are related, and how the controller affects them. For every object, its position at time step t depends on its position at $t-1$. The position of the tree, $\boldsymbol{z}_T^t$, is not affected by anything, since neither the robot nor the ball can change its position. The robot, $\boldsymbol{z}_R^t$, changes its position based on both the action, $\boldsymbol{a}^{t-1}$ and the position of the tree, $\boldsymbol{z}_T^{t-1}$ (in case of collision). The ball position, $\boldsymbol{z}_B^t$, is affected by

both the robot, which can kick it around by running into it, and the tree, on which it can bounce. The key observations here are that (i) the different objects interact *sparsely* with one another and (ii) the action $\boldsymbol{a}^t$ affects very few objects (in this case, only one). The theorems of Section 5.3 show how one can leverage this sparsity for disentanglement.

5.2.2. Entanglement maps & entanglement graphs

In this section, we define *entanglement maps*, which describes the functional relationship between the learned and ground-truth representations, and *entanglement graphs*, which describes their entanglement structure.

Definition 5.1 (Entanglement maps)**.** *Let $\boldsymbol{f}$ and $\tilde{\boldsymbol{f}}$ be two diffeomorphisms from $\mathbb{R}^{d_z}$ to their images such that $\boldsymbol{f}(\mathbb{R}^{d_z}) = \tilde{\boldsymbol{f}}(\mathbb{R}^{d_z})$. The **entanglement map** of the pair $(\boldsymbol{f}, \tilde{\boldsymbol{f}})$ is given by*

$$\boldsymbol{v} := \boldsymbol{f}^{-1} \circ \tilde{\boldsymbol{f}} \,. \tag{5.5}$$

This map will be crucial throughout this work, especially to define disentanglement. Intuitively, the entanglement map for a pair of decoders $(\boldsymbol{f}, \tilde{\boldsymbol{f}})$ translates the representation of one model to that of the other. In general, the entanglement maps of $(\boldsymbol{f}, \tilde{\boldsymbol{f}})$ and $(\tilde{\boldsymbol{f}}, \boldsymbol{f})$ are different.

We now define the *dependency graph* of some function $\boldsymbol{h}$ to be such that each edge indicates that some input i influences some output j:

Definition 5.2 (Functional dependency graph)**.** *Let $\boldsymbol{h}$ be a function from $\mathbb{R}^n$ to $\mathbb{R}^m$. The **dependency graph** of $\boldsymbol{h}$ is a bipartite directed graph from $[n]$ to $[m]$ with adjacency matrix $\boldsymbol{H} \in \{0,1\}^{m \times n}$ such that*

$$\boldsymbol{H}_{i,j} = 0 \iff \text{There is a function } \bar{\boldsymbol{h}} \text{ such that, for all } \boldsymbol{a} \in \mathbb{R}^n,\ \boldsymbol{h}_i(\boldsymbol{a}) = \bar{\boldsymbol{h}}_i(\boldsymbol{a}_{-j})\,, \tag{5.6}$$

where $\boldsymbol{a}_{-j}$ is $\boldsymbol{a}$ with its jth coordinate removed.

Example 5.1 (Dependency graph of a linear map)**.** *Let $\boldsymbol{h}(\boldsymbol{z}) := \boldsymbol{W}\boldsymbol{z}$ where $\boldsymbol{W} \in \mathbb{R}^{m \times n}$ and let $\boldsymbol{H}$ be the dependency graph of $\boldsymbol{h}$. Then, $\boldsymbol{H}_{i,j} = 0 \iff \boldsymbol{W}_{i,j} = 0$.*

We will be particularly interested in the dependency graph of the entanglement map $\boldsymbol{v} := \boldsymbol{f}^{-1} \circ \hat{\boldsymbol{f}}$, denoted by $\boldsymbol{V}$.

Definition 5.3 (Entanglement graphs)**.** *Let $\boldsymbol{f}$ and $\tilde{\boldsymbol{f}}$ be two diffeomorphisms from $\mathbb{R}^{d_z}$ to their images such that $\boldsymbol{f}(\mathbb{R}^{d_z}) = \tilde{\boldsymbol{f}}(\mathbb{R}^{d_z})$. The **entanglement graph** of the pair $(\boldsymbol{f}, \tilde{\boldsymbol{f}})$ is the dependency graph (Definition 5.2) of their entanglement map $\boldsymbol{v} := \boldsymbol{f}^{-1} \circ \tilde{\boldsymbol{f}}$, which we denote $\boldsymbol{V} \in \{0,1\}^{d_z \times d_z}$.*

We now relate the dependency graph of a function to the zeros of its Jacobian matrix. A proof can be found in Appendix A.2.

Proposition 5.1 (Linking dependency graph and Jacobian)**.** *Let $\boldsymbol{h}$ be a C^1 function, i.e. continuously differentiable, from $\mathbb{R}^n$ to $\mathbb{R}^m$ and let $\boldsymbol{H}$ be its dependency graph (Definition 5.2). Then,*

$$\boldsymbol{H}_{i,j} = 0 \iff \text{For all } \boldsymbol{a} \in \mathbb{R}^n \,, D\boldsymbol{h}(\boldsymbol{a})_{i,j} = 0 \,. \tag{5.7}$$

The equivalence (5.7) can be seen as an equivalent definition of dependency graph for differentiable functions.

5.2.3. Identifiability and observational equivalence

To analyse formally whether a specific algorithm is expected to yield a disentangled representation, we will rely on the notion of *identifiability*. Before defining what we mean by identifiability, we will need the notion of *observationally equivalent* models. Two models are observationally equivalent, if both models represent the same distribution over observations. The following formalizes this definition.

Definition 5.4 (Observational equivalence)**.** *We say two models $\boldsymbol{\theta} := (\boldsymbol{f}, p, \boldsymbol{G})$ and $\tilde{\boldsymbol{\theta}} := (\tilde{\boldsymbol{f}}, \tilde{p}, \tilde{\boldsymbol{G}})$ satisfying Assumption 5.1 are **observationally equivalent**, denoted $\boldsymbol{\theta} \sim_{\text{obs}} \tilde{\boldsymbol{\theta}}$, if and only if, for all $\boldsymbol{a}^{<T} \in \mathcal{A}^T$ and all $\boldsymbol{x}^{\leq T} \in \mathbb{R}^{d_x \times T}$,*

$$p(\boldsymbol{x}^{\leq T} \mid \boldsymbol{a}^{<T}) = \tilde{p}(\boldsymbol{x}^{\leq T} \mid \boldsymbol{a}^{<T}) \,. \tag{5.8}$$

Formally, we say a parameter $\boldsymbol{\theta}$ is **identifiable up to some equivalence relation** $\sim$, when

$$\boldsymbol{\theta} \sim_{\text{obs}} \tilde{\boldsymbol{\theta}} \implies \boldsymbol{\theta} \sim \tilde{\boldsymbol{\theta}} \,. \tag{5.9}$$

This work is mainly concerned with proving statements of the above form by making assumptions both on $\boldsymbol{\theta}$ and $\hat{\boldsymbol{\theta}}$. The stronger the assumptions on $\boldsymbol{\theta}$ and $\hat{\boldsymbol{\theta}}$ are, the stronger the equivalence relation $\sim$ will be. The following sections present two equivalence relations over models, namely, $\sim_{\text{diff}}$ and $\sim_{\text{perm}}$. We note that the equivalence relation $\sim_{\text{perm}}$ will help us formalize disentanglement.

Practically speaking, observational equivalence between the learned model $\hat{\boldsymbol{\theta}}$ and the ground-truth model $\boldsymbol{\theta}$ can be achieved via maximum likelihood estimation in the infinite data regime. Thus, identifiability results of the form of (5.9) guarantee that if the learned model is perfectly fitted on the data (assumed infinite), its parameter $\hat{\boldsymbol{\theta}}$ is $\sim$-equivalent to the that of the ground-truth model, $\boldsymbol{\theta}$.

5.2.4. Equivalence up to diffeomorphism

We start by defining *equivalence up to diffeomorphism*. This equivalence relation is important since we will show later on that it is actually the same as observational equivalence and will thus be our first step in all our identifiability results. In what follows, we overload the notation and write $\boldsymbol{v}(\boldsymbol{z}^{<t}) := [\boldsymbol{v}(\boldsymbol{z}^1), \ldots, \boldsymbol{v}(\boldsymbol{z}^{t-1})]$, and similarly for other functions.

Definition 5.5 (Equivalence up to diffeomorphism)**.** *We say two models* $\boldsymbol{\theta} := (\boldsymbol{f}, p, \boldsymbol{G})$ *and* $\tilde{\boldsymbol{\theta}} := (\tilde{\boldsymbol{f}}, \tilde{p}, \tilde{\boldsymbol{G}})$ *satisfying Assumption 5.1 are* ***equivalent up to diffeomorphism****, denoted* $\boldsymbol{\theta} \sim_{\text{diff}} \tilde{\boldsymbol{\theta}}$*, if and only if* $\boldsymbol{f}(\mathbb{R}^{d_z}) = \tilde{\boldsymbol{f}}(\mathbb{R}^{d_z})$ *and, for all* $t \in [T]$*, all* $\boldsymbol{a}^{<t} \in \mathcal{A}^t$ *and all* $\boldsymbol{z}^{\leq t} \in \mathbb{R}^{d_z \times t}$*,*

$$\tilde{p}(\boldsymbol{z}^t \mid \boldsymbol{z}^{<t}, \boldsymbol{a}^{<t}) = p(\boldsymbol{v}(\boldsymbol{z}^t) \mid \boldsymbol{v}(\boldsymbol{z}^{<t}), \boldsymbol{a}^{<t}) |\det D\boldsymbol{v}(\boldsymbol{z}^t)|\,, \tag{5.10}$$

where $\boldsymbol{v} := \boldsymbol{f}^{-1} \circ \tilde{\boldsymbol{f}}$ *(entanglement map) is a diffeomorphism and* $D\boldsymbol{v}$ *denotes its Jacobian matrix.*

The fact that the relation $\sim_{\text{diff}}$ is indeed an *equivalence* comes from the fact that the set of diffeomorphisms from a set to itself forms a group under composition.

To better understand the above definition, let $\boldsymbol{z}^t := \boldsymbol{g}(\boldsymbol{z}^{<t}, \boldsymbol{a}^{<t}; \epsilon^t)$ and $\tilde{\boldsymbol{z}}^t := \tilde{\boldsymbol{g}}(\tilde{\boldsymbol{z}}^{<t}, \boldsymbol{a}^{<t}; \tilde{\epsilon}^t)$ where ϵ^t and $\tilde{\epsilon}^t$ are noise variables and $\boldsymbol{g}$ and $\tilde{\boldsymbol{g}}$ are functions such that the random variables $\boldsymbol{z}^t$ and $\tilde{\boldsymbol{z}}^t$ have conditional densities given by $p(\boldsymbol{z}^t \mid \boldsymbol{z}^{<t}, \boldsymbol{a}^{<t})$ and $\tilde{p}(\tilde{\boldsymbol{z}}^t \mid \tilde{\boldsymbol{z}}^{<t}, \boldsymbol{a}^{<t})$, respectively. Using the change-of-variable formula for densities, one can rewrite (5.10) as

$$\tilde{\boldsymbol{g}}(\tilde{\boldsymbol{z}}^{<t}, \boldsymbol{a}^{<t}; \tilde{\epsilon}^t) \overset{d}{=} \boldsymbol{v}^{-1} \circ \boldsymbol{g}(\boldsymbol{v}(\tilde{\boldsymbol{z}}^{<t}), \boldsymbol{a}^{<t}; \epsilon^t)\,, \tag{5.11}$$

where "$\overset{d}{=}$" denotes equality in distribution. This equation has a nice interpretation: applying the latent transition model $\tilde{\boldsymbol{\theta}}$ to go from $(\tilde{\boldsymbol{z}}^{<t}, \boldsymbol{a}^{<t})$ to $\tilde{\boldsymbol{z}}^t$ is the same as first applying $\boldsymbol{v}$, then applying the latent transition model $\boldsymbol{\theta}$ and finally applying $\boldsymbol{v}^{-1}$. Equation (5.11) is reminiscent of Ahuja et al. [2022a], in which the mechanism $\tilde{\boldsymbol{g}}$ would be called an *imitator* of $\boldsymbol{g}$. Ahuja et al. [2022a] showed that $\sim_{\text{obs}}$ and $\sim_{\text{diff}}$ are actually one and the same. For completeness, we present an analogous argument here. We start by showing that $\boldsymbol{\theta} \sim_{\text{diff}} \tilde{\boldsymbol{\theta}}$ implies $\boldsymbol{\theta} \sim_{\text{obs}} \tilde{\boldsymbol{\theta}}$.

$$\begin{aligned}
p(\boldsymbol{x}^{\leq T} \mid \boldsymbol{a}^{<T}) &= \int \prod_{t=1}^{T} \left[p(\boldsymbol{x}^t \mid \boldsymbol{z}^t) p(\boldsymbol{z}^t \mid \boldsymbol{z}^{<t}, \boldsymbol{a}^{<t})\right] d\boldsymbol{z}^{\leq T} \\
&= \int \prod_{t=1}^{T} \left[p(\boldsymbol{x}^t \mid \boldsymbol{v}(\boldsymbol{z}^t)) p(\boldsymbol{v}(\boldsymbol{z}^t) \mid \boldsymbol{v}(\boldsymbol{z}^{<t}), \boldsymbol{a}^{<t})\right] |\det D\boldsymbol{v}(\boldsymbol{z}^{\leq T})| d\boldsymbol{z}^{\leq T} \\
&= \int \prod_{t=1}^{T} \left[p(\boldsymbol{x}^t \mid \boldsymbol{v}(\boldsymbol{z}^t)) p(\boldsymbol{v}(\boldsymbol{z}^t) \mid \boldsymbol{v}(\boldsymbol{z}^{<t}), \boldsymbol{a}^{<t}) |\det D\boldsymbol{v}(\boldsymbol{z}^t)|\right] d\boldsymbol{z}^{\leq T} \\
&= \int \prod_{t=1}^{T} \left[\tilde{p}(\boldsymbol{x}^t \mid \boldsymbol{z}^t) \tilde{p}(\boldsymbol{z}^t \mid \boldsymbol{z}^{<t}, \boldsymbol{a}^{<t})\right] d\boldsymbol{z}^{\leq T} = \tilde{p}(\boldsymbol{x}^{\leq T} \mid \boldsymbol{a}^{<T})\,,
\end{aligned}$$

where the second equality used the change-of-variable formula, the third equality used the fact that the Jacobian of $\boldsymbol{v}(\boldsymbol{z}^{\leq T})$ is block-diagonal (each block corresponds to a time step t) and the next to last equality used the definition of $\sim_{\text{diff}}$ and the fact that

$$p(\boldsymbol{x}^t \mid \boldsymbol{v}(\boldsymbol{z}^t)) = \mathcal{N}(\boldsymbol{x}^t; \boldsymbol{f}(\boldsymbol{f}^{-1} \circ \tilde{\boldsymbol{f}}(\boldsymbol{z}^t)), \sigma^2 \boldsymbol{I}) = \mathcal{N}(\boldsymbol{x}^t; \tilde{\boldsymbol{f}}(\boldsymbol{z}^t), \sigma^2 \boldsymbol{I}) = \tilde{p}(\boldsymbol{x}^t \mid \boldsymbol{z}^t)\,.$$

The following proposition establishes the converse, i.e. that $\boldsymbol{\theta} \sim_{\text{obs}} \tilde{\boldsymbol{\theta}}$ implies $\boldsymbol{\theta} \sim_{\text{diff}} \tilde{\boldsymbol{\theta}}$. Since its proof is more involved, we present it in the Appendix A.3. Note that this first identifiability result

is relatively weak and should be seen as a first step towards stronger guarantees. A very similar result was shown by Ahuja et al. [2022a, Theorem 3.1] to highlight the fact that the representation $\boldsymbol{f}$ is identifiable up to the equivariances $\boldsymbol{v}$ of the transition model p.

Proposition 5.2 (Identifiability up to diffeomorphism)**.** *Let $\boldsymbol{\theta} := (\boldsymbol{f}, p, \boldsymbol{G})$ and $\hat{\boldsymbol{\theta}} := (\hat{\boldsymbol{f}}, \hat{p}, \hat{\boldsymbol{G}})$ be two models satisfying Assumption 5.1. If $\boldsymbol{\theta} \sim_{\text{obs}} \hat{\boldsymbol{\theta}}$ (Def. 5.4), then $\boldsymbol{\theta} \sim_{\text{diff}} \hat{\boldsymbol{\theta}}$ (Def. 5.5).*

Intuitively, Proposition 5.2 shows that if two models agree on the distribution of the observations, then their "data manifold" $\boldsymbol{f}(\mathbb{R}^{d_z})$ and $\hat{\boldsymbol{f}}(\mathbb{R}^{d_z})$ are equal and their respective transition models are related via $\boldsymbol{v} := \boldsymbol{f}^{-1} \circ \hat{\boldsymbol{f}}$.

5.2.5. Disentanglement and equivalence up to permutation

A disentangled representation is often defined intuitively as a representation in which the coordinates are in one-to-one correspondence with *natural factors of variation* in the data. We are going to assume that these natural factors are captured by an unknown ground-truth decoder $\boldsymbol{f}$. Given a learned decoder $\hat{\boldsymbol{f}}$ such that $\boldsymbol{f}(\mathbb{R}^{d_z}) = \hat{\boldsymbol{f}}(\mathbb{R}^{d_z})$, the entanglement map $\boldsymbol{v} := \boldsymbol{f}^{-1} \circ \hat{\boldsymbol{f}}$ gives a correspondence between the learned representation $\hat{\boldsymbol{f}}$ and the natural factors of variations of $\boldsymbol{f}$. The following equivalence relation will help us define disentanglement.

Definition 5.6 (Equivalence up to permutation)**.** *We say two models $\boldsymbol{\theta} := (\boldsymbol{f}, p, \boldsymbol{G})$ and $\tilde{\boldsymbol{\theta}} := (\tilde{\boldsymbol{f}}, \tilde{p}, \tilde{\boldsymbol{G}})$ satisfying Assumptions 5.1, 5.2 & 5.3 are **equivalent up to permutation**, denoted $\boldsymbol{\theta} \sim_{\text{perm}} \tilde{\boldsymbol{\theta}}$, if and only if there exists a permutation matrix $\boldsymbol{P}$ such that*

(1) $\boldsymbol{\theta} \sim_{\text{diff}} \tilde{\boldsymbol{\theta}}$ (Def. 5.5) and $\tilde{\boldsymbol{G}}^a = \boldsymbol{P}\boldsymbol{G}^a$ and $\tilde{\boldsymbol{G}}^z = \boldsymbol{P}\boldsymbol{G}^z\boldsymbol{P}^\top$; and

(2) The entanglement map $\boldsymbol{v} := \boldsymbol{f}^{-1} \circ \tilde{\boldsymbol{f}}$ can be written as $\boldsymbol{v} = \boldsymbol{d} \circ \boldsymbol{P}^\top$, where $\boldsymbol{d}$ is element-wise, i.e. $\boldsymbol{d}_i(\boldsymbol{z})$ depends only on $\boldsymbol{z}_i$, for all i. In other words, the entanglement graph is $\boldsymbol{V} = \boldsymbol{P}^\top$.

The fact that the relation $\sim_{\text{perm}}$ is an equivalence relation is actually a special case of a more general result that we present later on in Section 5.3.3.

This allows us to give a formal definition of (complete) disentanglement. Note the we use the term *complete* to contrast with *partial* disentanglement.

Definition 5.7 (Complete disentanglement)**.** *Given a ground-truth model $\boldsymbol{\theta}$, we say a learned model $\hat{\boldsymbol{\theta}}$ is **completely disentangled** when $\boldsymbol{\theta} \sim_{\text{perm}} \hat{\boldsymbol{\theta}}$.*

Intuitively, a learned representation is *completely disentangled* when there is a one-to-one correspondence between its coordinates and those of the ground-truth representation (see Figure 5.2).

We define *partial* disentanglement, as something which lives strictly between equivalence up to diffeomorphism and equivalence up to permutation:

Definition 5.8 (Partial disentanglement)**.** *Given a ground-truth model $\boldsymbol{\theta}$, we say a learned model $\hat{\boldsymbol{\theta}}$ is **partially disentangled** when $\boldsymbol{\theta} \sim_{\text{diff}} \hat{\boldsymbol{\theta}}$ with an entanglement graph $\boldsymbol{V}$ (Definition 5.3) that is not a permutation nor the complete graph.*

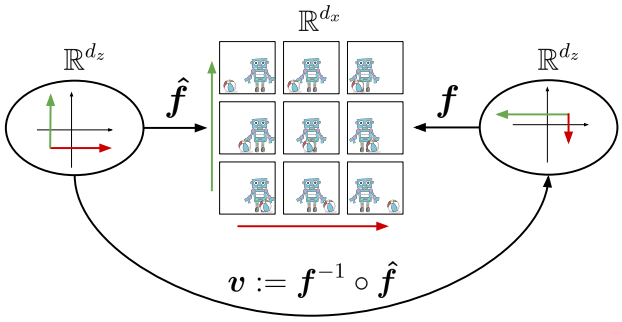


Figure 5.2. An illustration of disentanglement (Definition 5.7). The ground-truth decoder $\boldsymbol{f}$ captures the "natural factors of variations", which here are the x-positions of the robot and ball. The learned decoder $\hat{\boldsymbol{f}}$ is disentangled here because each of its latent coordinates corresponds exactly one objects in the scene. Mathematically, this is captured by the special structure of the entanglement map $\boldsymbol{v} := \boldsymbol{f}^{-1} \circ \hat{\boldsymbol{f}}$, which is a permutation composed with an element-wise invertible transformation.

This definition of partial disentanglement ranges from models that are almost completely entangled, i.e. those with a very dense entanglement graphs $\boldsymbol{V}$, to ones that are very close to being completely disentangled, i.e. those with a very sparse $\boldsymbol{V}$. The following section will make more precise how one can learn a completely or partially disentangled representation from data and exactly what form the entanglement graph is going to take.

5.3. Nonparametric partial disentanglement via mechanism sparsity

In this section, we provide a first theoretical insight as to why mechanism sparsity can lead to disentanglement (Section 5.3.1), introduce the machinery of $\boldsymbol{G}$-preserving maps (Section 5.3.2) which leads up to theorems showing identifiability up to $\boldsymbol{a}$-consistency (Section 5.3.3) and $\boldsymbol{z}$-consistency (Section 5.3.4), which corresponds to partial disentanglement. We also relate these results to interventions (Section 5.3.3.1), show how to combine both regularization on $\hat{\boldsymbol{G}}^a$ and $\hat{\boldsymbol{G}}^z$ to obtain stronger guarantees (Section 5.3.5) and introduce a graphical criterion guaranteeing complete disentanglement (Section 5.3.6). Finally, we introduce the *sufficient influence assumptions* and prove the identifiability results (Section 5.3.7), and provide multiple examples to build intuition (Section 5.3.8).

Before going further, we briefly introduce an abuse of notation that will be handy throughout: we will sometimes use vectors and matrices as sets of indices corresponding to their supports.

Definition 5.9 (Vectors & matrices as index sets)**.** *Let $\boldsymbol{a} \in \mathbb{R}^n$ and $\boldsymbol{A} \in \mathbb{R}^{m \times n}$. We will sometimes use $\boldsymbol{a}$ to denote the set of indices corresponding to the support of the vector $\boldsymbol{a}$, i.e.*

$$\boldsymbol{a} \sim \{i \in [n] \mid \boldsymbol{a}_i \neq 0\} . \tag{5.12}$$

This will allow us to write things like $i \in \boldsymbol{a}$ or $\boldsymbol{a} \subseteq \boldsymbol{b}$, where $\boldsymbol{b} \in \mathbb{R}^n$. We will use an analogous convention for matrices, i.e.,

$$\boldsymbol{A} \sim \{(i,j) \in [m] \times [n] \mid \boldsymbol{A}_{i,j} \neq 0\}\,, \tag{5.13}$$

This will allow us to write things like $(i,j) \in \boldsymbol{A}$ and $\boldsymbol{A} \subseteq \boldsymbol{B}$, where $\boldsymbol{B} \in \mathbb{R}^{m \times n}$.

5.3.1. A first mathematical insight for disentanglement via mechanism sparsity

In this section, we derive a first insight pointing towards how mechanism sparsity regularization, i.e. regularizing $\hat{\boldsymbol{G}}$ to be sparse, can promote disentanglement.

Recall that we would like to show that $\boldsymbol{\theta} \sim_{\text{obs}} \hat{\boldsymbol{\theta}}$ implies $\boldsymbol{\theta} \sim_{\text{perm}} \hat{\boldsymbol{\theta}}$, i.e. disentanglement (or partial disentanglement). Our approach will be to start from (5.10), which is guaranteed by Proposition 5.2, and perform a series of algebraic manipulations to gain mathematical insight into how regularizing $\hat{\boldsymbol{G}}$ to be sparse (mechanism sparsity) can induce disentanglement. A key manipulation will be taking first and second order derivatives. For this to be possible, we require a certain level of smoothness for the transition models:

Assumption 5.4 (Smoothness of transition model)**.** *When $\boldsymbol{a}$ is continuous, the transition densities $p(\boldsymbol{z}_i^t \mid \boldsymbol{z}^{<t}, \boldsymbol{a}^{<t})$ are C^2 functions from $\mathbb{R} \times \mathbb{R}^{d_z \times (t-1)} \times \mathcal{A}^t$ to $\mathbb{R}$ and $\mathcal{A} \subseteq \mathbb{R}^\ell$ is regular closed*[3]*. When $\boldsymbol{a}$ is discrete (e.g. Section 5.3.3.1), for all $\boldsymbol{a}^{<t}$, $p(\boldsymbol{z}_i^t \mid \boldsymbol{z}^{<t}, \boldsymbol{a}^{<t})$ are C^2 functions from $\mathbb{R} \times \mathbb{R}^{d_z \times (t-1)}$ to $\mathbb{R}$.*

We start by taking the $\log$ on both sides of (5.10) and let $q := \log p$ and $\hat{q} := \log \hat{p}$:

$$\hat{q}(\boldsymbol{z}^t \mid \boldsymbol{z}^{<t}, \boldsymbol{a}^{<t}) = q(\boldsymbol{v}(\boldsymbol{z}^t) \mid \boldsymbol{v}(\boldsymbol{z}^{<t}), \boldsymbol{a}^{<t}) + \log |\det D\boldsymbol{v}(\boldsymbol{z}^t)|\,. \tag{5.14}$$

We then take the derivative w.r.t. $\boldsymbol{z}^t$ on both sides:

$$D_z^t \hat{q}(\boldsymbol{z}^t \mid \boldsymbol{z}^{<t}, \boldsymbol{a}^{<t}) = D_z^t q(\boldsymbol{v}(\boldsymbol{z}^t) \mid \boldsymbol{v}(\boldsymbol{z}^{<t}), \boldsymbol{a}^{<t}) D\boldsymbol{v}(\boldsymbol{z}^t) + \eta(\boldsymbol{z}^t) \in \mathbb{R}^{1 \times d_z}\,, \tag{5.15}$$

where $D_z^t q$ denotes the Jacobian of $q(\boldsymbol{z}^t \mid \boldsymbol{z}^{<t}, \boldsymbol{a}^{<t})$ w.r.t. $\boldsymbol{z}^t$ and analogously for $D_z^t \hat{q}$. The term $\eta(\boldsymbol{z}^t)$ is the derivative of $\log |\det D\boldsymbol{v}(\boldsymbol{z}^t)|$ w.r.t. $\boldsymbol{z}^t$.

We differentiate[4] yet once more w.r.t. $\boldsymbol{a}^\tau$ for some $\tau < t$ (assuming $\boldsymbol{a}^t$ is continuous for now) and obtain

$$H_{z,a}^{t,\tau} \hat{q}(\boldsymbol{z}^t \mid \boldsymbol{z}^{<t}, \boldsymbol{a}^{<t}) = D\boldsymbol{v}(\boldsymbol{z}^t)^\top H_{z,a}^{t,\tau} q(\boldsymbol{v}(\boldsymbol{z}^t) \mid \boldsymbol{v}(\boldsymbol{z}^{<t}), \boldsymbol{a}^{<t}) \in \mathbb{R}^{d_z \times d_a}\,, \tag{5.16}$$

where $H_{z,a}^{t,\tau} q \in \mathbb{R}^{d_z \times d_a}$ is the Hessian matrix of second derivatives w.r.t. $\boldsymbol{z}^t$ and $\boldsymbol{a}^\tau$ and similarly for $H_{z,a}^{t,\tau} \hat{q}$.

[3] A set $\mathcal{A} \subseteq \mathbb{R}^\ell$ is regular closed when it is equal to the closure of its interior, i.e. $\overline{\mathcal{A}^\circ} = \mathcal{A}$.

[4] This derivative is well defined on $\mathcal{A}$ (in the sense that it does not depend on its C^k extension) since $\mathcal{A}$ is regular closed. We prove this general fact in Lemma 5.4 in the appendix.

We now look more closely at some specific entry (i, ℓ) of the Hessian $H^{t,\tau}_{z,a} q$. We first see that

$$\frac{\partial^2}{\partial \boldsymbol{a}^\tau_\ell \partial \boldsymbol{z}^t_i} q(\boldsymbol{z}^t \mid \boldsymbol{z}^{<t}, \boldsymbol{a}^{<t}) = \frac{\partial^2}{\partial \boldsymbol{a}^\tau_\ell \partial \boldsymbol{z}^t_i} \sum_{j=1}^{d_z} q(\boldsymbol{z}^t_j \mid \boldsymbol{z}^{<t}_{\mathbf{Pa}^z_j}, \boldsymbol{a}^{<t}_{\mathbf{Pa}^a_j}) \tag{5.17}$$

$$= \frac{\partial}{\partial \boldsymbol{a}^\tau_\ell} \sum_{j=1}^{d_z} \frac{\partial}{\partial \boldsymbol{z}^t_i} q(\boldsymbol{z}^t_j \mid \boldsymbol{z}^{<t}_{\mathbf{Pa}^z_j}, \boldsymbol{a}^{<t}_{\mathbf{Pa}^a_j}) \tag{5.18}$$

$$= \frac{\partial}{\partial \boldsymbol{a}^\tau_\ell} \frac{\partial}{\partial \boldsymbol{z}^t_i} q(\boldsymbol{z}^t_i \mid \boldsymbol{z}^{<t}_{\mathbf{Pa}^z_i}, \boldsymbol{a}^{<t}_{\mathbf{Pa}^a_i}) \,, \tag{5.19}$$

where the first equality holds by (5.2) & (5.3) and a basic property of logarithms. It is clear that (5.19) equals zero when $\ell \notin \mathbf{Pa}^a_i$. This is a crucial observation, since it implies that whenever $\boldsymbol{G}^a_{i,\ell} = 0$, we also have $(H^{t,\tau}_{z,a} q)_{i,\ell} = 0$. In other words, $H^{t,\tau}_{z,a} q \subseteq \boldsymbol{G}^a$. Note that the same argument can also be applied to get $H^{t,\tau}_{z,a} \hat{q} \subseteq \hat{\boldsymbol{G}}^a$.

Intuitive argument. We can start to see why regularizing $\hat{\boldsymbol{G}}$ to be sparse might induce disentanglement. Intuitively, a sparse $\hat{\boldsymbol{G}}^a$ forces $D\boldsymbol{v}(\boldsymbol{z}^t)$ to be sparse since otherwise the l.h.s. of (5.20) will not be sparse:

$$\underbrace{H^{t,\tau}_{z,a} \hat{q}(\boldsymbol{z}^t \mid \boldsymbol{z}^{<t}, \boldsymbol{a}^{<t})}_{\subseteq \hat{\boldsymbol{G}}^a} = \underbrace{D\boldsymbol{v}(\boldsymbol{z}^t)^\top}_{\text{forced to be sparse}} \underbrace{H^{t,\tau}_{z,a} q(\boldsymbol{v}(\boldsymbol{z}^t) \mid \boldsymbol{v}(\boldsymbol{z}^{<t}), \boldsymbol{a}^{<t})}_{\subseteq \boldsymbol{G}^a} \,, \tag{5.20}$$

And of course, the sparser $D\boldsymbol{v}(\boldsymbol{z}^t)$ is, the more disentangled $\hat{\boldsymbol{f}}$ is, since $D\boldsymbol{v}_{i,j} = 0$ everywhere implies $\boldsymbol{V}_{i,j} = 0$ under weak assumptions (Proposition 5.1). The above argument is not rigorous and is provided only to build intuition. It will be made formal later on.

Sparse temporal dependencies. In what precedes, we made use of the sparsity of the graph $\hat{\boldsymbol{G}}^a$ to argue that $D\boldsymbol{v}$ must also be sparse. We now show a similar intuition based on the sparsity of $\hat{\boldsymbol{G}}^z$. Starting from (5.15), instead of differentiating w.r.t. $\boldsymbol{a}^\tau$, we will differentiate w.r.t. $\boldsymbol{z}^\tau$, for some $\tau < t$, which yields:

$$H^{t,\tau}_{z,z} \hat{q}(\boldsymbol{z}^t \mid \boldsymbol{z}^{<t}, \boldsymbol{a}^{<t}) = D\boldsymbol{v}(\boldsymbol{z}^t)^\top H^{t,\tau}_{z,z} q(\boldsymbol{v}(\boldsymbol{z}^t) \mid \boldsymbol{v}(\boldsymbol{z}^{<t}), \boldsymbol{a}^{<t}) D\boldsymbol{v}(\boldsymbol{z}^\tau) \in \mathbb{R}^{d_z \times d_z} \,, \tag{5.21}$$

where $H^{t,\tau}_{z,z} q$ is the Hessian matrix of second derivatives of q w.r.t. $\boldsymbol{z}^t$ and $\boldsymbol{z}^\tau$, and analogously for $H^{t,\tau}_{z,z} \hat{q}$. Using an argument perfectly analogous to Equations (5.17) to (5.19), we can show that, whenever $\boldsymbol{G}^z_{i,j} = 0$, we also have $(H^{t,\tau}_{z,z} q)_{i,j} = 0$, and similarly for $\hat{\boldsymbol{G}}^z$ and $H^{t,\tau}_{z,z} \hat{q}$. In other words, $H^{t,\tau}_{z,z} q \subseteq \boldsymbol{G}^z$ and $H^{t,\tau}_{z,z} \hat{q} \subseteq \hat{\boldsymbol{G}}^z$. Therefore, analogously to (5.20), regularizing $\hat{\boldsymbol{G}}^z$ to be sparse intuitively should force $D\boldsymbol{v}$ to be sparse as well, i.e. bringing us closer to disentanglement:

$$\underbrace{H^{t,\tau}_{z,z} \hat{q}(\boldsymbol{z}^t \mid \boldsymbol{z}^{<t}, \boldsymbol{a}^{<t})}_{\subseteq \hat{\boldsymbol{G}}^z} = \underbrace{D\boldsymbol{v}(\boldsymbol{z}^t)^\top}_{\text{forced to be sparse}} \underbrace{H^{t,\tau}_{z,z} q(\boldsymbol{v}(\boldsymbol{z}^t) \mid \boldsymbol{v}(\boldsymbol{z}^{<t}), \boldsymbol{a}^{<t})}_{\subseteq \boldsymbol{G}^z} \underbrace{D\boldsymbol{v}(\boldsymbol{z}^\tau)}_{\text{forced to be sparse}} \,. \tag{5.22}$$

The crux of our technical contribution in this work is to make the above arguments formal and characterize precisely what will be the sparsity structure of $D\boldsymbol{v}(\boldsymbol{z})$ (hence of $\boldsymbol{V}$) based on the

ground-truth graph $\boldsymbol{G}$ (Theorems 5.1, 5.2 & 5.3). We also provide conditions on $\boldsymbol{G}$ to guarantee complete disentanglement (Proposition 5.7).

5.3.2. Graph preserving maps

Theorems 5.1, 5.2, 5.3 & 5.5 will show how regularizing $\hat{\boldsymbol{G}}$ to be sparse can force the dependency graph of the entanglement map $\boldsymbol{v}$ to be sparse as well. These results characterize the functional dependency structure of the entanglement map $\boldsymbol{v}$ as a function of the ground-truth graph $\boldsymbol{G}$. This link will be made precise thanks to the notion of graph preserving maps, which we define next. Before going further, we need to set up the following notation.

Definition 5.10 (Aligned subspaces of $\mathbb{R}^m$ and $\mathbb{R}^{m\times n}$)**.** *Given a binary vector $\boldsymbol{b} \in \{0,1\}^m$, let*

$$\mathbb{R}^m_{\boldsymbol{b}} := \{\boldsymbol{x} \in \mathbb{R}^m \mid \boldsymbol{b}_i = 0 \implies \boldsymbol{x}_i = 0\} \tag{5.23}$$

Given a binary matrix $\boldsymbol{B} \in \{0,1\}^{m\times n}$, let

$$\mathbb{R}^{m\times n}_{\boldsymbol{B}} := \{\boldsymbol{M} \in \mathbb{R}^{m\times n} \mid \boldsymbol{B}_{i,j} = 0 \implies \boldsymbol{M}_{i,j} = 0\}\,. \tag{5.24}$$

Note that $\mathbb{R}^m_{\boldsymbol{b}}$ and $\mathbb{R}^{m\times n}_{\boldsymbol{B}}$ are vector spaces under addition. This means that given $\boldsymbol{a}^{(1)}, \dots, \boldsymbol{a}^{(k)} \in \mathbb{R}^m_{\boldsymbol{b}}$, we have that $\operatorname{span}\{\boldsymbol{a}^{(1)}, \dots, \boldsymbol{a}^{(k)}\} \subseteq \mathbb{R}^m_{\boldsymbol{b}}$, where span denotes the subspace of all linear combinations. Similarly, given $\boldsymbol{A}^{(1)}, \dots, \boldsymbol{A}^{(k)} \in \mathbb{R}^{m\times n}_{\boldsymbol{B}}$, we have that $\operatorname{span}\{\boldsymbol{A}^{(1)}, \dots, \boldsymbol{A}^{(k)}\} \subseteq \mathbb{R}^{m\times n}_{\boldsymbol{B}}$.

To start reasoning formally about what will be the result of regularizing $\hat{\boldsymbol{G}}^a$ to be sparse, we temporarily assume that $\hat{\boldsymbol{G}}^a = \boldsymbol{G}^a$. With this assumption, we can interpret (5.20) as meaning that $D\boldsymbol{v}(\boldsymbol{z}^t)^\top$ must *preserve* the "sparsity structure" of the matrix $H^{t,\tau}_{z,a}q$. This observation motivates the following definitions, which will be central to our contribution.

Definition 5.11 ($\boldsymbol{G}$-preserving matrix)**.** *Given $\boldsymbol{G} \in \{0,1\}^{m\times n}$, a matrix $\boldsymbol{C} \in \mathbb{R}^{m\times m}$ is **$\boldsymbol{G}$-preserving** when*

$$\boldsymbol{C}^\top \mathbb{R}^{m\times n}_{\boldsymbol{G}} \subseteq \mathbb{R}^{m\times n}_{\boldsymbol{G}}\,.$$

Definition 5.12 ($\boldsymbol{G}$-preserving functions)**.** *Given $\boldsymbol{G} \in \{0,1\}^{m\times n}$, a function $\boldsymbol{c} : \mathbb{R}^m \to \mathbb{R}^m$ is **$\boldsymbol{G}$-preserving** when its dependency graph $\boldsymbol{C}$ (Definition 5.2) is $\boldsymbol{G}$-preserving.*

Without surprise, a linear map $\boldsymbol{c}(\boldsymbol{z}) := \boldsymbol{C}\boldsymbol{z}$ where $\boldsymbol{C} \in \mathbb{R}^{m\times m}$ is $\boldsymbol{G}$-preserving (Definition 5.12) if and only if the matrix $\boldsymbol{C}$ is $\boldsymbol{G}$-preserving (Definition 5.11).

We now show that $\boldsymbol{G}$-preserving functions can be defined alternatively in terms of a simple condition on their dependency graph. This characterization of $\boldsymbol{G}$-preserving functions is key to understand how (partial) disentanglement results from sparsity regularization.

Proposition 5.3. *A function $\boldsymbol{c}$ with dependency graph $\boldsymbol{C}$ (Definition 5.2) is $\boldsymbol{G}$-preserving if and only*

$$\boldsymbol{G}_{i,\cdot} \not\subseteq \boldsymbol{G}_{j,\cdot} \implies \boldsymbol{C}_{i,j} = 0, \text{ for all } i, j\,.$$

Proof We start by showing the "only if" statement. We suppose $\boldsymbol{G}_{i,\cdot} \not\subseteq \boldsymbol{G}_{j,\cdot}$ and must now show that $\boldsymbol{C}_{i,j} = 0$. We know there exists k such that $\boldsymbol{G}_{i,k} = 1$ but $\boldsymbol{G}_{j,k} = 0$. Since $\boldsymbol{C}^\top \mathbb{R}_{\boldsymbol{G}}^{m\times n} \subseteq \mathbb{R}_{\boldsymbol{G}}^{m\times n}$ and $\boldsymbol{e}_i \boldsymbol{e}_k^\top \in \mathbb{R}_{\boldsymbol{G}}^{m\times n}$, we must have that $\boldsymbol{C}^\top \boldsymbol{e}_i \boldsymbol{e}_k^\top \in \mathbb{R}_{\boldsymbol{G}}^{m\times n}$. Since $\boldsymbol{G}_{j,k} = 0$, we must have that $0 = (\boldsymbol{C}^\top \boldsymbol{e}_i \boldsymbol{e}_k^\top)_{j,k} = \boldsymbol{C}_{i,j}$.

We now show the "if" statement. Let $\boldsymbol{A} \in \mathbb{R}_{\boldsymbol{G}}^{m\times n}$. Take some (i, j) such that $\boldsymbol{G}_{i,j} = 0$. We must now show that $(\boldsymbol{C}^\top \boldsymbol{A})_{i,j} = 0$. We have that $(\boldsymbol{C}^\top \boldsymbol{A})_{i,j} = \sum_k \boldsymbol{C}_{k,i} \boldsymbol{A}_{k,j}$. We now check that each term in this sum must be zero. If $\boldsymbol{A}_{k,j} = 0$, of course the corresponding term is zero. If $\boldsymbol{A}_{k,j} \neq 0$, it implies that $\boldsymbol{G}_{k,j} = 1$ and thus $\boldsymbol{G}_{k,\cdot} \not\subseteq \boldsymbol{G}_{i,\cdot}$. By assumption, this implies that $\boldsymbol{C}_{k,i} = 0$ and thus $\boldsymbol{C}_{k,i} \boldsymbol{A}_{k,j} = 0$. Hence $(\boldsymbol{C}^\top \boldsymbol{A})_{i,j} = 0$ as desired. ■

We now characterize differentiable $\boldsymbol{G}$-preserving functions in terms of their Jacobian matrices.

Lemma 5.1. *A differentiable function $\boldsymbol{c} : \mathbb{R}^m \to \mathbb{R}^m$ is $\boldsymbol{G}$-preserving if and only if, for all $\boldsymbol{z} \in \mathbb{R}^m$, $D\boldsymbol{c}(\boldsymbol{z})$ is $\boldsymbol{G}$-preserving.*

Proof Assume $\boldsymbol{c}$ is $\boldsymbol{G}$-preserving with dependency graph $\boldsymbol{C}$. By Proposition 5.3, this is equivalent to having that, for all $i, j \in [n]$,

$$\boldsymbol{G}_{i,\cdot} \not\subseteq \boldsymbol{G}_{j,\cdot} \implies \boldsymbol{C}_{i,j} = 0\,. \tag{5.25}$$

But by Proposition 5.1, this statement is equivalent to

$$\boldsymbol{G}_{i,\cdot} \not\subseteq \boldsymbol{G}_{j,\cdot} \implies \forall \boldsymbol{z} \in \mathbb{R}^m,\ D\boldsymbol{c}(\boldsymbol{z})_{i,j} = 0\,, \tag{5.26}$$

which is equivalent to saying that $D\boldsymbol{c}(\boldsymbol{z})$ is $\boldsymbol{G}$-preserving for all $\boldsymbol{z} \in \mathbb{R}^m$ (again by Proposition 5.3). ■

We will now show that $\boldsymbol{G}$-preserving diffeomorphisms form a group under composition. To do so, we start by showing that invertible $\boldsymbol{G}$-preserving matrices form a group under matrix multiplication (Proposition 5.4) and extend the result to diffeomorphisms in Proposition 5.5.

Proposition 5.4. *Invertible $\boldsymbol{G}$-preserving matrices form a group under matrix multiplication.*

Proof We must show that the set of invertible $\boldsymbol{G}$-preserving matrices contains the identity, is closed under matrix multiplication and is closed under inversion.

Clearly, $\boldsymbol{I}$ is $\boldsymbol{G}$-preserving since $\boldsymbol{I}^\top \mathbb{R}_{\boldsymbol{G}}^{m\times n} = \mathbb{R}_{\boldsymbol{G}}^{m\times n}$.

Let $\boldsymbol{C}_1$ and $\boldsymbol{C}_2$ be $\boldsymbol{G}$-preserving. Then, $\boldsymbol{C}_1\boldsymbol{C}_2$ is $\boldsymbol{G}$-preserving because

$$(\boldsymbol{C}_1\boldsymbol{C}_2)^\top \mathbb{R}_{\boldsymbol{G}}^{m\times n} = \boldsymbol{C}_2^\top \boldsymbol{C}_1^\top \mathbb{R}_{\boldsymbol{G}}^{m\times n} \subseteq \boldsymbol{C}_2^\top \mathbb{R}_{\boldsymbol{G}}^{m\times n} \subseteq \mathbb{R}_{\boldsymbol{G}}^{m\times n}\,.$$

Let $\boldsymbol{C}$ be $\boldsymbol{G}$-preserving and invertible. Since $\boldsymbol{C}^\top$ is invertible as a map from $\mathbb{R}^{m\times n}$ to $\mathbb{R}^{m\times n}$, the dimensionality of the subspace $\mathbb{R}_{\boldsymbol{G}}^{m\times n}$ must be equal to the dimensionality of $\boldsymbol{C}^\top \mathbb{R}_{\boldsymbol{G}}^{m\times n}$. This fact combined with $\boldsymbol{C}^\top \mathbb{R}_{\boldsymbol{G}}^{m\times n} \subseteq \mathbb{R}_{\boldsymbol{G}}^{m\times n}$ imply that $\boldsymbol{C}^\top \mathbb{R}_{\boldsymbol{G}}^{m\times n} = \mathbb{R}_{\boldsymbol{G}}^{m\times n}$. Hence $\mathbb{R}_{\boldsymbol{G}}^{m\times n} = (\boldsymbol{C}^{-1})^\top \mathbb{R}_{\boldsymbol{G}}^{m\times n}$,

i.e. $\boldsymbol{C}^{-1}$ is $\boldsymbol{G}$-preserving. ■

We now extend the above results to diffeomorphisms using Proposition 5.1.

Proposition 5.5. *The set of $\boldsymbol{G}$-preserving diffeomorphims forms a group under composition.*

Proof We must show that the set of $\boldsymbol{G}$-preserving diffeomorphisms contains the identity, is closed under matrix multiplication and is closed under inversion.

The first statement is trivial since the entanglement graph of the identity diffeomorphism is the identity graph $\boldsymbol{C} := \boldsymbol{I}$, and of course it is $\boldsymbol{G}$-preserving.

We now prove the second statement. Let $\boldsymbol{c}$ and $\boldsymbol{c}'$ be two diffeomorphisms with dependency graph $\boldsymbol{C}$ and $\boldsymbol{C}'$ respectively. By the chain rule, we have that

$$D(\boldsymbol{c} \circ \boldsymbol{c}')(\boldsymbol{z}) = D\boldsymbol{c}(\boldsymbol{c}'(\boldsymbol{z}))D\boldsymbol{c}'(\boldsymbol{z})\,. \tag{5.27}$$

By Lemma 5.1, we have that $D\boldsymbol{c}(\boldsymbol{c}'(\boldsymbol{z}))$ and $D\boldsymbol{c}'(\boldsymbol{z})$ and $\boldsymbol{G}$-preserving matrices and, by Proposition 5.4 their product must also be $\boldsymbol{G}$-preserving. Hence $D(\boldsymbol{c} \circ \boldsymbol{c}')(\boldsymbol{z})$ is $\boldsymbol{G}$-preserving for all $\boldsymbol{z}$ and thus, by Lemma 5.1, $\boldsymbol{c} \circ \boldsymbol{c}'$ is $\boldsymbol{G}$-preserving.

The proof of the third statement has a similar flavor. By the inverse function theorem, we have

$$D\boldsymbol{c}^{-1}(\boldsymbol{z}) = D\boldsymbol{c}(\boldsymbol{c}^{-1}(\boldsymbol{z}))^{-1}\,. \tag{5.28}$$

Moreover, by Lemma 5.1, $D\boldsymbol{c}(\boldsymbol{c}^{-1}(\boldsymbol{z}))$ is $\boldsymbol{G}$-preserving. Furthermore, its inverse is also $\boldsymbol{G}$-preserving by Proposition 5.4. Similarly to the previous step, because $\boldsymbol{c}^{-1}$ is C^1, we can use Lemma 5.1 to conclude that $\boldsymbol{c}^{-1}$ is also $\boldsymbol{G}$-preserving. ■

5.3.3. Nonparameteric identifiability via auxiliary variables with sparse influence

In this section, we introduce our first identifiability results based on the sparsity of the graph $\boldsymbol{G}^a$ which describes the structure of the dependencies between $\boldsymbol{a}^{<t}$ and $\boldsymbol{z}^t$. We will see that, under some assumptions, regularizing the learned graph $\hat{\boldsymbol{G}}^a$ to be sparse will allow identifiability up to the following equivalence class:

Definition 5.13 ($\boldsymbol{a}$-consistency equivalence)**.** *We say two models $\boldsymbol{\theta} := (\boldsymbol{f}, p, \boldsymbol{G})$ and $\tilde{\boldsymbol{\theta}} := (\tilde{\boldsymbol{f}}, \tilde{p}, \tilde{\boldsymbol{G}})$ satisfying Assumptions 5.1, 5.2 & 5.3 are **$\boldsymbol{a}$-consistent**, denoted $\boldsymbol{\theta} \sim^{\boldsymbol{a}}_{\text{con}} \tilde{\boldsymbol{\theta}}$, if and only if there exists a permutation matrix $\boldsymbol{P}$ such that*

(1) $\boldsymbol{\theta} \sim_{\text{diff}} \tilde{\boldsymbol{\theta}}$ (Def. 5.5), and $\tilde{\boldsymbol{G}}^a = \boldsymbol{P}\boldsymbol{G}^a$; and

(2) the entanglement map $\boldsymbol{v} := \boldsymbol{f}^{-1} \circ \tilde{\boldsymbol{f}}$ can be written as $\boldsymbol{v} = \boldsymbol{c} \circ \boldsymbol{P}^\top$ where $\boldsymbol{c}$ is a $\boldsymbol{G}^a$-preserving diffeomorphism (Def. 5.12).

The main difference between $\boldsymbol{a}$-consistency (above definition) and permutation equivalence (Definition 5.6), is that, instead of having $\boldsymbol{v} = \boldsymbol{d} \circ \boldsymbol{P}^\top$ where $\boldsymbol{d}$ is element-wise, we have $\boldsymbol{v} = \boldsymbol{c} \circ \boldsymbol{P}^\top$ where $\boldsymbol{c}$ is $\boldsymbol{G}^a$-preserving, which allows for some mixing between the latent factors. Importantly, a $\boldsymbol{G}^a$-preserving map typically has missing edges in its dependency graph, as Proposition 5.3 shows. This means this equivalence relation imposes structure on the entanglement map $\boldsymbol{v}$. Depending on the structure of $\boldsymbol{G}^a$, this can mean either complete, partial or no disentanglement whatsoever. Note that the equivalence $\sim_{\text{perm}}$ is stronger than $\sim^{\boldsymbol{a}}_{\text{con}}$, in the sense that $\boldsymbol{\theta} \sim_{\text{perm}} \hat{\boldsymbol{\theta}} \implies \boldsymbol{\theta} \sim^{\boldsymbol{a}}_{\text{con}} \hat{\boldsymbol{\theta}}$. This is because element-wise transformations $\boldsymbol{d}$ are always $\boldsymbol{G}$-preserving, for any $\boldsymbol{G}$.

We demonstrate in Appendix A.4 that the $\boldsymbol{a}$-consistency relation is indeed an equivalence relation, as claimed in the the above definition. This follows from the fact that the set of $\boldsymbol{G}^a$-preserving diffeomorphisms forms a *group* under composition (Proposition 5.5).

The first result provides conditions under which regularizing the learned graph $\hat{\boldsymbol{G}}^a$ to be as sparse as the ground-truth graph $\boldsymbol{G}^a$ will induce the learned model to be $\boldsymbol{a}$-consistent with the ground-truth one.

Theorem 5.1 (Nonparametric disentanglement from continuous $\boldsymbol{a}$ with sparse influence)**.** *Let the parameters $\boldsymbol{\theta} := (\boldsymbol{f}, p, \boldsymbol{G})$ and $\hat{\boldsymbol{\theta}} := (\hat{\boldsymbol{f}}, \hat{p}, \hat{\boldsymbol{G}})$ correspond to two models satisfying Assumptions 5.1, 5.2, 5.3, & 5.4. Further assume that*

(1) ***[Observational equivalence]*** *$\boldsymbol{\theta} \sim_{\text{obs}} \hat{\boldsymbol{\theta}}$ (Def. 5.4);*

(2) ***[Sufficient influence of a]*** *The Hessian matrix $H^{t,\tau}_{z,a} \log p(\boldsymbol{z}^t \mid \boldsymbol{z}^{<t}, \boldsymbol{a}^{<t})$ varies "sufficiently", as formalized in Assumption 5.6;*

Then, there exists a permutation matrix $\boldsymbol{P}$ such that $\boldsymbol{P}\boldsymbol{G}^a \subseteq \hat{\boldsymbol{G}}^a$. Further assume that

(3) ***[Sparsity regularization]*** *$||\hat{\boldsymbol{G}}^a||_0 \leq ||\boldsymbol{G}^a||_0$;*

Then, $\boldsymbol{\theta} \sim^{\boldsymbol{a}}_{\text{con}} \hat{\boldsymbol{\theta}}$ (Def. 5.13).

The second assumption as well as a proof of this result is delayed to Section 5.3.7 for pedagogical reasons. We now describe and provide intuition about each assumption one by one.

Observational equivalence. The first assumption simply requires that both models agree about the observational model. In practice, this is achieved by fitting the model to data.

Sufficient influence. The second assumption requires that the "effect" of $\boldsymbol{a}^{<t}$ on $\boldsymbol{z}^t$ is "sufficiently strong". The assumption will be formalized and discussed in more details later in Sections 5.3.7 & 5.3.8, but we can already see that it concerns the Hessian matrix $H^{t,\tau}_{z,a} \log p$ that we saw earlier in Eq. (5.20) of Sec. 5.3.1.

Sparsity regularization. The first two assumptions imply that the learned graph $\hat{\boldsymbol{G}}^a$ is a supergraph of some permutation of the ground-truth graph $\boldsymbol{G}^a$. By adding the *sparsity regularization* assumption, we have that the learned graph $\hat{\boldsymbol{G}}^a$ *is exactly* a permutation of the ground-truth graph $\boldsymbol{G}^a$ and that, more precisely, the learned model is $\sim^{\boldsymbol{a}}_{\text{con}}$-equivalent to the ground-truth. This assumption is satisfied if $\hat{\boldsymbol{G}}^a$ is a minimal graph among all graphs that allow the model to exactly match the

ground-truth generative distribution. In Sec. 5.5, we suggest achieving this in practice by adding a sparsity penalty in the training objective, or by constraining the optimization problem.

$\boldsymbol{a}$-consistency. The final conclusion of the result states that the learned model is $\sim^{\boldsymbol{a}}_{\text{con}}$-equivalent to the ground-truth, which means the entanglement map $\boldsymbol{v} := \boldsymbol{f}^{-1} \circ \hat{\boldsymbol{f}}$ can be written as $\boldsymbol{v} = \boldsymbol{c} \circ \boldsymbol{P}^\top$ where $\boldsymbol{c}$ is $\boldsymbol{G}^a$-preserving. This is important since the $\boldsymbol{G}^a$-preserving condition imposes structure on the entanglement graph $\boldsymbol{V}$ (Definition 5.3), as implied by Proposition 5.3. In other words, the result predicts precisely which latent factors are expected to remain entangled.

Remark 5.1 (Inverse of $\boldsymbol{v}$)**.** *We defined $\boldsymbol{v}$ to be the mapping from the learned to the ground-truth representation, but in some context, it might be more telling to look at $\boldsymbol{v}^{-1}$, which maps from the ground-truth to the learned representation. If $\boldsymbol{v} = \boldsymbol{c} \circ \boldsymbol{P}^\top$ where $\boldsymbol{c}$ is $\boldsymbol{G}^a$-preserving (as predicted by Theorem 5.1), we know that its inverse is given by $\boldsymbol{v}^{-1} = \boldsymbol{P} \circ \boldsymbol{c}^{-1}$ where $\boldsymbol{c}^{-1}$ is $\boldsymbol{G}^a$-preserving, by closure under inversion (Proposition 5.5).*

The following result is the same as the above but for *discrete* auxiliary variables $\boldsymbol{a}$. This case is very important to cover the case where $\boldsymbol{a}$ indexes sparse interventions targeting the latent factors, which we discuss in more details in Section 5.3.3.1. Note that the only difference with the above theorem is the "sufficient influence" assumption, which we will present formally in Section 5.3.7 together with a proof of the result.

Theorem 5.2 (Nonparametric disentanglement via discrete $\boldsymbol{a}$ with sparse influence)**.** *Let the parameters $\boldsymbol{\theta} := (\boldsymbol{f}, p, \boldsymbol{G})$ and $\hat{\boldsymbol{\theta}} := (\hat{\boldsymbol{f}}, \hat{p}, \hat{\boldsymbol{G}})$ correspond to two models satisfying Assumptions 5.1, 5.2, 5.3 & 5.4. Further assume that*

*(1) **[Observational equivalence]** $\boldsymbol{\theta} \sim_{\text{obs}} \hat{\boldsymbol{\theta}}$ (Def. 5.4);*

*(2) **[Sufficient influence of $\boldsymbol{a}$]** The vector of derivatives $D^t_z \log p(\boldsymbol{z}^t \mid \boldsymbol{z}^{<t}, \boldsymbol{a}^{<t})$ depends "sufficiently strongly" on each component $\boldsymbol{a}_\ell$, as formalized in Assumption 5.7;*

Then, there exists a permutation matrix $\boldsymbol{P}$ such that $\boldsymbol{P}\boldsymbol{G}^a \subseteq \hat{\boldsymbol{G}}^a$. Further assume that

*(3) **[Sparsity regularization]** $||\hat{\boldsymbol{G}}^a||_0 \leq ||\boldsymbol{G}^a||_0$;*

Then, $\boldsymbol{\theta} \sim^{\boldsymbol{a}}_{\text{con}} \hat{\boldsymbol{\theta}}$ (Def. 5.13).

We now provide a few examples to illustrate how Theorems 5.1 & 5.2 can be applied. Here, we concentrate on the relationship between the graph $\boldsymbol{G}^a$ and the entanglement graph $\boldsymbol{V}$ (Definition 5.3). The question of whether or not the sufficient influence assumption is satisfied will be delayed to Section 5.3.8, where the examples will be made more concrete by specifying latent models more explicitly.

Example 5.2 ($\boldsymbol{G}^a = \boldsymbol{I}$ implies complete disentanglement)**.** *Assume $d_a = d_z$ and $\boldsymbol{G}^a = \boldsymbol{I}$, i.e. each latent variable is affected by only one auxiliary variable, and each auxiliary variable affects only one latent variable. The graph $\boldsymbol{G}^a$ is depicted is Figure 5.3a and $\boldsymbol{G}^z$ could be anything (see remark below). Assuming the ground-truth transition model satisfies the sufficient influence assumption of Theorem 5.1 or 5.2, we have that $\boldsymbol{\theta} \sim_{\text{obs}} \hat{\boldsymbol{\theta}}$ & $||\hat{\boldsymbol{G}}^a||_0 \leq ||\boldsymbol{G}^a||_0 \implies \boldsymbol{\theta} \sim^{\boldsymbol{a}}_{\text{con}} \hat{\boldsymbol{\theta}}$. This means there*

exists a permutation matrix $\boldsymbol{P}$ such that $\hat{\boldsymbol{G}}^a = \boldsymbol{P}\boldsymbol{G}^a$ and such that the entanglement map is given by $\boldsymbol{v} = \boldsymbol{c} \circ \boldsymbol{P}^\top$ where $\boldsymbol{c}$ is a $\boldsymbol{G}^a$-preserving diffeomorphism (Definition 5.11). But since $\boldsymbol{G}^a = \boldsymbol{I}$, Proposition 5.3 tells us that the dependency graph of $\boldsymbol{c}$ is simply $\boldsymbol{C} := \boldsymbol{I}$ and thus the entanglement graph is $\boldsymbol{V} = \boldsymbol{P}^\top$, i.e. complete disentanglement holds. In fact, one could add more columns to $\boldsymbol{G}^a$ (i.e. adding auxiliary variables) without changing the conclusion. Example 5.10 will provide a concrete example satisfying the sufficient influence assumption of Theorem 5.2.

Remark 5.2 (Temporal dependencies are not necessary)**.** *The above example did not mention anything about the temporal graph $\boldsymbol{G}^z$. That is because this graph could be anything, in fact, we could be in the special case where there is no temporal dependencies whatsoever, i.e. $T = 1$ and the latent model is simply $p(\boldsymbol{z} \mid \boldsymbol{a}) = \prod_{i=1}^{d_z} p(\boldsymbol{z}_i \mid \boldsymbol{a})$. In that case Theorems 5.1 & 5.2 could still be applied to prove identifiability of the representation, as long as their assumptions hold. This remark also applies to the next two examples.*

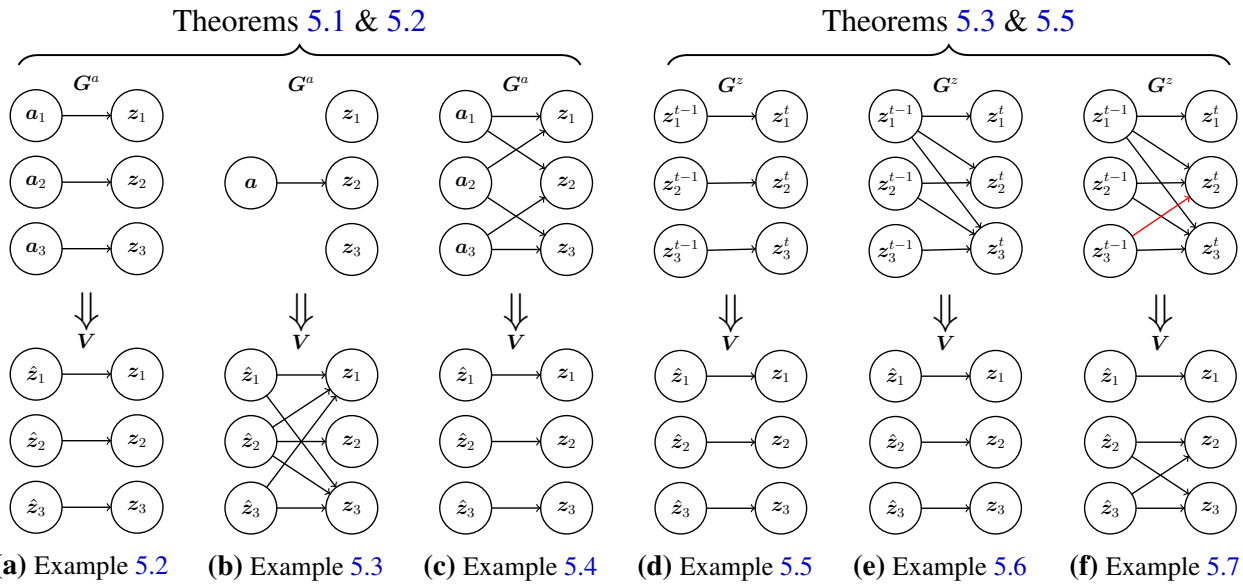
Example 5.3 (Action targeting a single latent variable identifies it)**.** *Consider the situation depicted in Figure 5.1 where $\boldsymbol{z}_1$ is the tree position, $\boldsymbol{z}_2$ is the robot position and $\boldsymbol{z}_3$ is the ball position ($d_z = 3$). Assume $\boldsymbol{a} \in \mathbb{R}$ corresponds to the torque applied to the wheels of the robot ($d_a = 1$). We thus have that $\boldsymbol{G}^a = [0, 1, 0]^\top$, i.e. $\boldsymbol{a}$ affects only $\boldsymbol{z}_2$. For the sake of this example, $\boldsymbol{G}^z$ can be anything, i.e. it does not have to be lower triangular like in Figure 5.1 (see remark above).*

If the sufficient influence assumption of Theorem 5.1 or 5.2 is satisfied, we have that $\boldsymbol{\theta} \sim_{\text{obs}} \hat{\boldsymbol{\theta}}$ & $||\hat{\boldsymbol{G}}^a||_0 \leq ||\boldsymbol{G}^a||_0$ implies $\boldsymbol{v} = \boldsymbol{c} \circ \boldsymbol{P}^\top$ where $\boldsymbol{P}$ is a permutation and $\boldsymbol{c}$ is a $\boldsymbol{G}^a$-preserving diffeomorphism. Using Proposition 5.3, this means the dependency graph of $\boldsymbol{c}$ is given by

$$\boldsymbol{C} = \begin{bmatrix} * & * & * \\ 0 & * & 0 \\ * & * & * \end{bmatrix}, \text{ since } \boldsymbol{G}^a_{2,\cdot} \not\subseteq \boldsymbol{G}^a_{1,\cdot} \text{ and } \boldsymbol{G}^a_{2,\cdot} \not\subseteq \boldsymbol{G}^a_{3,\cdot}\ , \tag{5.29}$$

where "$*$" indicates a potentially nonzero value. This means that one of the component of the learned representation will be an invertible transformation of the ground-truth variable $\boldsymbol{z}_2$ (robot position), while the other components could be a mixture of $\boldsymbol{z}_1$, $\boldsymbol{z}_2$ and $\boldsymbol{z}_3$. Figure 5.3b shows both the graph $\boldsymbol{G}^a$ and the corresponding entanglement graph $\boldsymbol{V}$ assuming $\boldsymbol{P} = \boldsymbol{I}$. Example 5.8 will make this example more concrete by specifying explicitly a latent model that satisfies the sufficient influence assumption of Theorem 5.1.

Example 5.4 (Complete disentanglement from multi-target actions)**.** *Assume $d_z = 3$ and $d_a = 3$ where $\boldsymbol{G}^a \in \mathbb{R}^{d_z \times d_a}$ is given by Figure 5.3c and the temporal graph $\boldsymbol{G}^z$ could be anything (see Remark 5.2 above). If the sufficient influence assumption of Theorem 5.1 or 5.2 is satisfied, then we have that $\boldsymbol{\theta} \sim_{\text{obs}} \hat{\boldsymbol{\theta}}$ & $||\hat{\boldsymbol{G}}^a||_0 \leq ||\boldsymbol{G}^a||_0$ implies $\boldsymbol{v} = \boldsymbol{c} \circ \boldsymbol{P}^\top$ where $\boldsymbol{P}$ is a permutation and $\boldsymbol{c}$ is a $\boldsymbol{G}^a$-preserving diffeomorphism. Proposition 5.3 implies that the dependency graph of $\boldsymbol{c}$ is simply $\boldsymbol{C} := \boldsymbol{I}$ because $\boldsymbol{G}^a_{i,\cdot} \not\subseteq \boldsymbol{G}^a_{j,\cdot}$ for all distinct i, j. This means we have complete disentanglement (Definition 5.7). Examples 5.9, 5.11 and 5.12 will explore more concrete instantiations of this*



(a) Example 5.2 **(b)** Example 5.3 **(c)** Example 5.4 **(d)** Example 5.5 **(e)** Example 5.6 **(f)** Example 5.7

Figure 5.3. Graphs $\boldsymbol{G}^a$ and $\boldsymbol{G}^z$ from Examples 5.2, 5.3, 5.4, 5.5, 5.6 & 5.7 with their respective entanglement graphs $\boldsymbol{V}$ (Definition 5.3) guaranteed by Theorems 5.1, 5.2, 5.3 & 5.5 (assuming $\boldsymbol{P} = \boldsymbol{I}$ for simplicity). Recall, that $\boldsymbol{V}$ describes the dependency structure of $\boldsymbol{v} = \boldsymbol{f}^{-1} \circ \hat{\boldsymbol{f}}$, which maps $\hat{\boldsymbol{z}}$ to $\boldsymbol{z}$. By Remark 5.1, the functional dependency graph of $\boldsymbol{v}^{-1}$ is exactly the same except for $\boldsymbol{z}$ and $\hat{\boldsymbol{z}}$ being interchanged.

example by specifying concrete latent models satisfying the sufficient influence assumptions of Theorems 5.1 and 5.2.

5.3.3.1. Unknown-target interventions on the latent factors. An important special case of Theorem 5.2 is when $\boldsymbol{a}^{t-1}$ corresponds to a one-hot vector indexing an *intervention with unknown targets* on the latent variables $\boldsymbol{z}^t$. This specific kind of intervention has been explored previously in the context of causal discovery where the intervention occurs on *observed* variables instead of *latent* variables like in our case [Eaton and Murphy, 2007, Mooij et al., 2020, Squires et al., 2020, Jaber et al., 2020, Brouillard et al., 2020, Ke et al., 2019]. Recently, multiple works in causal representation learning have considered interventions on latent variables [Lachapelle et al., 2022, Lippe et al., 2023b, Ahuja et al., 2023, Squires et al., 2023, Buchholz et al., 2023, von Kügelgen et al., 2023, Zhang et al., 2023, Jiang and Aragam, 2023] (see Section 5.7 for more). Here is how our framework can accommodate such interventions: Assume $\boldsymbol{a}^{t-1} \in \{\vec{0}, \boldsymbol{e}_1, ..., \boldsymbol{e}_{d_a}\}$, where each $\boldsymbol{e}_\ell$ is a one-hot vector. The action $\boldsymbol{a}^{t-1} = \vec{0}$ corresponds to the *observational setting*, i.e. when no intervention occurred, while $\boldsymbol{a}^{t-1} = \boldsymbol{e}_\ell$ corresponds to the ℓth intervention. In that context, the unknown graph $\boldsymbol{G}^a$ describes which latents are targeted by the intervention, i.e. $\ell \in \mathbf{Pa}_i^a$ if and only if $\boldsymbol{z}_i$ is targeted by the ℓth intervention. To see this, recall that, under Assumption 5.3, we have

$$p(\boldsymbol{z}_i^t \mid \boldsymbol{z}^{<t}, \boldsymbol{a}^{<t}) = p(\boldsymbol{z}_i^t \mid \boldsymbol{z}_{\mathbf{Pa}_i^z}^{<t}, \boldsymbol{a}_{\mathbf{Pa}_i^a}^{t-1}) \,, \tag{5.30}$$

where we implicitly assumed that $p(\boldsymbol{z}_i^t \mid \boldsymbol{z}^{<t}, \boldsymbol{a}^{<t})$ does not depend on $\boldsymbol{a}^{<t-1}$. In the observational setting, i.e. when $\boldsymbol{a}^{t-1} = \vec{0}$, the conditional becomes $p(\boldsymbol{z}_i^t \mid \boldsymbol{z}^{<t}, \vec{0})$. Now suppose we are in the ℓth intervention, i.e. $\boldsymbol{a}^{t-1} = \boldsymbol{e}_\ell$. Then, if $\ell \notin \mathbf{Pa}_i^a$, we have that $\boldsymbol{a}_{\mathbf{Pa}_i^a}^{t-1} = \vec{0}$, which means the conditional is also $p(\boldsymbol{z}_i^t \mid \boldsymbol{z}^{<t}, \vec{0})$, meaning variable $\boldsymbol{z}_i^t$ *is not* targeted by the ℓth intervention. When $\ell \in \mathbf{Pa}_i^a$, we have $\boldsymbol{a}_{\mathbf{Pa}_i^a}^{t-1} \neq \vec{0}$ and thus the conditional is allowed to change freely, i.e. $\boldsymbol{z}_i^t$ *is* targeted by the ℓth intervention.

Importantly, the assumption that $\boldsymbol{G}^a$ is sparse corresponds precisely to the *sparse mechanism shift* hypothesis from Schölkopf et al. [2021], i.e. that *only a few mechanisms change at a time.* Thm. 5.2 thus provides precise conditions for when sparse mechanism shifts induce disentanglement. Interestingly our theory covers both hard and soft interventions, as long as the sufficient influence assumption is satisfied.

Remark 5.3 (Examples revisited)**.** *Examples 5.2, 5.3 and 5.4 can be revisited while keeping in mind the "unknown-target intervention interpretation" in which $\boldsymbol{G}^a$ describes which latent variable is targeted by each intervention. For instance, Example 5.2 tells us that if each latent variable is targeted by a single-node intervention, then complete disentanglement is guaranteed. Examples 5.10, 5.11 and 5.12 provides mathematically concrete latent models where $\boldsymbol{a}$ is interpreted to be an intervention.*

Remark 5.4 (Causal representation learning without temporal dependencies)**.** *The special case where $T = 1$, i.e. no temporal dependencies, is of special interest. In that case, the latent variable model is simply $p(\boldsymbol{z} \mid \boldsymbol{a}) = \prod_{i=1}^{d_z} p(\boldsymbol{z}_i \mid \boldsymbol{a})$. In other words, the causal graph relating the $\boldsymbol{z}_i$ is empty. In contrast, recent work in causal representation learning showed how to obtain disentanglement in general latent causal graphical models without temporal dependencies, but are limited to single-node interventions [Ahuja et al., 2023, Squires et al., 2023, Buchholz et al., 2023, von Kügelgen et al., 2023, Zhang et al., 2023, Jiang and Aragam, 2023]. Although our framework with $T = 1$ assumes the causal graph between latent variables is empty, it allows for multi-node interventions which are sometimes sufficient to disentangle (Example 5.12). See Section 5.3.8.2 for more on this.*

5.3.4. Nonparametric identifiability via sparse temporal dependencies

This section is analogous to the previous one, but instead of leveraging the sparsity of $\boldsymbol{G}^a$ to show identifiability, it leverages the sparsity of $\boldsymbol{G}^z$, which describes the structure of the dependencies between the latents from one time step to another. We will see that, under some assumptions, regularizing the learned graph $\hat{\boldsymbol{G}}^z$ to be sparse will allow identifiability up to the following equivalence class:

Definition 5.14 ($\boldsymbol{z}$-consistency equivalence)**.** *We say two models $\boldsymbol{\theta} := (\boldsymbol{f}, p, \boldsymbol{G})$ and $\tilde{\boldsymbol{\theta}} := (\tilde{\boldsymbol{f}}, \tilde{p}, \tilde{\boldsymbol{G}})$ satisfying Assumptions 5.1, 5.2 & 5.3 are **$\boldsymbol{z}$-consistent**, denoted $\boldsymbol{\theta} \sim_{\text{con}}^{z} \tilde{\boldsymbol{\theta}}$, if and only if there exists a permutation matrix $\boldsymbol{P}$ such that*

(1) $\boldsymbol{\theta} \sim_{\text{diff}} \tilde{\boldsymbol{\theta}}$ *(Def. 5.5) and* $\tilde{\boldsymbol{G}}^z = \boldsymbol{P}\boldsymbol{G}^z\boldsymbol{P}^\top$*; and*

(2) the entanglement map $\boldsymbol{v} := \boldsymbol{f}^{-1} \circ \tilde{\boldsymbol{f}}$ *can be written as* $\boldsymbol{v} = \boldsymbol{c} \circ \boldsymbol{P}^\top$ *where* $\boldsymbol{c}$ *is a* $\boldsymbol{G}^z$*-preserving and* $(\boldsymbol{G}^z)^\top$*-preserving diffeomorphism (Definition 5.12).*

This relation can be shown to be an *equivalence* relation, as was the case for $\sim^a_{\text{con}}$. This is shown in Appendix A.4. Analogously to $\sim^a_{\text{con}}$, the equivalence relation $\sim^z_{\text{con}}$ relates the structure of the entanglement map $\boldsymbol{v}$ to the graph $\boldsymbol{G}^z$ via the notion of $\boldsymbol{G}$-preserving maps. It is also true that $\boldsymbol{\theta} \sim_{\text{perm}} \hat{\boldsymbol{\theta}} \implies \boldsymbol{\theta} \sim^z_{\text{con}} \hat{\boldsymbol{\theta}}$.

The following result is analogous to Theorems 5.1 and 5.2 where, instead of regularizing $\hat{\boldsymbol{G}}^a$ to be sparse, we regularize $\hat{\boldsymbol{G}}^z$. The next theorem shows how this type of sparsity regularization can induce the learned model to be $\boldsymbol{z}$-consistent with the ground-truth one.

Theorem 5.3 (Nonparametric disentanglement via sparse temporal dependencies)**.** *Let the parameters* $\boldsymbol{\theta} := (\boldsymbol{f}, p, \boldsymbol{G})$ *and* $\hat{\boldsymbol{\theta}} := (\hat{\boldsymbol{f}}, \hat{p}, \hat{\boldsymbol{G}})$ *correspond to two models satisfying Assumptions 5.1, 5.2, 5.3 & 5.4. Further assume that*

(1) ***[Observational equivalence]*** $\boldsymbol{\theta} \sim_{\text{obs}} \hat{\boldsymbol{\theta}}$ *(Def. 5.4);*

(2) ***[Sufficient influence of z]*** *The Hessian matrix* $H^{t,\tau}_{z,z} \log p(\boldsymbol{z}^t \mid \boldsymbol{z}^{<t}, \boldsymbol{a}^{<t})$ *varies "sufficiently", as formalized in Assumption 5.8;*

Then, there exists a permutation matrix $\boldsymbol{P}$ *such that* $\boldsymbol{P}\boldsymbol{G}^z\boldsymbol{P}^\top \subseteq \hat{\boldsymbol{G}}^z$*. Further assume that*

(3) ***[Sparsity regularization]*** $||\hat{\boldsymbol{G}}^z||_0 \leq ||\boldsymbol{G}^z||_0$*;*

Then, $\boldsymbol{\theta} \sim^z_{\text{con}} \hat{\boldsymbol{\theta}}$ *(Def. 5.14).*

The structure of the above theorem is very similar to Theorem 5.1 & 5.2. For example, we still have a "sufficient influence" condition, but this time it concerns the Hessian matrix $H^{t,\tau}_{z,z} \log p$ which we saw in Section 5.3.1, Equation (5.22). The conclusion is that both model will be $\boldsymbol{z}$-consistent, which means we recover the graph $\boldsymbol{G}^z$ up to permutation and have that the entanglement map $\boldsymbol{v}$ has a dependency graph given by $\boldsymbol{V} = \boldsymbol{C}\boldsymbol{P}^\top$ where $\boldsymbol{C}$ is $\boldsymbol{G}^z$- and $(\boldsymbol{G}^z)^\top$-preserving. Section 5.3.7 introduces the sufficient influence assumption formally as well as a proof of Theorem 5.3.

We now build intuition via some minimal examples which shows how one can apply the above theorem to draw links between the graph $\boldsymbol{G}^z$ and the resulting entanglement graph $\boldsymbol{V}$ (Definition 5.3). For now we simply assume that the assumption of sufficient influence (Assumption 5.8) is satisfied and wait until Section 5.3.8.3 to present more concrete transition models satisfying it.

Example 5.5 (Disentanglement via independent factors with temporal dependencies)**.** *Consider the situation depicted in Figure 5.3d where the graph* $\boldsymbol{G}^z = \boldsymbol{I}$*, i.e. the latents* $\boldsymbol{z}^t_i$ *are dependent in time but independent across dimensions. For this example, actions are unnecessary. Assuming the sufficient influence assumption of Theorem 5.3 is satisfied, we have that* $\boldsymbol{\theta} \sim_{\text{obs}} \hat{\boldsymbol{\theta}}$ *&* $||\hat{\boldsymbol{G}}^z||_0 \leq ||\boldsymbol{G}^z||_0 \implies \boldsymbol{\theta} \sim^z_{\text{con}} \hat{\boldsymbol{\theta}}$*, meaning there exists a permutation* $\boldsymbol{P}$ *such that* $\hat{\boldsymbol{G}}^z = \boldsymbol{P}\boldsymbol{G}^z\boldsymbol{P}^\top$ *and such that the entanglement map is given by* $\boldsymbol{v} = \boldsymbol{c} \circ \boldsymbol{P}^\top$ *where* $\boldsymbol{c}$ *is* $\boldsymbol{G}^z$*- and* $(\boldsymbol{G}^z)^\top$*-preserving. Using Proposition 5.3, one can verify that the dependency graph of* $\boldsymbol{c}$ *is* $\boldsymbol{C} = \boldsymbol{I}$ *and thus* $\boldsymbol{V} = \boldsymbol{P}^\top$*, i.e. the*

learned representation is completely disentangled. Example 5.13 will provide a concrete transition model where the sufficient influence assumption of Theorem 5.3 holds for this simple graph $\boldsymbol{G}^z$.

Example 5.6 (Disentanglement via sparsely dependent factors with temporal dependencies)**.** *The previous examples assumed independent latents, i.e. $\boldsymbol{G}^z = \boldsymbol{I}$. Instead, we now consider a more interesting "lower triangular" graph $\boldsymbol{G}^z$, as depicted in Figures 5.3e (This is the same graph as in the tree-robot-ball example of Figure 5.1). Again using Proposition 5.3, one can verify that $\boldsymbol{C} = \boldsymbol{I}$ and thus $\boldsymbol{V} = \boldsymbol{P}^\top$, i.e. the learned representation is completely disentangled. Example 5.13 will provide a concrete transition model where the sufficient influence assumption of Theorem 5.3 holds.*

Example 5.7 (Partial disentanglement via temporal sparsity)**.** *Assume the same situation as previously, but add an additional edge from $\boldsymbol{z}_B^{t-1}$ to $\boldsymbol{z}_R^t$ (see Figure 5.3f). This could occur, for example, if the robot tries to follow the ball, and is thus influenced by it. Using Proposition 5.3, one can show that $\boldsymbol{c}$ being $\boldsymbol{G}^z$- and $(\boldsymbol{G}^z)^\top$-preserving means that its dependency graph is given by*

$$\boldsymbol{C} = \begin{bmatrix} * & 0 & 0 \\ 0 & * & * \\ 0 & * & * \end{bmatrix} . \tag{5.31}$$

This means the robot and the ball remain entangled in the learned representation.

5.3.5. Combining sparsity regularization on $\hat{\boldsymbol{G}}^a$ & $\hat{\boldsymbol{G}}^z$

A natural question at this point is whether Theorem 5.1 (or Theorem 5.2) can be combined with Theorem 5.3 to obtain stronger guarantees. The answer is yes. In this section, we explain how this can be done. We would like to show how combining assumptions of Theorem 5.1 and Theorem 5.3 can yield identifiability up to the following stronger equivalence relation.

Definition 5.15 ($(\boldsymbol{a}, \boldsymbol{z})$-consistency equivalence)**.** *We say two models $\boldsymbol{\theta} := (\boldsymbol{f}, p, \boldsymbol{G})$ and $\tilde{\boldsymbol{\theta}} := (\tilde{\boldsymbol{f}}, \tilde{p}, \tilde{\boldsymbol{G}})$ satisfying Assumptions 5.1, 5.2 & 5.3 are $(\boldsymbol{a}, \boldsymbol{z})$-**consistent**, denoted $\boldsymbol{\theta} \sim_{\text{con}}^{\boldsymbol{z},\boldsymbol{a}} \tilde{\boldsymbol{\theta}}$, if and only if there exists a permutation matrix $\boldsymbol{P}$ such that*

(1) $\boldsymbol{\theta} \sim_{\text{diff}} \tilde{\boldsymbol{\theta}}$ (Def. 5.5) and $\tilde{\boldsymbol{G}}^a = \boldsymbol{P}^\top \boldsymbol{G}^a$ and $\tilde{\boldsymbol{G}}^z = \boldsymbol{P}^\top \boldsymbol{G}^z \boldsymbol{P}$; and

(2) the entanglement map $\boldsymbol{v} := \boldsymbol{f}^{-1} \circ \tilde{\boldsymbol{f}}$ can be written as $\boldsymbol{v} = \boldsymbol{c} \circ \boldsymbol{P}^\top$ where $\boldsymbol{c}$ is a $\boldsymbol{G}^a$-, $\boldsymbol{G}^z$- and $(\boldsymbol{G}^z)^\top$-preserving diffeomorphism (Def. 5.12).

Of course, if assumptions of both theorems hold, we must have that $\boldsymbol{\theta} \sim_{\text{con}}^{\boldsymbol{a}} \hat{\boldsymbol{\theta}}$ and $\boldsymbol{\theta} \sim_{\text{con}}^{\boldsymbol{z}} \hat{\boldsymbol{\theta}}$. As one might guess, this implies $\boldsymbol{\theta} \sim_{\text{con}}^{\boldsymbol{a},\boldsymbol{z}} \hat{\boldsymbol{\theta}}$, as the following proposition shows. The reason this result is not completely trivial is that the permutations $\boldsymbol{P}$ given by $\sim_{\text{con}}^{\boldsymbol{a}}$ and $\sim_{\text{con}}^{\boldsymbol{z}}$ might not be the same. Its proof can be found in Appendix A.4.1.

Proposition 5.6. *Let $\boldsymbol{\theta} := (\boldsymbol{f}, p, \boldsymbol{G})$ and $\tilde{\boldsymbol{\theta}} := (\tilde{\boldsymbol{f}}, \tilde{p}, \tilde{\boldsymbol{G}})$ be two models satisfying Assumptions 5.1, 5.2 & 5.3. We have $\boldsymbol{\theta} \sim_{\text{con}}^{\boldsymbol{z},\boldsymbol{a}} \tilde{\boldsymbol{\theta}}$ if and only if $\boldsymbol{\theta} \sim_{\text{con}}^{\boldsymbol{a}} \tilde{\boldsymbol{\theta}}$ and $\boldsymbol{\theta} \sim_{\text{con}}^{\boldsymbol{z}} \tilde{\boldsymbol{\theta}}$.*

We can thus combine both Theorems 5.1 (or Theorem 5.2) with Theorem 5.3 to obtain stronger guarantees. Practically, this means that regularizing both $\hat{\boldsymbol{G}}^a$ and $\hat{\boldsymbol{G}}^z$ to be sparse will lead to a more disentangled representation, i.e. a sparser entanglement graph $\boldsymbol{V}$, than if regularization was applied on only $\hat{\boldsymbol{G}}^a$ or only $\hat{\boldsymbol{G}}^z$.

5.3.6. Graphical criterion for complete disentanglement

The previous sections introduced results guaranteeing identifiability up to $\sim^{\boldsymbol{a}}_{\text{con}}$, $\sim^{\boldsymbol{z}}_{\text{con}}$ and $\sim^{\boldsymbol{z},\boldsymbol{a}}_{\text{con}}$ which all correspond to potentially *partial* disentanglement. This section provides an additional assumption to guarantee identifiability up to $\sim_{\text{perm}}$, i.e. *complete* disentanglement.

One can easily see from the definitions that $\boldsymbol{\theta} \sim_{\text{perm}} \hat{\boldsymbol{\theta}}$ holds precisely when $\boldsymbol{\theta} \sim^{\boldsymbol{a},\boldsymbol{z}}_{\text{con}} \hat{\boldsymbol{\theta}}$ with $\boldsymbol{C} = \boldsymbol{I}$. This condition can be achieved by making an extra assumption on $\boldsymbol{G}$. This assumption is taken directly from Lachapelle et al. [2022].

Assumption 5.5 (Graphical criterion, Lachapelle et al. [2022])**.** *Let $\boldsymbol{G} = [\boldsymbol{G}^z \; \boldsymbol{G}^a]$ be a graph. For all $i \in \{1, ..., d_z\}$,*

$$\left(\bigcap_{j \in \mathbf{Ch}^z_i} \mathbf{Pa}^z_j\right) \cap \left(\bigcap_{j \in \mathbf{Pa}^z_i} \mathbf{Ch}^z_j\right) \cap \left(\bigcap_{\ell \in \mathbf{Pa}^a_i} \mathbf{Ch}^a_\ell\right) = \{i\}\,,$$

where $\mathbf{Pa}^z_i$ and $\mathbf{Ch}^z_i$ are the sets of parents and children of node $\boldsymbol{z}_i$ in $\boldsymbol{G}^z$, respectively, while $\mathbf{Ch}^a_\ell$ is the set of children of $\boldsymbol{a}_\ell$ in $\boldsymbol{G}^a$.

The following proposition shows that when $\boldsymbol{G}$ satisfies the above criterion, the set of models that are $\sim^{\boldsymbol{a},\boldsymbol{z}}_{\text{con}}$-equivalent to $\boldsymbol{\theta}$ is equal to the set of models that are $\sim_{\text{perm}}$-equivalent to $\boldsymbol{\theta}$, thus allowing complete disentanglement. See Appendix A.6 for a proof.

Proposition 5.7 (Complete disentanglement as a special case)**.** *Let $\boldsymbol{\theta} := (\boldsymbol{f}, p, \boldsymbol{G})$ and $\hat{\boldsymbol{\theta}} := (\hat{\boldsymbol{f}}, \hat{p}, \hat{\boldsymbol{G}})$ be two models satisfying Assumptions 5.1, 5.2 & 5.3. If $\boldsymbol{\theta} \sim^{\boldsymbol{z},\boldsymbol{a}}_{\text{con}} \hat{\boldsymbol{\theta}}$ and $\boldsymbol{G}$ satisfies Assumption 5.5, then $\boldsymbol{\theta} \sim_{\text{perm}} \hat{\boldsymbol{\theta}}$.*

The above result shows that our general theory can guarantee complete disentanglement as a special case. This is one way in which our work generalizes the work of Lachapelle et al. [2022], in addition to relaxing the exponential family assumption. The following section explores how the exponential family assumption fits into our nonparameteric theory and how it allows one to simplify the "sufficient influence assumptions". But before, we provide some example to illustrate when Assumption 5.5 holds.

For example, the graphical criterion of Assumption 5.5 is trivially satisfied when $\boldsymbol{G}^z$ is diagonal, since $\{i\} = \mathbf{Pa}^z_i$ for all i (actions are not necessary here). This simple case amounts to having mutual independence between the sequences $\boldsymbol{z}_i^{\leq T}$, which is a standard assumption in the ICA literature [Tong et al., 1990, Hyvarinen and Morioka, 2017, Klindt et al., 2021]. The illustrative example we introduced in Fig. 5.1 has a more interesting "non-diagonal" graph satisfying our

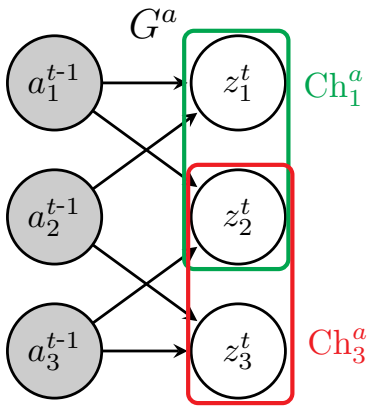


Figure 5.4. An example satisfying Assumption 5.5. Indeed, $\{\boldsymbol{z}_1\} = \mathbf{Ch}_1^a \cap \mathbf{Ch}_2^a$, $\{\boldsymbol{z}_2\} = \mathbf{Ch}_1^a \cap \mathbf{Ch}_3^a$ and $\{\boldsymbol{z}_3\} = \mathbf{Ch}_2^a \cap \mathbf{Ch}_3^a$.

criterion. Indeed, we have that $\{T\} = \mathbf{Pa}_T^{\boldsymbol{z}}$, $\{R\} = \mathbf{Ch}_R^{\boldsymbol{z}} \cap \mathbf{Pa}_R^{\boldsymbol{z}}$ and $\{B\} = \mathbf{Ch}_B^{\boldsymbol{z}}$. This example is actually part of an interesting family of graphs that satisfy our criterion:

Proposition 5.8 (Sufficient condition for the graphical criterion)**.** *If $\boldsymbol{G}_{i,i}^{\boldsymbol{z}} = 1$ for all i (all nodes have a self-loop) and $\boldsymbol{G}^{\boldsymbol{z}}$ has no 2-cycles, then $\boldsymbol{G}$ satisfies Assumption 5.5.*

Proof Self-loops guarantee $i \in \mathbf{Pa}_i^{\boldsymbol{z}} \cap \mathbf{Ch}_i^{\boldsymbol{z}}$ for all i. Suppose $j \in \mathbf{Pa}_i^{\boldsymbol{z}} \cap \mathbf{Ch}_i^{\boldsymbol{z}}$ for some $i \neq j$. This implies i and j form a 2-cycle, which is a contradiction. Thus $\{i\} = \mathbf{Pa}_i^{\boldsymbol{z}} \cap \mathbf{Ch}_i^{\boldsymbol{z}}$ for all i. ■

5.3.7. Proofs of Theorems 5.1, 5.2 & 5.3 and their sufficient influence assumptions

In this section, we introduce the sufficient influence assumptions and use them to prove Theorems 5.1, 5.2 & 5.3. In the next section (Section 5.3.8), we provide multiple examples to gain intuition about the sufficient influence assumptions. Throughout, the following lemma will come in handy.

Lemma 5.2 (Invertible matrix contains a permutation)**.** *Let $\boldsymbol{L} \in \mathbb{R}^{m \times m}$ be an invertible matrix. Then, there exists a permutation σ such that $\boldsymbol{L}_{i,\sigma(i)} \neq 0$ for all i, or in other words, $\boldsymbol{P}^\top \subseteq \boldsymbol{L}$ where $\boldsymbol{P}$ is the permutation matrix associated with σ, i.e. $\boldsymbol{P}\boldsymbol{e}_i = \boldsymbol{e}_{\sigma(i)}$. Note that this implies $\boldsymbol{P}\boldsymbol{L}$ and $\boldsymbol{L}\boldsymbol{P}$ have no zero on their diagonals.*

Proof Since the matrix $\boldsymbol{L}$ is invertible, its determinant is non-zero, i.e.

$$\det(\boldsymbol{L}) := \sum_{\sigma \in \mathfrak{S}_m} \operatorname{sign}(\sigma) \prod_{i=1}^{m} \boldsymbol{L}_{i,\sigma(i)} \neq 0\,, \tag{5.32}$$

where $\mathfrak{S}_m$ is the set of m-permutations. This equation implies that at least one term of the sum is non-zero, meaning there exists a permutation σ such that, for all i, $\boldsymbol{L}_{i,\sigma(i)} \neq 0$. ■

5.3.7.1. Sufficient influence assumption of Theorem 5.1 and its proof. We start by introducing the sufficient influence assumption of Theorem 5.1. Although it may seem terse at a first read, the reason why it is necessary will become clear when we prove the theorem.

Assumption 5.6 (Sufficient influence of $\boldsymbol{a}$ (nonparametric/continuous))**.** *For almost all $\boldsymbol{z} \in \mathbb{R}^{d_z}$ (i.e. except on a set with zero Lebesgue measure) and all $\ell \in [d_a]$, there exists*

$$\left\{(t_{(r)}, \tau_{(r)}, \boldsymbol{z}_{(r)}, \boldsymbol{a}_{(r)})\right\}_{r=1}^{|\mathbf{Ch}_\ell^{\boldsymbol{a}}|},$$

such that $t_{(r)} \in [T]$, $\tau_{(r)} < t_{(r)}$, $\boldsymbol{z}_{(r)} \in \mathbb{R}^{d_z \times (t_{(r)}-1)}$, $\boldsymbol{a}_{(r)} \in \mathcal{A}^{t_{(r)}}$ and

$$\operatorname{span}\left\{H_{z,a}^{t_{(r)},\tau_{(r)}} \log p(\boldsymbol{z} \mid \boldsymbol{z}_{(r)}, \boldsymbol{a}_{(r)})_{\cdot,\ell}\right\}_{r=1}^{|\mathbf{Ch}_\ell^{\boldsymbol{a}}|} = \mathbb{R}^{d_z}_{\mathbf{Ch}_\ell^{\boldsymbol{a}}}.$$

Proof [of Theorem 5.1] Recall equation (5.20), which we derived in Section 5.3.1:

$$\underbrace{H_{z,a}^{t,\tau}\hat{q}(\boldsymbol{z}^t \mid \boldsymbol{z}^{<t}, \boldsymbol{a}^{<t})}_{\subseteq \hat{\boldsymbol{G}}^a} = D\boldsymbol{v}(\boldsymbol{z}^t)^\top \underbrace{H_{z,a}^{t,\tau} q(\boldsymbol{v}(\boldsymbol{z}^t) \mid \boldsymbol{v}(\boldsymbol{z}^{<t}), \boldsymbol{a}^{<t})}_{\subseteq \boldsymbol{G}^a}. \tag{5.33}$$

Notice that Assumption 5.6 holds only "almost everywhere", i.e. on a set $\mathbb{R}^{d_z} \setminus E_0$ where E_0 has zero Lebesgue measure. Fix an arbitrary $\boldsymbol{z}^t \in \mathbb{R}^{d_z} \setminus E_0$. For notational convenience, define

$$\Lambda(\boldsymbol{z}, \gamma) := H_{z,a}^{t,\tau} q(\boldsymbol{v}(\boldsymbol{z}) \mid \boldsymbol{v}(\boldsymbol{z}^{<t}), \boldsymbol{a}^{<t}) \qquad \hat{\Lambda}(\boldsymbol{z}, \gamma) := H_{z,a}^{t,\tau} \hat{q}(\boldsymbol{z} \mid \boldsymbol{z}^{<t}, \boldsymbol{a}^{<t}),$$

where $\gamma := (t, \tau, \boldsymbol{z}^{<t}, \boldsymbol{a}^{<t})$. This allows us to rewrite (5.33) with a much lighter notation:

$$\hat{\Lambda}(\boldsymbol{z}, \gamma) = D\boldsymbol{v}(\boldsymbol{z})^\top \Lambda(\boldsymbol{z}, \gamma). \tag{5.34}$$

Now, notice that the sufficient influence assumption (Assumption 5.6) requires that, for all $\ell \in [d_a]$ there exists $\{\gamma_{(r)}\}_{r=1}^{|\mathbf{Ch}_\ell^a|}$ such that $\operatorname{span}\{\Lambda(\boldsymbol{z}, \gamma_{(r)})_{\cdot,\ell}\}_{r=1}^{|\mathbf{Ch}_\ell^a|} = \mathbb{R}^{d_z}_{\mathbf{Ch}_\ell^a}$. We can thus write

$$D\boldsymbol{v}(\boldsymbol{z})^\top \mathbb{R}^{d_z}_{\boldsymbol{G}^a_{\cdot,\ell}} = D\boldsymbol{v}(\boldsymbol{z})^\top \operatorname{span}\{\Lambda(\boldsymbol{z}, \gamma_{(r)})_{\cdot,\ell}\}_{r=1}^{|\mathbf{Ch}_\ell^a|} = \operatorname{span}\{\hat{\Lambda}(\boldsymbol{z}, \gamma_{(r)})_{\cdot,\ell}\}_{r=1}^{|\mathbf{Ch}_\ell^a|} \subseteq \mathbb{R}^{d_z}_{\hat{\boldsymbol{G}}^a_{\cdot,\ell}} \tag{5.35}$$

Since $D\boldsymbol{v}(\boldsymbol{z})$ is invertible, there exists a permutation $\boldsymbol{P}(\boldsymbol{z})$ such that $D\boldsymbol{v}(\boldsymbol{z})\boldsymbol{P}(\boldsymbol{z})$ has no zero on its diagonal (Lemma 5.2). Let $\boldsymbol{C}(\boldsymbol{z}) := D\boldsymbol{v}(\boldsymbol{z})\boldsymbol{P}(\boldsymbol{z})$. By left-multiplying (5.35) by $\boldsymbol{P}(\boldsymbol{z})^\top$, we get

$$\boldsymbol{C}(\boldsymbol{z})^\top \mathbb{R}^{d_z}_{\boldsymbol{G}^a_{\cdot,\ell}} \subseteq \mathbb{R}^{d_z}_{\boldsymbol{P}(\boldsymbol{z})^\top \hat{\boldsymbol{G}}^a_{\cdot,\ell}}. \tag{5.36}$$

We would like to show that $\boldsymbol{C}(\boldsymbol{z})$ is $\boldsymbol{G}^a$-preserving. Notice how the above equation is almost exactly the definition of $\boldsymbol{G}^a$-preserving. All that is left to prove is that $\boldsymbol{P}(\boldsymbol{z})^\top \hat{\boldsymbol{G}}^a = \boldsymbol{G}^a$.

We start by showing $\boldsymbol{P}(\boldsymbol{z})^\top \hat{\boldsymbol{G}}^a \supseteq \boldsymbol{G}^a$. Take $(i, \ell) \in \boldsymbol{G}^a$. Since $\boldsymbol{e}_i \in \mathbb{R}^{d_z}_{\boldsymbol{G}^a_{\cdot,\ell}}$, equation (5.36) implies

$$\boldsymbol{C}(\boldsymbol{z})^\top \boldsymbol{e}_i = \boldsymbol{C}(\boldsymbol{z})_{i,\cdot} \in \mathbb{R}^{d_z}_{\boldsymbol{P}(\boldsymbol{z})^\top \hat{\boldsymbol{G}}^a_{\cdot,\ell}}.$$

Since $\boldsymbol{C}(\boldsymbol{z})_{i,i} \neq 0$ (all elements on its diagonal are nonzero), we must have that $(i, \ell) \in \boldsymbol{P}(\boldsymbol{z})^\top \hat{\boldsymbol{G}}^a$.

Now, since $||\boldsymbol{P}(\boldsymbol{z})^\top \hat{\boldsymbol{G}}^a||_0 = ||\hat{\boldsymbol{G}}^a||_0 \leq ||\boldsymbol{G}^a||_0$, we have $\boldsymbol{P}(\boldsymbol{z})^\top \hat{\boldsymbol{G}}^a = \boldsymbol{G}^a$. This implies

$$\boldsymbol{C}(\boldsymbol{z})^\top \mathbb{R}^{d_z \times d_a}_{\boldsymbol{G}^a} \subseteq \mathbb{R}^{d_z \times d_a}_{\boldsymbol{G}^a}, \tag{5.37}$$

i.e. $\boldsymbol{C}(\boldsymbol{z})$ is a $\boldsymbol{G}^a$-preserving matrix, as desired.

To recap, we now have that, for all $\boldsymbol{z} \in \mathbb{R}^{d_z} \backslash E_0$, there exists a permutation $\boldsymbol{P}(\boldsymbol{z})$ s.t. $D\boldsymbol{v}(\boldsymbol{z})\boldsymbol{P}(\boldsymbol{z})$ is $\boldsymbol{G}^a$-preserving. We are not done yet, since, a priori, the permutation $\boldsymbol{P}(\boldsymbol{z})$ can be different for different values of $\boldsymbol{z}$, and we do not know what happens on the measure-zero set E_0. What we need to show is that there exists a permutation $\boldsymbol{P}$ such that, for all $\boldsymbol{z}$, $D\boldsymbol{v}(\boldsymbol{z})\boldsymbol{P}$ is $\boldsymbol{G}^a$-preserving. Lemma 5.12 in Appendix A.5 shows precisely this, by leveraging the continuity of $D\boldsymbol{v}(\boldsymbol{z})$ ($\boldsymbol{v}$ is a diffeomorphism and thus C^1).

Notice that $D(\boldsymbol{v} \circ \boldsymbol{P})(\boldsymbol{z}) = D\boldsymbol{v}(\boldsymbol{P}\boldsymbol{z})\boldsymbol{P}$, which is $\boldsymbol{G}^a$-preserving everywhere. Using Lemma 5.1, we conclude that the function $\boldsymbol{c} := \boldsymbol{v} \circ \boldsymbol{P}$ is $\boldsymbol{G}^a$-preserving. This concludes the proof. ■

Remark 5.5 (Alternative view on sufficient influence assumptions)**.** *Assumption 5.6, and all sufficient influence assumptions we present later on, can be thought of in terms of linear independence of functions. By definition, a family of functions $(f^{(i)} : X \to \mathbb{R})_{i=1}^n$ is linearly independent when $\sum_i \alpha_i f^{(i)}(x) = 0$ for all $x \in X$ implies $\alpha_i = 0$ for all i. It turns out that Assumption 5.6 is equivalent to requiring that, for all $\boldsymbol{z} \in \mathbb{R}^{d_z}$ and $\ell \in [d_a]$, the family of functions $(H^{t,\tau}_{z,a} \log p(\boldsymbol{z} \mid \boldsymbol{z}^{<t}, \boldsymbol{a}^{<t})_{i,\ell})_{i \in \mathbf{Ch}^a_\ell}$ (seen as functions of $t, \tau, \boldsymbol{z}^{<t}$ and $\boldsymbol{a}^{<t}$) is linearly independent. To see this, note that, in general, $(f^{(i)} : X \to \mathbb{R})_{i=1}^n$ is linearly independent iff there exist $x_1, ..., x_n \in X$ s.t. the vectors $((f^{(1)}(x_i), ..., f^{(n)}(x_i)))_{i=1}^n$ are linearly independent (see Appendix A.1 for a proof).*

5.3.7.2. Sufficient influence assumption of Theorem 5.2 and its proof. One can see that, if $\boldsymbol{a}$ is discrete, Theorem 5.1 cannot be applied because its sufficient influence assumption (Assumption 5.6) refers to the cross derivative of $\log p$ w.r.t. $\boldsymbol{z}^t$ and $\boldsymbol{a}^\tau$, which, of course, is not well defined when $\boldsymbol{a}$ is discrete. The discrete case is important to discuss interventions with unknown-targets as we did in Section 5.3.3.1, which is why we have a specialized result (Theorem 5.2) which has an analogous sufficient influence assumption based on *partial differences*.

Definition 5.16 (Partial difference)**.** *Let us define*

$$\Delta^{\tau,\epsilon}_{a,\ell} D^t_z \log p(\boldsymbol{z}^t \mid \boldsymbol{z}^{<t}, \boldsymbol{a}^{<t}) := D^t_z \log p(\boldsymbol{z}^t \mid \boldsymbol{z}^{<t}, \boldsymbol{a}^{<t} + \epsilon \boldsymbol{E}^{(\ell,\tau)}) - D^t_z \log p(\boldsymbol{z}^t \mid \boldsymbol{z}^{<t}, \boldsymbol{a}^{<t}),$$

where $\epsilon \in \mathbb{R}$ and $\boldsymbol{E}^{(\ell,\tau)}$ is a matrix with a one at entry (ℓ, τ) and zeros everywhere else.

One can see that $\Delta^{\tau,\epsilon}_{a,\ell} D^t_z \log p$ is essentially the discrete analog of $(H^{t,\tau}_{z,a} \log p)_{\cdot,\ell}$. Apart from this difference, the sufficient influence assumption for discrete $\boldsymbol{a}$ is the same as for continuous $\boldsymbol{a}$.

Assumption 5.7 (Sufficient influence of $\boldsymbol{a}$ (nonparametric/discrete))**.** *For almost all $\boldsymbol{z} \in \mathbb{R}^{d_z}$ (i.e. except on a set with zero Lebesgue measure) and all $\ell \in [d_a]$, there exists*

$$\{(t_{(r)}, \tau_{(r)}, \boldsymbol{z}_{(r)}, \boldsymbol{a}_{(r)}^{<t}, \epsilon_{(r)})\}_{r=1}^{|\mathbf{Ch}_\ell^{\boldsymbol{a}}|},$$

such that $t_{(r)} \in [T]$, $\tau_{(r)} < t_{(r)}$, $\boldsymbol{z}_{(r)} \in \mathbb{R}^{d_z \times (t_{(r)}-1)}$, $\boldsymbol{a}_{(r)} \in \mathcal{A}^{t_{(r)}}$, $\epsilon_{(r)} \in \mathbb{R}$, $(\boldsymbol{a}_{(r)})_{\cdot,\tau_{(r)}} + \epsilon_{(r)}\boldsymbol{e}_\ell \in \mathcal{A}$ and

$$\operatorname{span}\left\{\Delta_{a,\ell}^{\tau_{(r)},\epsilon_{(r)}} D_z^{t_{(r)}} \log p(\boldsymbol{z} \mid \boldsymbol{z}_{(r)}, \boldsymbol{a}_{(r)})\right\}_{r=1}^{|\mathbf{Ch}_\ell^{\boldsymbol{a}}|} = \mathbb{R}^{d_z}_{\mathbf{Ch}_\ell^{\boldsymbol{a}}}.$$

We can now provide a proof of Theorem 5.2. Note that it is almost identical to the proof of Theorem 5.1 except for the very first steps where we take a partial difference instead of a partial derivative.

Proof [of Theorem 5.2] We recall equation (5.15) derived in Section 5.3.1:

$$D_z^t \hat{q}(\boldsymbol{z}^t \mid \boldsymbol{z}^{<t}, \boldsymbol{a}^{<t}) = D_z^t q(\boldsymbol{v}(\boldsymbol{z}^t) \mid \boldsymbol{v}(\boldsymbol{z}^{<t}), \boldsymbol{a}^{<t}) D\boldsymbol{v}(\boldsymbol{z}^t) + \eta(\boldsymbol{z}^t) \in \mathbb{R}^{1 \times d_z}. \tag{5.38}$$

Now, instead of differentiating w.r.t. $\boldsymbol{a}_\ell^\tau$ for some $\tau < t$ and $\ell \in [d_a]$, we are going to take a partial difference. That is, we evaluate the above equation on at $\boldsymbol{a}^{<t}$ and $\boldsymbol{a}^{<t} + \epsilon \boldsymbol{E}^{(\ell,\tau)}$ and $\epsilon \in \mathbb{R}$, where $\boldsymbol{E}^{(\ell,\tau)}$ is a "one-hot matrix", while keeping everything else constant, and take the difference. This yields:

$$\begin{aligned}
&[D_z^t \hat{q}(\boldsymbol{z}^t \mid \boldsymbol{z}^{<t}, \boldsymbol{a}^{<t} + \epsilon \boldsymbol{E}^{(\ell,\tau)}) - D_z^t \hat{q}(\boldsymbol{z}^t \mid \boldsymbol{z}^{<t}, \boldsymbol{a}^{<t})]^\top \\
&\quad = D\boldsymbol{v}(\boldsymbol{z}^t)^\top [D_z^t q(\boldsymbol{v}(\boldsymbol{z}^t) \mid \boldsymbol{v}(\boldsymbol{z}^{<t}), \boldsymbol{a}^{<t} + \epsilon \boldsymbol{E}^{(\ell,\tau)}) - D_z^t q(\boldsymbol{v}(\boldsymbol{z}^t) \mid \boldsymbol{v}(\boldsymbol{z}^{<t}), \boldsymbol{a}^{<t})]^\top
\end{aligned} \tag{5.39}$$

$$\Delta_{a,\ell}^{\tau,\epsilon} D_z^t \hat{q}(\boldsymbol{z}^t \mid \boldsymbol{z}^{<t}, \boldsymbol{a}^{<t})^\top = D\boldsymbol{v}(\boldsymbol{z}^t)^\top \Delta_{a,\ell}^{\tau,\epsilon} D_z^t q(\boldsymbol{v}(\boldsymbol{z}^t) \mid \boldsymbol{v}(\boldsymbol{z}^{<t}), \boldsymbol{a}^{<t})^\top, \tag{5.40}$$

where we used the notation for partial differences introduced in Definition 5.16. Notice that the difference on the left is $\subseteq \hat{\boldsymbol{G}}_{\cdot,\ell}^a$ and the difference on the right is $\subseteq \boldsymbol{G}_{\cdot,\ell}^a$. This equation is thus analogous to (5.33) from the continuous case. For that reason, we can employ a completely analogous strategy. Hence, we define

$$\hat{\Lambda}(\boldsymbol{z}^t, \gamma)_{\cdot,\ell} := \Delta_{a,\ell}^{\tau,\epsilon} D_z^t \hat{q}(\boldsymbol{z}^t \mid \boldsymbol{z}^{<t}, \boldsymbol{a}^{<t})^\top \qquad \Lambda(\boldsymbol{z}^t, \gamma)_{\cdot,\ell} := \Delta_{a,\ell}^{\tau,\epsilon} D_z^t q(\boldsymbol{v}(\boldsymbol{z}^t) \mid \boldsymbol{v}(\boldsymbol{z}^{<t}), \boldsymbol{a}^{<t})^\top,$$

where $\gamma = (t, \tau, \boldsymbol{z}^{<t}, \boldsymbol{a}^{<t}, \vec{\epsilon})$. This notation allows us to rewrite (5.40) more compactly as

$$\underbrace{\hat{\Lambda}(\boldsymbol{z}^t, \gamma)}_{\subseteq \hat{\boldsymbol{G}}^a} = \boldsymbol{L}(\boldsymbol{z}^t)^\top \underbrace{\Lambda(\boldsymbol{z}^t, \gamma)}_{\subseteq \boldsymbol{G}^a}. \tag{5.41}$$

From here, the rest of the argument is exactly analogous to the proof of Theorem 5.1. ■

5.3.7.3. Sufficient influence assumption of Theorem 5.3 and its proof. We now introduce the sufficient influence assumption of Theorem 5.3, which showed how regularizing the temporal dependency graph $\hat{\boldsymbol{G}}^z$ to be sparse can result in disentanglement. Again, it is very similar to other sufficient influence assumptionss we saw so far.

Assumption 5.8 (Sufficient influence of $\boldsymbol{z}$ (nonparameteric))**.** *For almost all $\boldsymbol{z} \in \mathbb{R}^{d_z}$ (i.e. except on a set with zero Lebesgue measure), there exists*

$$\{(t_{(r)}, \tau_{(r)}, \boldsymbol{z}_{(r)}, \boldsymbol{a}_{(r)})\}_{r=1}^{||\boldsymbol{G}^z||_0},$$

such that $t_{(r)} \in [T]$, $\tau_{(r)} < t_{(r)}$, $\boldsymbol{z}_{(r)} \in \mathbb{R}^{d_z \times (t_{(r)}-1)}$, $\boldsymbol{a}_{(r)} \in \mathcal{A}^{t_{(r)}}$, $\boldsymbol{z} = \boldsymbol{z}_{(r)}^{\tau_{(r)}}$ and

$$\operatorname{span}\left\{H_{z,z}^{t_{(r)},\tau_{(r)}} q(\boldsymbol{z} \mid \boldsymbol{z}_{(r)}, \boldsymbol{a}_{(r)})\right\}_{r=1}^{||\boldsymbol{G}^z||_0} = \mathbb{R}^{d_z}_{\boldsymbol{G}^z}.$$

Proof [of Theorem 5.3] We recall equation (5.22) derived in Section 5.3.1:

$$\underbrace{H_{z,z}^{t,\tau} \hat{q}(\boldsymbol{z}^t \mid \boldsymbol{z}^{<t}, \boldsymbol{a}^{<t})}_{\subseteq \hat{\boldsymbol{G}}^z} = D\boldsymbol{v}(\boldsymbol{z}^t)^\top \underbrace{H_{z,z}^{t,\tau} q(\boldsymbol{v}(\boldsymbol{z}^t) \mid \boldsymbol{v}(\boldsymbol{z}^{<t}), \boldsymbol{a}^{<t})}_{\subseteq \boldsymbol{G}^z} D\boldsymbol{v}(\boldsymbol{z}^\tau). \tag{5.42}$$

This equation holds for all pairs of $\boldsymbol{z}^t$ and $\boldsymbol{z}^\tau$ in $\mathbb{R}^{d_z}$. We can thus evaluate it at a point such that $\boldsymbol{z}^t = \boldsymbol{z}^\tau$, which yields

$$H_{z,z}^{t,\tau} \hat{q}(\boldsymbol{z}^t \mid \boldsymbol{z}^{<t}, \boldsymbol{a}^{<t}) = D\boldsymbol{v}(\boldsymbol{z}^t)^\top H_{z,z}^{t,\tau} q(\boldsymbol{v}(\boldsymbol{z}^t) \mid \boldsymbol{v}(\boldsymbol{z}^{<t}), \boldsymbol{a}^{<t}) D\boldsymbol{v}(\boldsymbol{z}^t). \tag{5.43}$$

Recall that Assumption 5.8 holds for all $\boldsymbol{z}^t \in \mathbb{R}^{d_z} \setminus E_0$ where E_0 has Lebesgue measure zero. Fix an arbitrary $\boldsymbol{z}^t \in \mathbb{R}^{d_z} \setminus E_0$ and set $\boldsymbol{z}^\tau = \boldsymbol{z}^t$. Let us define

$$\Lambda(\boldsymbol{z}^t, \gamma) := H_{z,z}^{t,\tau} q(\boldsymbol{v}(\boldsymbol{z}^t) \mid \boldsymbol{v}(\boldsymbol{z}^{<t}), \boldsymbol{a}^{<t}) \qquad \hat{\Lambda}(\boldsymbol{z}^t, \gamma) := H_{z,z}^{t,\tau} \hat{q}(\boldsymbol{z}^t \mid \boldsymbol{z}^{<t}, \boldsymbol{a}^{<t}),$$

where $\gamma := (t, \tau, \boldsymbol{z}_{-\tau}^{<t}, \boldsymbol{a}^{<t})$ and $\boldsymbol{z}_{-\tau}^{<t}$ is $\boldsymbol{z}^{<t}$ but without $\boldsymbol{z}^\tau$. We can now rewrite (5.43) compactly as

$$\hat{\Lambda}(\boldsymbol{z}^t, \gamma) = D\boldsymbol{v}(\boldsymbol{z}^t)^\top \Lambda(\boldsymbol{z}^t, \gamma) D\boldsymbol{v}(\boldsymbol{z}^t). \tag{5.44}$$

Now, notice that the sufficient influence assumption (Assumption 5.8) requires that, there exists $\{\gamma_{(r)}\}_{r=1}^{||\boldsymbol{G}^z||_0}$ such that $\operatorname{span}\{\Lambda(\boldsymbol{z}^t, \gamma_{(r)})\}_{r=1}^{||\boldsymbol{G}^z||_0} = \mathbb{R}^{d_z \times d_z}_{\boldsymbol{G}^z}$. We can thus write

$$D\boldsymbol{v}(\boldsymbol{z}^t)^\top \operatorname{span}\{\Lambda(\boldsymbol{z}^t, \gamma_{(r)})\}_{r=1}^{||\boldsymbol{G}^z||_0} D\boldsymbol{v}(\boldsymbol{z}^t) = \operatorname{span}\{\hat{\Lambda}(\boldsymbol{z}^t, \gamma_{(r)})\}_{r=1}^{||\boldsymbol{G}^z||_0} \subseteq \mathbb{R}^{d_z \times d_z}_{\hat{\boldsymbol{G}}^z} \tag{5.45}$$

$$\implies D\boldsymbol{v}(\boldsymbol{z}^t)^\top \mathbb{R}^{d_z \times d_z}_{\boldsymbol{G}^z} D\boldsymbol{v}(\boldsymbol{z}^t) \subseteq \mathbb{R}^{d_z \times d_z}_{\hat{\boldsymbol{G}}^z} \tag{5.46}$$

Since $D\boldsymbol{v}(\boldsymbol{z})$ is invertible, there exists a permutation $\boldsymbol{P}(\boldsymbol{z})$ such that $D\boldsymbol{v}(\boldsymbol{z})\boldsymbol{P}(\boldsymbol{z})$ has no zero on its diagonal (Lemma 5.2). Let $\boldsymbol{C}(\boldsymbol{z}) := D\boldsymbol{v}(\boldsymbol{z})\boldsymbol{P}(\boldsymbol{z})$. If we left and right-multiply (5.46) by $\boldsymbol{P}(\boldsymbol{z})^\top$ and $\boldsymbol{P}(\boldsymbol{z})$, respectively, we obtain

$$\boldsymbol{C}(\boldsymbol{z}^t)^\top \mathbb{R}^{d_z \times d_z}_{\boldsymbol{G}^z} \boldsymbol{C}(\boldsymbol{z}^t) \subseteq \mathbb{R}^{d_z \times d_z}_{\boldsymbol{P}(\boldsymbol{z})^\top \hat{\boldsymbol{G}}^z \boldsymbol{P}(\boldsymbol{z})}. \tag{5.47}$$

We now show that $\boldsymbol{G}^z \subseteq \boldsymbol{P}(\boldsymbol{z})^\top \hat{\boldsymbol{G}}^z \boldsymbol{P}(\boldsymbol{z})$. Take $(i,j) \in \boldsymbol{G}^z$. Since $\boldsymbol{e}_i \boldsymbol{e}_j^\top \in \mathbb{R}^{d_z \times d_z}_{\boldsymbol{G}^z}$, equation (5.47) implies

$$\boldsymbol{C}(\boldsymbol{z}^t)^\top \boldsymbol{e}_i \boldsymbol{e}_j^\top \boldsymbol{C}(\boldsymbol{z}^t) = (\boldsymbol{C}(\boldsymbol{z}^t)_{i,\cdot})^\top \boldsymbol{C}(\boldsymbol{z}^t)_{j,\cdot} \subseteq \mathbb{R}^{d_z \times d_z}_{\boldsymbol{P}(\boldsymbol{z})^\top \hat{\boldsymbol{G}}^z \boldsymbol{P}(\boldsymbol{z})} \tag{5.48}$$

Since $\boldsymbol{C}(\boldsymbol{z}^t)_{i,i} \boldsymbol{C}(\boldsymbol{z}^t)_{j,j} \neq 0$ (recall the diagonal of $\boldsymbol{C}(\boldsymbol{z}^t)$ has no zero), we must have $(i,j) \in \boldsymbol{P}(\boldsymbol{z})^\top \hat{\boldsymbol{G}}^z \boldsymbol{P}(\boldsymbol{z})$. This shows that $\boldsymbol{G}^z \subseteq \boldsymbol{P}(\boldsymbol{z})^\top \hat{\boldsymbol{G}}^z \boldsymbol{P}(\boldsymbol{z})$.

Since $||\boldsymbol{P}(\boldsymbol{z})^\top \hat{\boldsymbol{G}}^z \boldsymbol{P}(\boldsymbol{z})||_0 = ||\hat{\boldsymbol{G}}^z||_0 \leq ||\boldsymbol{G}^z||_0$, we must have $\boldsymbol{G}^z = \boldsymbol{P}(\boldsymbol{z})^\top \hat{\boldsymbol{G}}^z \boldsymbol{P}(\boldsymbol{z})$, which yields

$$\boldsymbol{C}(\boldsymbol{z}^t)^\top \mathbb{R}^{d_z \times d_z}_{\boldsymbol{G}^z} \boldsymbol{C}(\boldsymbol{z}^t) \subseteq \mathbb{R}^{d_z \times d_z}_{\boldsymbol{G}^z} . \tag{5.49}$$

We are now going to show that the above implies that $\boldsymbol{C}(\boldsymbol{z}^t)$ is both $\boldsymbol{G}^z$-preserving and $(\boldsymbol{G}^z)^\top$-preserving. Start by rewriting (5.48) as follows:

$$\text{for all } (i,j) \in \boldsymbol{G}^z,\ (\boldsymbol{C}(\boldsymbol{z}^t)_{i,\cdot})^\top \boldsymbol{C}(\boldsymbol{z}^t)_{j,\cdot} \subseteq \mathbb{R}^{d_z \times d_z}_{\boldsymbol{G}^z} . \tag{5.50}$$

We start by showing $\boldsymbol{G}^z$-preservation. To do so, we leverage the characterization of Proposition 5.3. We must show that $\boldsymbol{G}^z_{i,\cdot} \not\subseteq \boldsymbol{G}^z_{j,\cdot}$ implies $\boldsymbol{C}(\boldsymbol{z}^t)_{i,j} = 0$. Because $\boldsymbol{G}^z_{i,\cdot} \not\subseteq \boldsymbol{G}^z_{j,\cdot}$, there must exists k s.t. $\boldsymbol{G}^z_{i,k} = 1$ and $\boldsymbol{G}^z_{j,k} = 0$. We thus have, by (5.50), that $(\boldsymbol{C}(\boldsymbol{z}^t)_{i,\cdot})^\top \boldsymbol{C}(\boldsymbol{z}^t)_{k,\cdot} \subseteq \mathbb{R}^{d_z \times d_z}_{\boldsymbol{G}^z}$. Because $\boldsymbol{G}^z_{j,k} = 0$, we have $\boldsymbol{C}(\boldsymbol{z}^t)_{i,j} \boldsymbol{C}(\boldsymbol{z}^t)_{k,k} = 0$. But since $\boldsymbol{C}(\boldsymbol{z}^t)_{k,k} \neq 0$, we must have that $\boldsymbol{C}(\boldsymbol{z}^t)_{i,j} = 0$, as desired. To show $(\boldsymbol{G}^z)^\top$-preservation, one can use a completely analogous argument.

We showed that $\boldsymbol{C}(\boldsymbol{z}^t)$ is $\boldsymbol{G}^z$-preserving and $(\boldsymbol{G}^z)^\top$-preserving. It is easy to verify that this is equivalent to being $[\boldsymbol{G}^z\ (\boldsymbol{G}^z)^\top]$-preserving (where $[\cdot\ \cdot]$ stands for column concatenation). This remark will be useful below.

Similarly to the proof of Theorem 5.1, we must now show that there exists a single permutation that works for all $\boldsymbol{z}^t \in \mathbb{R}^{d_z}$. To achieve this, we use Lemma 5.12 with $\boldsymbol{G} := [\boldsymbol{G}^z\ (\boldsymbol{G}^z)^\top]$ and $\boldsymbol{L}(\boldsymbol{z}) := D\boldsymbol{v}(\boldsymbol{z})$. This allows us to say that there exists a permutation $\boldsymbol{P}$ such that $D\boldsymbol{v}(\boldsymbol{z})\boldsymbol{P}$ is $[\boldsymbol{G}^z\ (\boldsymbol{G}^z)^\top]$-preserving for all $\boldsymbol{z}$ (not "almost all").

Notice that $D(\boldsymbol{v} \circ \boldsymbol{P})(\boldsymbol{z}) = D\boldsymbol{v}(\boldsymbol{P}\boldsymbol{z})\boldsymbol{P}$, which is $[\boldsymbol{G}^z\ (\boldsymbol{G}^z)^\top]$-preserving everywhere. Using Lemma 5.1, we conclude that the function $\boldsymbol{c} := \boldsymbol{v} \circ \boldsymbol{P}$ is $[\boldsymbol{G}^z\ (\boldsymbol{G}^z)^\top]$-preserving. ■

5.3.8. Examples to illustrate the scope of the theory

In this section, we provide several examples in order to gain better intuition as to when our results apply. Specifically, we will provide mathematically concrete examples of latent models $p(\boldsymbol{z}^t \mid \boldsymbol{z}^{<t}, \boldsymbol{a}^{<t})$ illustrating the various sufficient influence assumptions we introduced. All these examples are summarized in Table 5.2.

Even though our results are nonparametric, we will concentrate on the special case of Gaussian models which are useful to get a good intuition of what the sufficient influence assumptions mean. The following simple lemma will be useful in the following examples. We present it without proof as it can be derived from simple computations.

Lemma 5.3. *Let $p(\boldsymbol{z}) = \mathcal{N}(\boldsymbol{z}; \boldsymbol{\mu}, \boldsymbol{\Sigma})$ where $\boldsymbol{\mu} \in \mathbb{R}^{d_z}$ and $\boldsymbol{\Sigma} := \operatorname{diag}(\sigma_1^2, ..., \sigma_{d_z}^2)$. Then,*

$$D_z \log p(\boldsymbol{z}) = -\left[(\boldsymbol{z}_1 - \boldsymbol{\mu}_1)/\sigma_1^2, \ldots, (\boldsymbol{z}_{d_z} - \boldsymbol{\mu}_{d_z})/\sigma_{d_z}^2\right] \in \mathbb{R}^{1 \times d_z} . \tag{5.51}$$

5.3.8.1. Continuous auxiliary variable (Theorem 5.1). We start by illustrating Assumption 5.6 from Theorem 5.1. Example 5.8 assumes we observe continuous actions that targets each latent factor individually while Example 5.9 gives a multi-target example.

Example 5.8 (Sufficient influence for continuous single-target actions)**.** *We make Example 5.3 more concrete by specifying a latent transition model explicitly. Recall the situation depicted in Figure 5.1 where $\boldsymbol{z}_1$ is the tree position, $\boldsymbol{z}_2$ is the robot position and $\boldsymbol{z}_3$ is the ball position ($d_z = 3$). Assume $\boldsymbol{a} \in [-1, 1]$ corresponds to the amount of torque applied to the wheels of the robot. We thus have that $\boldsymbol{G}^a = [0, 1, 0]^\top$, i.e. $\boldsymbol{a}$ affects only the robot position $\boldsymbol{z}_2$. For this example, $\boldsymbol{G}^z$ can be anything. Let $p(\boldsymbol{z}^t \mid \boldsymbol{z}^{t-1}, \boldsymbol{a}) = \mathcal{N}(\boldsymbol{z}^t; \boldsymbol{\mu}(\boldsymbol{z}^{t-1}, \boldsymbol{a}), \sigma^2 \boldsymbol{I})$ where*

$$\boldsymbol{\mu}(\boldsymbol{z}^{t-1}, \boldsymbol{a}) := \boldsymbol{z}^{t-1} + \boldsymbol{g}(\boldsymbol{z}^{t-1}) + \boldsymbol{a} \cdot \boldsymbol{G}^a .$$

where $\boldsymbol{g} : \mathbb{R}^{d_z} \to \mathbb{R}^{d_z}$ is some function that satisfies the dependency graph $\boldsymbol{G}^z$ (e.g. $\boldsymbol{g}(\boldsymbol{z}) := \boldsymbol{W}\boldsymbol{z}$ where $\boldsymbol{W} \in \mathbb{R}^{d_z \times d_z}_{\boldsymbol{G}^z}$). If no torque is applied ($\boldsymbol{a} = 0$), then the position of the robots is determined by the dynamics of the system. However, adding positive or negative torque ($\boldsymbol{a} \neq 0$) nudges the robot to the right or to the left. Using Lemma 5.3, we can compute that

$$H^t_{z,a} \log p(\boldsymbol{z}^t \mid \boldsymbol{z}^{t-1}, \boldsymbol{a}) = [0, 1/\sigma^2, 0]^\top , \tag{5.52}$$

which of course spans $\mathbb{R}^3_{\{2\}}$ and thus Assumption 5.6 holds.

Example 5.9 (Sufficient influence for continuous multi-target actions)**.** *We make Example 5.4 more concrete by specifying an explicit latent model. Recall that $\boldsymbol{G}^a$ is given by Figure 5.3c with $d_z = d_a = 3$. Assume there are no temporal dependencies ($T = 1$), that $\boldsymbol{a} \in \mathbb{R}^3$ and that the latent model is given by $p(\boldsymbol{z} \mid \boldsymbol{a}) = \mathcal{N}(\boldsymbol{z}; \boldsymbol{\mu}(\boldsymbol{a}), \sigma^2 \boldsymbol{I})$ where*

$$\boldsymbol{\mu}(\boldsymbol{a}) := \begin{bmatrix} \boldsymbol{a}_1 \\ \boldsymbol{a}_1^2 \\ 0 \end{bmatrix} + \begin{bmatrix} \boldsymbol{a}_2 \\ 0 \\ \boldsymbol{a}_2^2 \end{bmatrix} + \begin{bmatrix} 0 \\ \boldsymbol{a}_3 \\ \boldsymbol{a}_3^2 \end{bmatrix} . \tag{5.53}$$

Using Lemma 5.3 we can compute

$$H_{z,a}q(\boldsymbol{z} \mid a) = \frac{1}{\sigma^2}\begin{bmatrix} 1 & 1 & 0 \\ 2\boldsymbol{a}_1 & 0 & 1 \\ 0 & 2\boldsymbol{a}_2 & 2\boldsymbol{a}_3 \end{bmatrix}. \tag{5.54}$$

Consider $\ell = 1$ so that $\mathbf{Ch}_1^a = \{1, 2\}$. We can see that $H_{z,a}q(\boldsymbol{z} \mid a = 0)_{\cdot,1} = [1, 0, 0]^\top$ and $H_{z,a}q(\boldsymbol{z} \mid a = \boldsymbol{e}_1)_{\cdot,1} = [1, 2, 0]^\top$ span $\mathbb{R}^3_{\{1,2\}}$. Analogous conclusions can be reached also for $\ell = 2, 3$, which shows Assumption 5.6 holds.

Now suppose that we instead had that $\boldsymbol{\mu}(\boldsymbol{a})$ was a linear map, i.e. $\boldsymbol{\mu}(\boldsymbol{a}) := \boldsymbol{W}\boldsymbol{a}$ where $\boldsymbol{W} \in \mathbb{R}^{d_z \times d_a}_{\boldsymbol{G}^a}$. This would imply that $H_{z,a}q(\boldsymbol{z} \mid a) \propto \boldsymbol{W}$, which means it cannot satisfy the sufficient influence assumption (unless $||\boldsymbol{G}^a_{\cdot,\ell}||_0 \leq 1$ for all ℓ).

5.3.8.2. Discrete auxiliary variable or interventions (Theorem 5.2). We now provide three concrete examples of latent models $p(\boldsymbol{z}^t \mid \boldsymbol{z}^{<t}, \boldsymbol{a}^{<t})$ that satisfy Assumption 5.7, from Theorem 5.2. Here, we interpret the discrete auxiliary variable $\boldsymbol{a}$ as an *intervention index*, as discussed in Section 5.3.3.1, but note that other interpretations are possible (like $\boldsymbol{a}$ as an action). Recall that our identifiability result do not require the knowledge of the targets of the interventions, these can be learned.

Example 5.10 shows how single target interventions can be used to obtain complete disentanglement without temporal dependencies, Example 5.11 shows how multi-target interventions can be leverage for disentanglement if temporal dependencies are present and Example 5.11 shows how *grouped* multi-target interventions allow disentanglement even when there is no time dependencies (Remark 5.6).

Example 5.10 (Single-target interventions for complete disentanglement without time)**.** *We make Example 5.2 more concrete by specifying an explicit latent model. Assume $d_a = d_z$ and that $\boldsymbol{a} \in \mathcal{A} := \{\mathbf{0}, \boldsymbol{e}_1, \ldots, \boldsymbol{e}_{d_a}\}$ is interpreted to be an intervention index (see Section 5.3.3.1). Furthermore, Example 5.2 assumed $\boldsymbol{G}^a = \boldsymbol{I}$, i.e. each latent factor is targeted once by an intervention that targets only this factor (the example actually allowed to add arbitrary columns to $\boldsymbol{G}^a$, i.e. adding more interventions, without compromising complete disentanglement). Assume there are no temporal dependencies, i.e. $T = 1$, and that $p(\boldsymbol{z} \mid \boldsymbol{a}) := \mathcal{N}(\boldsymbol{z}; \boldsymbol{\mu}(\boldsymbol{a}), \operatorname{diag}(\boldsymbol{\sigma}^2(\boldsymbol{a})))$ with*

$$\boldsymbol{\mu}(\boldsymbol{a}) := \boldsymbol{\mu} \odot \boldsymbol{a} \quad \text{and} \quad \boldsymbol{\sigma}^2(\boldsymbol{a}) := \mathbb{1} + \boldsymbol{\delta} \odot \boldsymbol{a}\,, \tag{5.55}$$

where $\odot$ denotes the Hadamard product (a.k.a. element-wise product), $\boldsymbol{\mu} \in \mathbb{R}^{d_z}$ is the vector of means for each intervention and $\boldsymbol{\delta} \in \mathbb{R}^{d_z}$ is the vector of shifts in variance for all interventions. Thus, in the observational setting ($\boldsymbol{a} = \mathbf{0}$), we have $\mu(\boldsymbol{a}) = \mathbf{0}$ and $\boldsymbol{\sigma}(\boldsymbol{a}) = \mathbb{1}$ while in the ℓth intervention ($\boldsymbol{a} = \boldsymbol{e}_\ell$), the mean and variance of the targeted latent shift while the others stay the same, i.e. $\boldsymbol{\mu}(\boldsymbol{a}) = \boldsymbol{\mu}_\ell \boldsymbol{e}_\ell$ and $\boldsymbol{\sigma}^2(\boldsymbol{a}) = \mathbb{1} + \boldsymbol{\delta}\boldsymbol{e}_\ell$ (assume the shifted variance is > 0). Using

Lemma 5.3, we can compute

$$\Delta^{\epsilon=1}_{a,\ell} D_z \log p(\boldsymbol{z} \mid \boldsymbol{a} = \boldsymbol{0}) := D_z \log p(\boldsymbol{z} \mid \boldsymbol{a} = \boldsymbol{e}_\ell) - D_z \log p(\boldsymbol{z} \mid \boldsymbol{a} = \boldsymbol{0}) = \frac{\boldsymbol{\mu}_\ell + \boldsymbol{\delta}_\ell \boldsymbol{z}_\ell}{1 + \boldsymbol{\delta}_\ell} \boldsymbol{e}_\ell \,,$$

which must span $\mathbb{R}^{d_z}_{\{\ell\}}$ unless $\boldsymbol{\mu}_\ell + \boldsymbol{\delta}_\ell \boldsymbol{z}_\ell = 0$. But note that when, for all ℓ, $\boldsymbol{\mu}_\ell \neq 0$ or $\boldsymbol{\delta}_\ell \neq 0$ (i.e. all interventions truly have an effect), the set $\{\boldsymbol{z} \in \mathbb{R}^{d_z} \mid \boldsymbol{\mu}_\ell + \boldsymbol{\delta}_\ell \boldsymbol{z}_\ell = 0 \text{ for some } \ell\}$ has zero Lebesgue measure in $\mathbb{R}^{d_z}$, which is allowed by Assumption 5.7.

Remark 5.6 (Potential issues with multi-target interventions without time)**.** *What if an intervention targets more than one latent at a time? Can it still satisfy the sufficient influence assumption? We will now see that, without time-dependencies ($T = 1$), it is impossible. Consider the simple situation where $d_z = 3$, $d_a = 1$, $\boldsymbol{a} \in \{0, 1\}$ and $\boldsymbol{G}^a = [1, 1, 0]^\top$, i.e. there is a single intervention targeting $\boldsymbol{z}_1$ and $\boldsymbol{z}_2$. In that case, there is a single possible difference vector which is*

$$\Delta^{\epsilon=1}_{a} D_z \log p(\boldsymbol{z} \mid \boldsymbol{a} = 0) = D_z \log p(\boldsymbol{z} \mid \boldsymbol{a} = 1) - D_z \log p(\boldsymbol{z} \mid \boldsymbol{a} = 0) \in \mathbb{R}^{d_z}_{\{1,2\}} \,.$$

Since this is the only difference vector, we can see that we cannot span the 2-dimensional space $\mathbb{R}^{d_z}_{\{1,2\}}$. Therefore, to leverage multi-target interventions in our framework, more "variability" is required. Example 5.11 below shows how temporal dependencies can provide this additional variability while Example 5.12 shows how having "groups" of interventions known to have the same (unknown) targets can also provide the required variability.

Example 5.11 (Multi-target interventions for complete disentanglement with time)**.** *We make Example 5.4 more concrete by specifying an explicit latent model that satisfies Assumption 5.7. Recall $d_z = 3$, $d_a = 3$ and $\boldsymbol{G}^a$ is depicted in Figure 5.3c. This time, we assume there are temporal dependencies, i.e. $T > 1$ and $\boldsymbol{G}^z$ is non-trivial. Suppose $\boldsymbol{a} \in \mathcal{A} := \{\boldsymbol{0}, \boldsymbol{e}_1, \boldsymbol{e}_2, \boldsymbol{e}_3\}$ where $\boldsymbol{e}_\ell$ is the ℓth one-hot and we interpret $\boldsymbol{0}$ to correspond to the observational setting and $\boldsymbol{e}_\ell$ to correspond to the ℓth intervention. Recall that in this interpretation, $\boldsymbol{G}^a$ describes which latent variable is targeted by each intervention. Let $p(\boldsymbol{z}^t \mid \boldsymbol{z}^{t-1}, \boldsymbol{a}) = \mathcal{N}(\boldsymbol{z}^t; \boldsymbol{\mu}(\boldsymbol{z}^{t-1}, \boldsymbol{a}), \sigma^2 \boldsymbol{I})$ where*

$$\boldsymbol{\mu}(\boldsymbol{z}^{t-1}, \boldsymbol{a}) := \boldsymbol{z}^{t-1} + (\mathbb{1} - \boldsymbol{G}^a \boldsymbol{a}) \odot \boldsymbol{g}(\boldsymbol{z}^{t-1}) \,,$$

where $\boldsymbol{g} : \mathbb{R}^{d_z} \to \mathbb{R}^{d_z}$ is some function respecting the graph $\boldsymbol{G}^z$ (e.g. $\boldsymbol{g}(\boldsymbol{z}) = \boldsymbol{W}\boldsymbol{z}$ where $\boldsymbol{W} \in \mathbb{R}^{d_z \times d_z}_{\boldsymbol{G}^z}$). The observational dynamics is then $\boldsymbol{\mu}(\boldsymbol{z}^t, \boldsymbol{a} = \boldsymbol{0}) = \boldsymbol{z}^{t-1} + \boldsymbol{g}(\boldsymbol{z}^{t-1})$ and the interventional settings correspond to zeroing out the elements of $\boldsymbol{g}(\boldsymbol{z}^{t-1})$ targeted by the intervention. Using Lemma 5.3, we can compute

$$\begin{aligned} &\Delta^{\epsilon=1}_{\boldsymbol{a},\ell} D_z q(\boldsymbol{z}^t \mid \boldsymbol{z}^{t-1}, \boldsymbol{a} = \boldsymbol{0}) \\ &= D_z q(\boldsymbol{z}^t \mid \boldsymbol{z}^{t-1}, \boldsymbol{a} = \boldsymbol{e}_\ell) - D_z q(\boldsymbol{z}^t \mid \boldsymbol{z}^{t-1}, \boldsymbol{a} = \boldsymbol{0}) = -\frac{1}{\sigma^2} \boldsymbol{G}^a_{\cdot,\ell} \odot \boldsymbol{g}(\boldsymbol{z}^{t-1}) \,. \end{aligned}$$

One can see that, as soon as the image of $\boldsymbol{g}$ spans $\mathbb{R}^{d_z}$, Assumption 5.7 is satisfied since we can choose values $\boldsymbol{z}_{(1)}, \ldots, \boldsymbol{z}_{(d_z)} \in \mathbb{R}^{d_z}$ such that $\operatorname{span}\{\boldsymbol{g}(\boldsymbol{z}_{(1)}), \ldots, \boldsymbol{g}(\boldsymbol{z}_{(d_z)})\} = \mathbb{R}^{d_z}$, which implies

$\operatorname{span}\{\boldsymbol{G}^a_{\cdot,\ell} \odot \boldsymbol{g}(\boldsymbol{z}_{(1)}), \dots, \boldsymbol{G}^a_{\cdot,\ell} \odot \boldsymbol{g}(\boldsymbol{z}_{(d_z)})\} = \mathbb{R}^{d_z}_{\mathbf{Ch}^a_\ell}$. *An example of transition function* $\boldsymbol{g}$ *satisfying this property is* $\boldsymbol{g}(\boldsymbol{z}) := \boldsymbol{W}\boldsymbol{z}$ *where* $\boldsymbol{W} \in \mathbb{R}^{d_z \times d_z}_{\boldsymbol{G}^z}$ *is invertible.*

Note that even if the temporal dependencies are not sparse, they are still helpful for identifiability as they make it more likely to satisfy the sufficient influence assumption (Assumption 5.7).

Example 5.12 (Grouped multi-target interventions for disentanglement without time)**.** *In this example, we assume there are no temporal dependencies* ($T = 1$) *and that the learner has access to* d_a *groups of interventions where the interventions belonging to the* ℓ*th group are known to target the same latent variables given by* $\boldsymbol{G}^a_{\cdot,\ell}$ *(these targets are unknown). Here is how this setting can be accommodated by our framework: given we have* d_a *groups of interventions where the* ℓ*th group contains* k_ℓ *interventions, we set* $\mathcal{A} := \{\mathbf{0}, 1\boldsymbol{e}_1, ..., k_1\boldsymbol{e}_1, 1\boldsymbol{e}_2, ..., k_2\boldsymbol{e}_2, ..., k_{d_a}\boldsymbol{e}_{d_a}\}$*. In this setting,* $\boldsymbol{a} = j\boldsymbol{e}_\ell$ *corresponds to the* j*th intervention of the* ℓ*th group. Moreover, the sufficient influence assumption requires that the interventions within a group* ℓ *span* $\mathbb{R}^{d_z}_{\mathbf{Ch}^a_\ell}$*. More precisely, we need* $\operatorname{span}\{\Delta^\epsilon_{a,\ell} D_z \log p(\boldsymbol{z} \mid \boldsymbol{a} = 0)\}_{\epsilon=1}^{k_\ell} = \mathbb{R}^{d_z}_{\mathbf{Ch}^a_\ell}$.

5.3.8.3. Temporal dependencies (Theorem 5.3). Finally, we provide an example (Example 5.13) where temporal dependencies alone (no auxiliary variable $\boldsymbol{a}$) is enough to disentangle. We start with an important remark about the sufficient influence assumption of Theorem 5.3.

Remark 5.7 (Auxiliary variables or non-Markovianity are required)**.** *An important observation is that, if the transition model does not have an auxiliary variable* $\boldsymbol{a}$ *and is Markovian, i.e.* $p(\boldsymbol{z}^t \mid \boldsymbol{z}^{<t}, \boldsymbol{a}^{<t}) = p(\boldsymbol{z}^t \mid \boldsymbol{z}^{t-1})$*, then Assumption 5.8 cannot be satisfied (except in trivial circumstances). To see this, simply note that, in that case,* $H^{t,t-1}_{z,z} q(\boldsymbol{z}^t \mid \boldsymbol{z}^{t-1})$ *depends only on* $\boldsymbol{z}^{t-1}$*, which is forced to be equal to* $\boldsymbol{z}^t$*. This means the span of the Hessian must be at most one-dimensional, which means that the assumption cannot hold as soon as* $||\boldsymbol{G}^z||_0 > 1$*. Therefore, when no auxiliary variable* $\boldsymbol{a}$ *is observed, Assumption 5.8 requires the transition model to be non-Markovian. In Example 5.13, we provide a concrete example of transition model without auxiliary variable* $\boldsymbol{a}$ *that satisfies this assumption. We will also see in Section 5.4 that if the transition model* $p(\boldsymbol{z}^t \mid \boldsymbol{z}^{t-1})$ *is in the exponential family, this assumption can be relaxed so that non-Markovianity is not required anymore.*

Example 5.13 (Sparse temporal dependencies for disentanglement without auxiliary variables)**.** *We continue with Examples 5.5 & 5.6 which were based on the graphs* $\boldsymbol{G}^z$ *depicted in Figures 5.3d & 5.3e, respectively. Assume that no action is observed, i.e. we can only leverage the sparsity of* $\boldsymbol{G}^z$ *to disentangle. Examples 5.5 & 5.6 already showed that these graph structures allow for complete disentanglement, as long as the sufficient influence of* $\boldsymbol{z}$ *assumption (Assumption 5.8) is satisfied. We now provide concrete transition models* $p(\boldsymbol{z}^t \mid \boldsymbol{z}^{<t})$ *that satisfies this requirement. Similarly to previous examples, assume* $p(\boldsymbol{z}^t \mid \boldsymbol{z}^{<t}) = \mathcal{N}(\boldsymbol{z}^t \mid \boldsymbol{\mu}(\boldsymbol{z}^{t-1}, \boldsymbol{z}^{t-2}), \sigma^2 \boldsymbol{I})$ *where*

$$\boldsymbol{\mu}(\boldsymbol{z}^{t-1}, \boldsymbol{z}^{t-2}) := \boldsymbol{z}^{t-1} + \boldsymbol{W}(\boldsymbol{z}^{t-2})\boldsymbol{z}^{t-1}\,, \tag{5.56}$$

where $\boldsymbol{W} : \mathbb{R}^{d_z} \to \mathbb{R}^{d_z \times d_z}_{\boldsymbol{G}^z}$ *is some function of* $\boldsymbol{z}^{t-2}$*. Using Lemma 5.3, we can derive*

$$H^{t,t-1}_{z,z} q(\boldsymbol{z}^t \mid \boldsymbol{z}^{<t}) = \frac{1}{\sigma^2}[\boldsymbol{I} + \boldsymbol{W}(\boldsymbol{z}^{t-2})]\,. \tag{5.57}$$

Thus Assumption 5.8 holds when there exists $\{\boldsymbol{z}^{t-2}_{(r)}\}_{r=1}^{||\boldsymbol{G}^z||_0}$ *such that*

$$\operatorname{span}\{\boldsymbol{I} + \boldsymbol{W}(\boldsymbol{z}^{t-2}_{(r)})\}_{r=1}^{||\boldsymbol{G}^z||_0} = \mathbb{R}^{d_z \times d_z}_{\boldsymbol{G}^z}\,. \tag{5.58}$$

One can directly see that, if $\boldsymbol{W}(\boldsymbol{z}^{t-2})$ *was actually constant in* $\boldsymbol{z}^{t-2}$*, the assumption could not hold (unless* $||\boldsymbol{G}^z||_0 \leq 1$*). This case would correspond to a simple linear model of the form* $\boldsymbol{\mu}(\boldsymbol{z}^{t-1}) := \boldsymbol{z}^{t-1} + \boldsymbol{W}\boldsymbol{z}^{t-1}$*. Our theory suggests this transition function is "too simple" to allow disentanglement.*

Nevertheless, we can find examples satisfying (5.58). For example, if $\boldsymbol{G}^z = \boldsymbol{I}$*, we can take*

$$\boldsymbol{W}(\boldsymbol{z}) = \begin{bmatrix} \boldsymbol{z}_1 & 0 & 0 \\ 0 & \boldsymbol{z}_2 & 0 \\ 0 & 0 & \boldsymbol{z}_3 \end{bmatrix} \tag{5.59}$$

and see that the family of functions $(1 + \boldsymbol{z}_1, 1 + \boldsymbol{z}_2, 1 + \boldsymbol{z}_3)$ *is linearly independent (when seen as functions from* $\mathbb{R}^3$ *to* $\mathbb{R}$*). By Lemma 5.5 in the appendix, this is equivalent to the existence of* $\boldsymbol{z}_{(1)}, \boldsymbol{z}_{(2)}, \boldsymbol{z}_{(3)} \in \mathbb{R}^{d_z}$ *such that (5.58) holds (see also Remark 5.5). In other words, the sufficient influence assumption holds. In the case where* $\boldsymbol{G}^z$ *is lower triangular like in Figure 5.3e, one can take*

$$\boldsymbol{W}(\boldsymbol{z}) = \begin{bmatrix} \boldsymbol{z}_1 & 0 & 0 \\ \boldsymbol{z}_2^2 & \boldsymbol{z}_2 & 0 \\ \boldsymbol{z}_3^3 & \boldsymbol{z}_3^2 & \boldsymbol{z}_3 \end{bmatrix} \tag{5.60}$$

and see that the family of functions $(1 + \boldsymbol{z}_1, 1 + \boldsymbol{z}_2, 1 + \boldsymbol{z}_3, \boldsymbol{z}_2^2, \boldsymbol{z}_3^2, \boldsymbol{z}_3^3)$ *are linearly independent, which similarly implies the existence of* $\boldsymbol{z}_{(1)}, \ldots, \boldsymbol{z}_{(6)} \in \mathbb{R}^{d_z}$ *such that (5.58) holds.*

Example 5.15 will show how one can leverage the exponential family assumption to allow for Markovianity even without auxiliary variables.

5.4. Partial disentanglement via mechanism sparsity in exponential families

The goal of this section is to understand how restricting the transition model to be in the *exponential family* allows us to weaken the sufficient influence assumption of Theorem 5.3. Section 5.4.1 introduces the exponential family assumption. Section 5.4.2 follows Khemakhem et al. [2020a] and shows that this additional assumption guarantees that the entanglement map $\boldsymbol{v}$ is "quasi-linear", which means $\boldsymbol{v}(\boldsymbol{z}) := \boldsymbol{s}^{-1}(\boldsymbol{L}\boldsymbol{s}(\boldsymbol{z}) + \boldsymbol{b})$, where $\boldsymbol{L}$ is a matrix and $\boldsymbol{s}$ is an element-wise invertible

function. Section 5.4.3 will introduce an identifiability result analogous to Theorem 5.3 for sparse $\hat{\boldsymbol{G}}^z$ that leverages the quasi-linearity of $\boldsymbol{v}$ to weaken Assumption 5.8 (sufficient influence of $\boldsymbol{z}$). We also briefly discuss an additional result from Appendix B.4 that shows connections between the nonparametric sufficient influence assumptions of this work (Assumptions 5.7 & 5.8) and their counterparts in Lachapelle et al. [2022] (Assumptions 5.11 & 5.12).

5.4.1. Exponential family latent transition models

We will assume that the conditional densities $p(\boldsymbol{z}_i^t \mid \boldsymbol{z}^{<t}, \boldsymbol{a}^{<t})$ are from an *exponential family* [Wainwright and Jordan, 2008]:

Assumption 5.9 (Exponential family transition model)**.** *For all $i \in [d_z]$, we have*

$$p(\boldsymbol{z}_i^t \mid \boldsymbol{z}^{<t}, \boldsymbol{a}^{<t}) = h_i(\boldsymbol{z}_i^t) \exp\{\boldsymbol{s}_i(\boldsymbol{z}_i^t)^\top \boldsymbol{\lambda}_i(\boldsymbol{z}^{<t}, \boldsymbol{a}^{<t}) - \psi_i(\boldsymbol{z}^{<t}, \boldsymbol{a}^{<t})\} \,. \tag{5.61}$$

Well-known distributions which belong to this family include the Gaussian and beta distribution. In the Gaussian case, the *sufficient statistic* is $\boldsymbol{s}_i(z) := (z, z^2)$ and the *base measure* is $h_i(z) := \frac{1}{\sqrt{2\pi}}$. The function $\boldsymbol{\lambda}_i(\boldsymbol{z}^{<t}, \boldsymbol{a}^{<t})$ outputs the *natural parameter* vector for the conditional distribution and can be itself parametrized, for instance, by a multi-layer perceptron (MLP) or a recurrent neural network (RNN). We will refer to the functions $\boldsymbol{\lambda}_i$ as the *mechanisms* or the *transition functions*. In the Gaussian case, the natural parameter is two-dimensional and is related to the usual parameters μ and σ^2 via the equation $(\lambda_1, \lambda_2) = (\frac{\mu}{\sigma^2}, -\frac{1}{2\sigma^2})$. We will denote by k the dimensionality of the natural parameter and that of the sufficient statistic (which are equal). Thus, $k = 2$ in the Gaussian case. The remaining term $\psi_i(\boldsymbol{z}^{<t}, \boldsymbol{a}^{<t})$ acts as a normalization constant.

We define $\boldsymbol{\lambda}(\boldsymbol{z}^{<t}, \boldsymbol{a}^{<t}) \in \mathbb{R}^{kd_z}$ to be the concatenation of all $\boldsymbol{\lambda}_i(\boldsymbol{z}^{<t}, \boldsymbol{a}^{<t})$ and similarly for $\boldsymbol{s}(\boldsymbol{z}^t) \in \mathbb{R}^{kd_z}$. Similarly to the nonparameteric case, the learnable parameters are $\boldsymbol{\theta} := (\boldsymbol{f}, \boldsymbol{\lambda}, \boldsymbol{G})$. Note that throughout, we assume that the sufficient statistic $\boldsymbol{s}$ is not learned and known in advance. With this notation, we can write the full transition model as

$$p(\boldsymbol{z}^t \mid \boldsymbol{z}^{<t}, \boldsymbol{a}^{<t}) = h(\boldsymbol{z}^t) \exp\{\boldsymbol{s}(\boldsymbol{z}^t)^\top \boldsymbol{\lambda}(\boldsymbol{z}^{<t}, \boldsymbol{a}^{<t}) - \psi(\boldsymbol{z}^{<t}, \boldsymbol{a}^{<t})\} \,, \tag{5.62}$$

where $h := \prod_{i=1}^{d_z} h_i$ and $\psi = \sum_{i=1}^{d_z} \psi_i$.

Remark 5.8 (Applying nonparametric identifiability results to exponential families)**.** *One can apply the nonparametric results (Theorems 5.1, 5.2 & 5.3) to models satisfying the exponential family assumption. In fact, all examples of Section 5.3.8 were Gaussians and thus are in the exponential family.*

5.4.2. Conditions for quasi-linear identifiability

In this section, we follow Khemakhem et al. [2020a] and show that the exponential family assumption combined with an additional sufficient variability assumption allows to go from identifiability up to diffeomorphism (Definition 5.5) to identifiability up to quasi-linearity, which we define next:

Definition 5.17 (Quasi-linear equivalence)**.** *We say two models $\boldsymbol{\theta} := (\boldsymbol{f}, \boldsymbol{\lambda}, \boldsymbol{G})$ and $\tilde{\boldsymbol{\theta}} := (\tilde{\boldsymbol{f}}, \boldsymbol{\lambda}, \tilde{\boldsymbol{G}})$ satisfying Assumptions 5.1, 5.2 & 5.9 are **equivalent up to quasi-linearity**, denoted $\boldsymbol{\theta} \sim_{\text{lin}} \tilde{\boldsymbol{\theta}}$, if and only if $\boldsymbol{\theta} \sim_{\text{diff}} \tilde{\boldsymbol{\theta}}$ and there exist an invertible matrix $\boldsymbol{L} \in \mathbb{R}^{kd_z \times kd_z}$ and a vector $\boldsymbol{b} \in \mathbb{R}^{kd_z}$ such that the map $\boldsymbol{v} := \boldsymbol{f}^{-1} \circ \tilde{\boldsymbol{f}}$ satisfies*

$$\boldsymbol{s}(\boldsymbol{v}(\boldsymbol{z})) = \boldsymbol{L}\boldsymbol{s}(\boldsymbol{z}) + \boldsymbol{b}, \ \forall \boldsymbol{z} \in \mathbb{R}^{d_z} . \tag{5.63}$$

If the sufficient statistic $\boldsymbol{s}$ is invertible, one obtains

$$\boldsymbol{v}(\boldsymbol{z}) = \boldsymbol{s}^{-1}(\boldsymbol{L}\boldsymbol{s}(\boldsymbol{z}) + \boldsymbol{b}), \ \forall \boldsymbol{z} \in \mathbb{R}^{d_z} . \tag{5.64}$$

Equation (5.64) is particularly interesting, as it says that the mapping relating both representations is "almost" linear in the following sense: although the map is not necessarily linear because the sufficient statistic $\boldsymbol{s}$ might not be, the "mixing" between components is linear. Indeed, notice that the sufficient statistic $\boldsymbol{s}$ and its inverse operates "element-wise". The mixing between components is only due to the matrix $\boldsymbol{L}$. This specific form simplifies a few steps in the identifiability proof, which might explain the popularity of this assumption in the literature on nonlinear ICA [Hyvarinen and Morioka, 2016, Khemakhem et al., 2020a,b, Hälvä and Hyvärinen, 2020, Morioka et al., 2021, Yang et al., 2021, Lachapelle et al., 2022, Liu et al., 2023, Xi and Bloem-Reddy, 2023].

The following theorem provides conditions to guarantee identifiability up to quasi-linearity. This is an adaptation and minor extension of Theorem 1 from Khemakhem et al. [2020a]. For completeness, we provide a proof in Appendix B.2.

Theorem 5.4 (Conditions for linear identifiability - Adapted from Khemakhem et al. [2020a])**.** *Let $\boldsymbol{\theta} := (\boldsymbol{f}, \boldsymbol{\lambda}, \boldsymbol{G})$ and $\hat{\boldsymbol{\theta}} := (\hat{\boldsymbol{f}}, \hat{\boldsymbol{\lambda}}, \hat{\boldsymbol{G}})$ be two models satisfying Assumptions 5.1, 5.2 & 5.9. Further assume that*

*(1) **[Observational equivalence]** $\boldsymbol{\theta} \sim_{\text{obs}} \hat{\boldsymbol{\theta}}$ (Definition 5.4);*

*(2) **[Minimal sufficient statistics]** For all i, the sufficient statistic $\boldsymbol{s}_i$ is minimal (see below).*

*(3) **[Sufficient variability]** The natural parameter $\boldsymbol{\lambda}$ varies "sufficiently" as formalized by Assumption 5.10 (see below).*

Then, $\boldsymbol{\theta} \sim_{\text{lin}} \hat{\boldsymbol{\theta}}$ (Def. 5.17).

The "minimal sufficient statistics" assumption is a standard one saying that $\boldsymbol{s}_i$ is defined appropriately to ensure that the parameters of the exponential family are identifiable (see e.g. Wainwright and Jordan [2008, p. 40]). See Definition 5.20 for a formal definition of minimality.

The last assumption is sometimes called the *assumption of variability* [Hyvärinen et al., 2019], and requires that the conditional distribution of $\boldsymbol{z}^t$ depends "sufficiently strongly" on $\boldsymbol{z}^{<t}$ and/or $\boldsymbol{a}^{<t}$. We stress the fact that this assumption concerns the ground-truth data generating model $\boldsymbol{\theta}$.

Assumption 5.10 (Sufficient variability in exponential families)**.** *There exist* $(\boldsymbol{z}_{(r)}, \boldsymbol{a}_{(r)})_{p=0}^{kd_z}$ *in their respective supports such that the* kd_z*-dimensional vectors* $(\boldsymbol{\lambda}(\boldsymbol{z}_{(r)}, \boldsymbol{a}_{(r)}) - \boldsymbol{\lambda}(\boldsymbol{z}_{(0)}, \boldsymbol{a}_{(0)}))_{r=1}^{kd_z}$ *are linearly independent.*

Notice that the $\boldsymbol{z}_{(r)}$ represent values of $\boldsymbol{z}^{<t}$ for potentially different values of t and can thus have different dimensions.

The following example builds on Example 5.10 and shows that the sufficient variability of the above theorem might hold or not. The first case is interesting since it guarantees that $\boldsymbol{v}$ is linear while the second is interesting because it showcases a situation where the theory of Khemakhem et al. [2020a] and Lachapelle et al. [2022] do not apply (since they both rely on the above theorem) thus highlighting the importance of our nonparametric extension.

Example 5.14 (Satisfying or not the sufficient variability assumption of Theorem 5.4)**.** *We recall Example 5.10 in which* $d_a = d_z$*,* $\boldsymbol{a} \in \mathcal{A} := \{\mathbf{0}, \boldsymbol{e}_1, \dots, \boldsymbol{e}_{d_a}\}$ *and* $\boldsymbol{G}^a = \boldsymbol{I}$ *without temporal dependencies: For all* $i \in [d_z]$*,* $p(\boldsymbol{z}_i \mid \boldsymbol{a}) = \mathcal{N}(\boldsymbol{z}; \boldsymbol{\mu}_i \boldsymbol{a}_i, 1 + \boldsymbol{\delta}_i \boldsymbol{a}_i)$ *where* $\boldsymbol{\mu}_i \in \mathbb{R}$ *and* $\boldsymbol{\delta}_i > -1$*. We consider the cases where* $\forall i,\ \boldsymbol{\delta}_i = 0$ *(unchanged variances) and* $\forall i,\ \boldsymbol{\delta}_i \neq 0$ *(variances change).*

If $\forall i,\ \boldsymbol{\delta}_i = 0$*, we can represent* $p(\boldsymbol{z}_i \mid \boldsymbol{a})$ *in its exponential form with a one-dimensional sufficient statistic given by* $\boldsymbol{s}_i(\boldsymbol{z}_i) = \boldsymbol{z}_i$ *and natural parameter given by* $\boldsymbol{\lambda}_i(\boldsymbol{a}) = \boldsymbol{\mu}_i \boldsymbol{a}_i$*. It can be easily seen that if* $\forall i,\ \boldsymbol{\mu}_i \neq 0$ *(i.e. the mean changes after the intervention), then the sufficient variability assumption of Theorem 5.4 holds since the vectors* $\boldsymbol{\lambda}(\boldsymbol{e}_i) - \boldsymbol{\lambda}(\mathbf{0}) = \boldsymbol{\mu}_i \boldsymbol{e}_i$ *do span* $\mathbb{R}^{d_z}$*.*

If $\forall i,\ \boldsymbol{\delta}_i \neq 0$*, we can represent* $p(\boldsymbol{z}_i \mid \boldsymbol{a})$ *in its exponential form with a two-dimensional sufficient statistics given by* $\boldsymbol{s}_i(\boldsymbol{z}_i) = (\boldsymbol{z}_i, \boldsymbol{z}_i^2)$ *and natural parameter given by* $\boldsymbol{\lambda}_i(\boldsymbol{a}) = \left(\frac{\boldsymbol{\mu}_i \boldsymbol{a}_i}{1+\boldsymbol{\delta}_i \boldsymbol{a}_i}, \frac{-1}{2(1+\boldsymbol{\delta}_i \boldsymbol{a}_i)}\right)$*. Note that, because we only have* d_z *interventions, for any choice of* $\boldsymbol{a}_{(0)} \in \mathcal{A}$*, the vectors* $\{\boldsymbol{\lambda}(\boldsymbol{a}) - \boldsymbol{\lambda}(\boldsymbol{a}_{(0)})\}_{\boldsymbol{a} \in \mathcal{A}}$ *can span at most a* d_z*-dimensional subspace, which is insufficient variability according to Theorem 5.4 since it requires spanning* $\mathbb{R}^{2d_z}$*.*

5.4.3. Partial disentanglement via sparse time dependencies in exponential families

We now provide a (partial) disentanglement guarantee which leverages sparsity regularization of $\hat{\boldsymbol{G}}^z$ and is specialized for exponential families with a one-dimensional sufficient statistic ($k = 1$). We will see that this extra parametric assumption on the transition model allows us to weaken the sufficient influence assumption of Theorem 5.3 (Assumption 5.8). In particular, this is going to allow for Markovian transitions without auxiliary variables, which was not allowed by the nonparametric result (Remark 5.7).

The sufficient influence of $\boldsymbol{z}$ assumption specialized to exponential families with $k = 1$ is directly taken from Lachapelle et al. [2022]:

Assumption 5.11 (Sufficient influence of $\boldsymbol{z}$ [Lachapelle et al., 2022])**.** *Assume $k = 1$ and $D\boldsymbol{s}(\boldsymbol{z})$ is invertible everywhere. There exist $\{(\boldsymbol{z}_{(r)}, \boldsymbol{a}_{(r)}, \tau_{(r)})\}_{r=1}^{||\boldsymbol{G}^z||_0}$ belonging to their respective support such that*

$$\operatorname{span}\left\{D_z^{\tau_{(r)}}\boldsymbol{\lambda}(\boldsymbol{z}_{(r)}, \boldsymbol{a}_{(r)})D\boldsymbol{s}(\boldsymbol{z}_{(r)}^{\tau_{(r)}})^{-1}\right\}_{r=1}^{||\boldsymbol{G}^z||_0} = \mathbb{R}_{\boldsymbol{G}^z}^{d_z \times d_z},$$

where $D_z^{\tau_{(r)}}\boldsymbol{\lambda}$ and $D\boldsymbol{s}$ are Jacobians with respect to $\boldsymbol{z}^{\tau_{(r)}}$ and $\boldsymbol{z}$, respectively.

In Appendix B.4, we show that the above assumption is implied by its nonparametric version (Assumption 5.8) when the transition model is in an exponential family with $k = 1$. However, Assumption 5.11 is *strictly weaker* than its nonparametric counterpart, Assumption 5.8. The reason is that, in the former, $\boldsymbol{z}^{\tau_{(r)}}$ can vary for different p whereas this is not allowed in the latter since we require $\boldsymbol{z} = \boldsymbol{z}^{\tau_{(r)}}$ for all r.

The following theorem, extended from Lachapelle et al. [2022], shows that making stronger parametric assumptions on the transition model allows to weaken the sufficient influence assumption. Note that its structure is nearly identical to Theorem 5.3. Its proof can be found in Appendix B.3.

Theorem 5.5 (Disentanglement via sparse temporal dependencies in exponential families)**.** *Let $\boldsymbol{\theta} := (\boldsymbol{f}, \boldsymbol{\lambda}, \boldsymbol{G})$ and $\hat{\boldsymbol{\theta}} := (\hat{\boldsymbol{f}}, \hat{\boldsymbol{\lambda}}, \hat{\boldsymbol{G}})$ be two models satisfying Assumptions 5.1, 5.2, 5.3, 5.4, 5.9 as well as all assumptions of Theorem 5.4. Further suppose that*

(1) The sufficient statistic $\boldsymbol{s}$ is d_z-dimensional ($k = 1$) and is a diffeomorphism from $\mathbb{R}^{d_z}$ to $\boldsymbol{s}(\mathbb{R}^{d_z})$;

*(2) **[Sufficient influence of z]** The Jacobian of the ground-truth transition function $\boldsymbol{\lambda}$ with respect to $\boldsymbol{z}$ varies "sufficiently", as formalized in Assumption 5.11;*

Then, there exists a permutation matrix $\boldsymbol{P}$ such that $\boldsymbol{P}\boldsymbol{G}^z\boldsymbol{P}^\top \subseteq \hat{\boldsymbol{G}}^z$. Further assume that

*(3) **[Sparsity regularization]** $||\hat{\boldsymbol{G}}^z||_0 \leq ||\boldsymbol{G}^z||_0$;*

Then, $\boldsymbol{\theta} \sim_{\text{con}}^{z} \hat{\boldsymbol{\theta}}$ (Def. 5.14) & $\boldsymbol{\theta} \sim_{\text{lin}} \hat{\boldsymbol{\theta}}$ (Def. 5.17), which together implies that

$$\boldsymbol{v}(\boldsymbol{z}) = \boldsymbol{s}^{-1}(\boldsymbol{C}\boldsymbol{P}^\top\boldsymbol{s}(\boldsymbol{z}) + \boldsymbol{b}),$$

where $\boldsymbol{b} \in \mathbb{R}^{d_z}$ and $\boldsymbol{C} \in \mathbb{R}^{d_z \times d_z}$ is invertible, $\boldsymbol{G}^z$- and $(\boldsymbol{G}^z)^\top$-preserving (Definition 5.11).

The reason we can simplify the sufficient influence assumption in the exponential family case has to do with the quasi-linear form of $\boldsymbol{v}$. Indeed, in that case, one can compute that the Jacobian of $\boldsymbol{v}$ takes a special form: $D\boldsymbol{v}(\boldsymbol{z}) = D\boldsymbol{s}(\boldsymbol{v}(\boldsymbol{z}))^{-1}\boldsymbol{L}D\boldsymbol{s}(\boldsymbol{z})$. Since $D\boldsymbol{s}$ is diagonal everywhere, one can see that the "non-diagonal part" of $D\boldsymbol{v}(\boldsymbol{z})$, i.e. $\boldsymbol{L}$, does not depend on $\boldsymbol{z}$, which simplifies the proof. See Appendix B.3 for details.

In Appendix D.2, we discuss how Khemakhem et al. [2020a] & Yao et al. [2022b] obtain disentanglement guarantee and how their assumptions differ from ours.

Example 5.15 (Markovian sparse temporal dependencies without auxiliary variables)**.** *Recall Remark 5.7 which pointed out that, without auxiliary variables, non-Markovianity was necessary to satisfy the nonparametric Assumption 5.8. We now illustrate that the analogous assumption specialized for exponential families with $k = 1$ (Assumption 5.11) is not as restrictive, i.e. it allows for Markovianity even when there are no auxiliary variables.*

We start from Example 5.6 which was based on the situation depicted in Figures 5.1 & 5.3e where the temporal graph $\boldsymbol{G}^z$ is lower triangular. Assume that no action is observed, i.e. we can only leverage the sparsity of $\boldsymbol{G}^z$ to disentangle. We now provide a concrete Markovian transition model $p(\boldsymbol{z}^t \mid \boldsymbol{z}^{t-1})$ that satisfies Assumption 5.11. Similarly to previous examples, assume $p(\boldsymbol{z}^t \mid \boldsymbol{z}^{t-1}) = \mathcal{N}(\boldsymbol{z}^t; \boldsymbol{\mu}(\boldsymbol{z}^{t-1}), \sigma^2 \boldsymbol{I})$ where

$$\boldsymbol{\mu}(\boldsymbol{z}) := \boldsymbol{z} + \begin{bmatrix} \boldsymbol{z}_1^2/2 \\ \boldsymbol{z}_1^3/3 \\ \boldsymbol{z}_1^4/4 \end{bmatrix} + \begin{bmatrix} 0 \\ \boldsymbol{z}_2^2/2 \\ \boldsymbol{z}_2^3/3 \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ \boldsymbol{z}_3^2/2 \end{bmatrix}. \tag{5.65}$$

Because the variance σ^2 is not influenced by $\boldsymbol{z}^{t-1}$, we can represent this transition model in an exponential family with $k = 1$ where the natural parameter is given by

$$\boldsymbol{\lambda}(\boldsymbol{z}^{t-1}) = \mu(\boldsymbol{z}^{t-1})/\sigma \tag{5.66}$$

and the sufficient statistic is given by $\boldsymbol{s}(\boldsymbol{z}) = \boldsymbol{z}/\sigma$. We can thus compute

$$D\boldsymbol{\lambda}(\boldsymbol{z}) D\boldsymbol{s}(\boldsymbol{z})^{-1} = \boldsymbol{I} + \begin{bmatrix} \boldsymbol{z}_1 & 0 & 0 \\ \boldsymbol{z}_1^2 & \boldsymbol{z}_2 & 0 \\ \boldsymbol{z}_1^3 & \boldsymbol{z}_2^2 & \boldsymbol{z}_3 \end{bmatrix} \tag{5.67}$$

which spans the 6-dimensional space $\mathbb{R}^{3\times 3}_{\boldsymbol{G}^z}$, as showed in Example 5.13.

Connecting with the sufficient influence assumption of $\boldsymbol{a}$ in Lachapelle et al. [2022]. The previous work of Lachapelle et al. [2022] could also leverage sparse influence of $\boldsymbol{a}$ to disentangle and was based on exponential family and sufficient influence assumptions. In Appendix B.4, Proposition 5.12 shows that their sufficient influence of $\boldsymbol{a}$ assumption is actually equivalent to our nonparametric version (Assumption 5.7) in the exponential family case with $k = 1$. An important conclusion of this observation is that the identifiability result via sparse $\boldsymbol{G}^a$ from Lachapelle et al. [2022], which was limited to the exponential family case with $k = 1$, can be derived from the more general nonparametric result of Theorem 5.2 we introduced earlier.

5.5. Model estimation with sparsity constraint

The identifiability results presented in this work are based on two crucial postulates: (i) the distribution over observations of both the learned and ground-truth models must match, i.e. $\hat{\boldsymbol{\theta}} \sim_{\text{obs}} \boldsymbol{\theta}$ (Definition 5.4), and (ii) the learned graphs $\hat{\boldsymbol{G}}^a$ and $\hat{\boldsymbol{G}}^z$ must be as sparse as their ground-truth

counterparts, respectively $\boldsymbol{G}^a$ and $\boldsymbol{G}^z$. The theory suggests that, in order to learn a (partially) disentangled representation, one should learn a model that satisfies these two requirements. In this section, we present one particular practical approach to achieve this approximately. Appendix C.2 provides further details.

Data fitting. The first condition can be achieved by fitting a model to data. Since the models discuss in this work present latent variable models, a natural idea is to use a maximum likelihood approach based on the well-known framework of variational autoencoders (VAEs) [Kingma and Welling, 2014] in which the decoder neural network corresponds to the mixing function $\hat{\boldsymbol{f}}$. We consider an approximate posterior of the form

$$q(\boldsymbol{z}^{\leq T} \mid \boldsymbol{x}^{\leq T}, \boldsymbol{a}^{<T}) := \prod_{t=1}^{T} q(\boldsymbol{z}^t \mid \boldsymbol{x}^t)\,, \tag{5.68}$$

where $q(\boldsymbol{z}^t \mid \boldsymbol{x}^t)$ is a Gaussian distribution with mean and diagonal covariance outputted by a neural network $\texttt{encoder}(\boldsymbol{x}^t)$. In our experiments, the latent model $\hat{p}(\boldsymbol{z}_i^t \mid \boldsymbol{z}^{<t}, \boldsymbol{a}^{<t})$ is a Gaussian distribution with mean $\hat{\boldsymbol{\mu}}_i(\boldsymbol{z}^{<t}, \boldsymbol{a}^{<t})$ parameterized as a fully connected neural network that "looks" only at a fixed window of s lagged latent variables.[5] Furthermore, the variances are learned but does not depend on $(\boldsymbol{z}^{<t}, \boldsymbol{a}^{<t})$ (see Appendix C.2 for details). This variational inference model induces the following evidence lower bound (ELBO) on $\log \hat{p}(\boldsymbol{x}^{\leq T} | \boldsymbol{a}^{<T})$:

$$\log \hat{p}(\boldsymbol{x}^{\leq T} | \boldsymbol{a}^{<T}) \geq \text{ELBO}(\hat{\boldsymbol{f}}, \hat{\boldsymbol{\mu}}, \hat{\boldsymbol{G}}, q; \boldsymbol{x}^{\leq T}, \boldsymbol{a}^{<T}) :=$$
$$\sum_{t=1}^{T} \mathop{\mathbb{E}}_{q(\boldsymbol{z}^t | \boldsymbol{x}^t)} [\log \hat{p}(\boldsymbol{x}^t \mid \boldsymbol{z}^t)] - \mathop{\mathbb{E}}_{q(\boldsymbol{z}^{<t} | \boldsymbol{x}^{<t})} KL(q(\boldsymbol{z}^t \mid \boldsymbol{x}^t) || \hat{p}(\boldsymbol{z}^t \mid \boldsymbol{z}^{<t}, \boldsymbol{a}^{<t}))\,. \tag{5.69}$$

We derive this fact in Appendix D.3. This lower bound can then be maximized using some variant of stochastic gradient ascent such as Adam [Kingma and Ba, 2015]. We note that many works have proposed learning dynamical models with latent variables using VAEs [Girin et al., 2020], with various choice of architectures and approximate posteriors. Our specific choices were made out of a desire for simplicity, but the reader should be aware of other possibilities.

The learned distribution will exactly match the ground truth distribution if (i) the model has enough capacity to express the ground-truth generative process, (ii) the approximate posterior has enough capacity to express the ground-truth posterior $p(\boldsymbol{z}^t | \boldsymbol{x}^{\leq T}, \boldsymbol{a}^{<T})$, (iii) the dataset is sufficiently large and (iv) the optimization finds the global optimum. If, in addition, the ground truth generative process satisfies the assumptions of Proposition 5.2, we can guarantee that the learned model $\hat{\boldsymbol{\theta}}$ will be equivalent to the ground truth model $\boldsymbol{\theta}$ up to diffeomorphism (Definition 5.5).

Learning $\hat{\boldsymbol{G}}$ with sparsity constraints. To go from equivalence up to diffeomorphism to actual disentanglement (partial or not), Theorems 5.1, 5.2, 5.3 & 5.5 suggest we should not only fit the

[5] The theory we developed would allow for a $\boldsymbol{\mu}$ function that depends on all previous time steps, not only the s previous ones. This could be achieved with a recurrent neural network or transformer, but we leave this to future work.

data, but also choose the learned graph $\hat{\boldsymbol{G}}$ such that $||\hat{\boldsymbol{G}}^a||_0 \leq ||\boldsymbol{G}^a||_0$ and/or $||\hat{\boldsymbol{G}}^z||_0 \leq ||\boldsymbol{G}^z||_0$. In order to allow for gradient-based optimization, our strategy consists in treating each edge $\hat{\boldsymbol{G}}_{i,j}$ as independent Bernoulli random variable with probability of success $\sigma(\boldsymbol{\gamma}_{i,j})$, where σ is the sigmoid function and $\boldsymbol{\gamma}_{i,j}$ is a parameter learned using the Gumbel-Softmax trick [Jang et al., 2017, Maddison et al., 2017]. Let $\text{ELBO}(\hat{\boldsymbol{f}}, \hat{\boldsymbol{\mu}}, \hat{\boldsymbol{G}}, q)$ be the ELBO objective of (5.69) averaged over the whole dataset. We tackle the following constrained optimization problem:

$$\max_{\hat{\boldsymbol{f}}, \hat{\boldsymbol{\mu}}, \boldsymbol{\gamma}, q} \mathbb{E}_{\hat{\boldsymbol{G}} \sim \sigma(\boldsymbol{\gamma})} \text{ELBO}(\hat{\boldsymbol{f}}, \hat{\boldsymbol{\mu}}, \hat{\boldsymbol{G}}, q) \;\; \text{subject to} \;\; \mathbb{E}_{\hat{\boldsymbol{G}} \sim \sigma(\boldsymbol{\gamma})} ||\hat{\boldsymbol{G}}||_0 \leq \beta \,. \tag{5.70}$$

where β is an hyperparameter (which should be set ideally to $\beta^* := ||\boldsymbol{G}||_0$, i.e. the number of edges in the ground-truth graph) and $\hat{\boldsymbol{G}} \sim \sigma(\boldsymbol{\gamma})$ means that $\hat{\boldsymbol{G}}_{i,j}$ are independent and distributed according to $\sigma(\boldsymbol{\gamma}_{i,j})$. Because $\mathbb{E}_{\hat{\boldsymbol{G}} \sim \sigma(\boldsymbol{\gamma})} ||\hat{\boldsymbol{G}}||_0 = ||\sigma(\boldsymbol{\gamma})||_1$ where $\sigma(\boldsymbol{\gamma})$ is matrix, the constraint becomes $||\sigma(\boldsymbol{\gamma})||_1 \leq \beta$. To solve this problem, we perform gradient descent-ascent on the Lagrangian function given by

$$\mathbb{E}_{\hat{\boldsymbol{G}} \sim \sigma(\boldsymbol{\gamma})} \text{ELBO}(\hat{\boldsymbol{f}}, \hat{\boldsymbol{\mu}}, \hat{\boldsymbol{G}}, q) - \alpha(||\sigma(\boldsymbol{\gamma})||_1 - \beta) \tag{5.71}$$

where the ascent step is performed w.r.t. $\hat{\boldsymbol{f}}, \hat{\boldsymbol{\mu}}, \hat{\boldsymbol{G}}$ and q; and the descent step is performed w.r.t. Lagrangian multiplier α, which is forced to remain greater or equal to zero via a simple projection step. As suggested by Gallego-Posada et al. [2021], we perform *dual restarts* which simply means that, as soon as the constraint is satisfied, the Lagrangian multiplier is reset to 0. We used the library `Cooper` [Gallego-Posada and Ramirez, 2022], which implements many constrained optimization procedure in Python, including the one described above. Note that we use Adam [Kingma and Ba, 2015] for the ascent steps and standard gradient descent for the descent step on the Lagrangian multiplier α.

We also found empircally that the following schedule for β is helpful: We start training with $\beta = \max_{\hat{\boldsymbol{G}}} ||\hat{\boldsymbol{G}}||_0$ and linearly decreasing its value until the desired number of edges is reached. This avoid getting a sparse graph too quickly while training, thus letting enough time to the model parameters to learn. In each experiment, we trained for 300K iterations, and the β takes 150K to go from its initial value to its desired value. We discuss how to select the hyperparameter β in Section 5.8.

5.6. Evaluation with R_{con} and SHD

In this section, we tackle the problem of evaluating quantitatively whether a learned representation $\hat{\boldsymbol{z}}$ is completely or partially disentangled w.r.t. the ground-truth representation $\boldsymbol{z}$, given a dataset of paired representations $\{(\boldsymbol{z}^i, \hat{\boldsymbol{z}}^i)\}_{i \in [N]}$. More precisely, want to evaluate whether two models are $\boldsymbol{a}$-consistent or $\boldsymbol{z}$-consistent (Definitions 5.13 & 5.14). To achieve this, we have to evaluate whether there exists a graph preserving map $\boldsymbol{c}$ (Definition 5.12) and a permutation matrix $\boldsymbol{P}$ such that for all

$i \in [N]$, $\boldsymbol{z}^i = \boldsymbol{c}(\boldsymbol{P}^\top \hat{\boldsymbol{z}}^i)$. For evaluation purposes, we assume we observe the ground-truth latent representation for each observation, i.e. we have $\{(\boldsymbol{x}^i, \boldsymbol{z}^i)\}_{i \in N}$ sampled i.i.d. from the ground-truth data generating process. We will take $\hat{\boldsymbol{z}}^i := \texttt{encoder}(\boldsymbol{x}^i)$ where `encoder` is from the learned VAE model introduced in Section 5.5. For simplicity, we assume that $\boldsymbol{c}$ is affine.[6]

We start with how to evaluate complete disentanglement. A popular choice for this is the *mean correlation coefficient* (MCC), which is obtained by first computing the Pearson correlation matrix $\boldsymbol{K} \in \mathbb{R}^{d_z \times d_z}$ between the ground-truth representation and the learned representation ($\boldsymbol{K}_{i,j}$ is the correlation between $\boldsymbol{z}_i$ and $\hat{\boldsymbol{z}}_j$). Then MCC $:= \max_{\boldsymbol{P} \in \text{permutations}} \frac{1}{d_z} \sum_{i=1}^{d_z} |(\boldsymbol{K}\boldsymbol{P})_{i,i}|$. We denote by $\hat{\boldsymbol{P}}$ the optimal permutation found by MCC.

To evaluate whether the learned representation is identified up to linear transformation (Definition 5.17), we perform linear regression to predict the ground-truth latent factors from the learned ones, and report the mean of the Pearson correlations between the predicted ground-truth latents and the actual ones. This metric is sometimes called the *coefficient of multiple correlation*, and happens to be the square root of the better known *coefficient of determination*, usually denoted by R^2. The advantage of using R instead of R^2 is that the former is comparable to MCC, and we always have MCC $\leq R$. Let us denote by $\hat{\boldsymbol{L}}$ the matrix of estimated coefficients, which should be thought of as an estimation of $\boldsymbol{L}$ in Definition 5.17 (assuming $\boldsymbol{s}(\boldsymbol{z}) = \boldsymbol{z}$, as is the case with Gaussian latents with fixed variance). Note that $\hat{\boldsymbol{L}}$ was fitted on standardized $\boldsymbol{z}$ and $\hat{\boldsymbol{z}}$ (shifted and scaled to have mean 0 and 1). This yields coefficients $\hat{\boldsymbol{L}}_{i,j}$ that are directly comparable without changing the value of the R score. We visualize $\hat{\boldsymbol{L}}$ in Figures 5.6 & 5.8.

To evaluate whether the learned representation is $\boldsymbol{a}$-consistent or $\boldsymbol{z}$-consistent to the ground-truth (Definitions 5.13 & 5.14), as predicted by Theorems 5.1 & 5.3, we introduce a novel metric, denoted by R_{con}. The idea behind R_{con} is to predict the ground-truth factors $\boldsymbol{z}$ from only the inferred factors $\hat{\boldsymbol{z}}$ that are allowed by the equivalence relations. For instance, for $\boldsymbol{a}$-consistency (Definition 5.13), the relation between $\boldsymbol{z}$ and $\hat{\boldsymbol{z}}$ is given by $\boldsymbol{z} = \boldsymbol{c}(\boldsymbol{P}^\top \hat{\boldsymbol{z}})$ where $\boldsymbol{c}$ is a $\boldsymbol{G}^a$-preserving diffeomorphism. Since we assume for simplicity that $\boldsymbol{c}$ is affine, we have $\boldsymbol{z} = \boldsymbol{C}\boldsymbol{P}^\top \hat{\boldsymbol{z}} + \boldsymbol{b}$ where $\boldsymbol{C}$ is a $\boldsymbol{G}^a$-preserving matrix. The idea is then to estimate both $\boldsymbol{P}$ and $\boldsymbol{C}$ using samples $(\boldsymbol{z}, \hat{\boldsymbol{z}})$. The permutation $\boldsymbol{P}$ is estimated by $\hat{\boldsymbol{P}}$, which was found when computing MCC (Section 5.8). To estimate $\boldsymbol{C}$, we compute $\hat{\boldsymbol{z}}_{\text{perm}} := \hat{\boldsymbol{P}}^\top \hat{\boldsymbol{z}}$ and then compute the mask $\boldsymbol{M} \in \{0,1\}^{d_z \times d_z}$ specifying which entries of $\boldsymbol{C}$ are allowed to be nonzero, as required by the $\boldsymbol{G}^a$-preservation property (Proposition 5.3). Then, for every i, we predict the ground-truth $\boldsymbol{z}_i$ by performing linear regression only on the *allowed* factors, i.e. $\boldsymbol{M}_{i,\cdot} \odot \hat{\boldsymbol{z}}_{\text{perm}}$, and compute the associated coefficient of multiple correlations $R_{\text{con},i}$ and report the mean, i.e. $R_{\text{con}} := \frac{1}{d_z} \sum_{i=1}^{d_z} R_{\text{con},i}$. It is easy to see that we must have $R_{\text{con}} \leq R$, since R_{con} was computed with less features than R. Moreover, MCC $\leq R_{\text{con}}$, because MCC can be

[6] This is not a simplification when the latent factors in the model and in the data-generating process are Gaussian with fixed variance and the assumptions of Theorem 5.4 hold. That is because the latent model is in the exponential family with sufficient statistic $\boldsymbol{s}(\boldsymbol{z}) = \boldsymbol{z}$ and, by Theorem 5.4, we must have that $\boldsymbol{z} = \boldsymbol{s}^{-1}(\boldsymbol{L}\boldsymbol{s}(\hat{\boldsymbol{z}}) + \boldsymbol{b}) = \boldsymbol{L}\hat{\boldsymbol{z}} + \boldsymbol{b}$.

thought of as computing exactly the same thing as for R_{con}, but by predicting $\boldsymbol{z}_i$ only from $\hat{\boldsymbol{z}}_{\text{perm},i}$, i.e. with less features than R_{con}. This means we always have $0 \leq \text{MCC} \leq R_{\text{con}} \leq R \leq 1$. This is a nice property which allows to compare all three metrics together and reflects the hierarchy between equivalence relations. Note that R_{con} depends implicitly on the ground-truth graph, since the matrix $\boldsymbol{M}$ indicating which entries of $\boldsymbol{C}$ are forced to be zero by the equivalence relation depends on $\boldsymbol{G}$.

To compare the learned graph $\hat{\boldsymbol{G}}$ to the ground-truth $\boldsymbol{G}$, we report the (normalized) *structural Hamming distance* (SHD) between the ground-truth graph and the estimated graph *permuted by* $\hat{\boldsymbol{P}}$. More precisely, we report SHD $= (||\boldsymbol{G}^a - \hat{\boldsymbol{P}}^\top \hat{\boldsymbol{G}}^a||_0 + ||\boldsymbol{G}^z - \hat{\boldsymbol{P}}^\top \hat{\boldsymbol{G}}^z \hat{\boldsymbol{P}}||_0)/(d_a d_z + d_z^2)$, where $\hat{\boldsymbol{P}}$ is the permutation found by MCC and $(d_a d_z + d_z^2)$ is the maximal number of edges $\boldsymbol{G}$ can have.

5.7. Related work

Linear and nonlinear ICA. The first results showing latent variables can be identified up to permutation and rescaling at least date back to classical linear ICA which assumes a linear mixing function $\boldsymbol{f}$ and mutually independent and non-Gaussian latent variables [Jutten and Herault, 1991, Tong et al., 1993, Comon, 1994]. Hyvärinen and Pajunen [1999] showed that when allowing $\boldsymbol{f}$ to be a general nonlinear transformation, a setting known as nonlinear ICA, mutual independence and non-Gaussianity alone are insufficient to identify the latent variables. This inspired multiple variations of nonlinear ICA that enabled identifiability by leveraging, e.g., nonstationarity [Hyvarinen and Morioka, 2016] and temporal dependencies [Hyvarinen and Morioka, 2017]. Hyvärinen et al. [2019] generalized these works by introducing a data generating process in which the latent variables are conditionally mutually independent given an observed *auxiliary variable* (corresponding to $\boldsymbol{a}$ in our work). These last three works rely on some form of *noise contrastive estimation* (NCE) [Gutmann and Hyvärinen, 2012], but similar identifiability results have also been shown for VAEs [Khemakhem et al., 2020a, Locatello et al., 2020a, Klindt et al., 2021], normalizing flows [Sorrenson et al., 2020] and energy-based models [Khemakhem et al., 2020b]. Kivva et al. [2022] showed that it is not necessary to observe the auxiliary variable to obtain disentanglement when the mixing function is piecewise affine and the latent factors are distributed according to a mixture of Gaussians with a diagonal covariance.

Causal representation learning (static). Since the publication of the first iteration of this work at CLeaR 2022, the field now known as *causal representation learning* (CRL) [Schölkopf et al., 2021] gained significant traction. The prototypical problem of CRL is similar to nonlinear ICA in that the goal is to identify latent factors of variations, but differs in that the latent variables are assumed to be related via a causal graphical model (CGM) and interventions on the latents are typically observed. While a few works assumed the causal graph structure is known [Kocaoglu et al., 2018, Shen et al., 2022, Nair et al., 2019, Liang et al., 2023], significant progress has been achieved recently in the setting where the latent causal graph is unknown and must be inferred from single-node

interventions targeting the latent variables [Ahuja et al., 2023, Squires et al., 2023, Buchholz et al., 2023, von Kügelgen et al., 2023, Zhang et al., 2023, Jiang and Aragam, 2023, Varici et al., 2023b,a]. In a similar spirit, Liu et al. [2023], Yang et al. [2021] leverage a form of nonstationarity that does not necessarily correspond to interventions and Bengio et al. [2020] suggests using adaptation speed as a heuristic objective to disentangle latent factors in the bivariate case, although without identifiability guarantees. The above works do not support temporal dependencies, unlike the framework presented in this work. While we do focus on temporal dependencies, the special case where $T = 1$ fleshed out in Examples 5.9, 5.10 & 5.12 can be categorized as static CRL with independent latent factors, i.e. empty latent causal graph. This approach has been applied to single-cell data with gene perturbations [Lopez et al., 2023, Bereket and Karaletsos, 2023]. Importantly, Example 5.12 illustrates how *multi-node* interventions on the latent factors can yield (partial) disentanglement in the independent factors regime. To the best of our knowledge, this constitutes the first identifiability guarantee from multi-node interventions with nonlinear mixing and should form an important step towards generalizing to arbitrary latent graphs. Note that Bing et al. [2023] recently proposed a disentanglement guarantee from multi-node interventions in the linear mixing setting.

CRL is closely related to methods that assume access to **paired observations** $(\boldsymbol{x}, \boldsymbol{x}')$ that are generated from a common decoder $\boldsymbol{f}$. These are in contrast with the works discussed above which assume the samples from observational and interventional distributions are **unpaired**. In the *paired* data regime, Locatello et al. [2020a] and Ahuja et al. [2022b] assume that only a small set of latent factors $S \subseteq [d_z]$ changes between $\boldsymbol{x}$ and $\boldsymbol{x}'$. Interestingly, Locatello et al. [2020a] assume that, for all i, $P(S \cap S' = \{i\}) > 0$ (for i.i.d S and S'), which resembles our graphical criterion for complete disentanglement (Definition 5.5). Karaletsos et al. [2016] proposed a related strategy based on triplets of observations and weak labels indicating which observation is closer to the reference in the (masked) latent space. Von Kügelgen et al. [2021] modelled the self-supervised setting with data augmentation using a similar idea and showed block-identifiability of the latent variables shared among $\boldsymbol{x}$ and $\boldsymbol{x}'$. Brehmer et al. [2022] assumes that the latent variables are sampled from a structural causal model (SCM) [Peters et al., 2017] and that $\boldsymbol{x}'$ is *counterfactual* in the sense that it is generated using the same SCM and exogenous noise values as $\boldsymbol{x}$ except for some noises which are modified randomly. Similar approaches can also provide identifiability guarantees in the *multi-view* setting where the decoders for the different views $\boldsymbol{x}$ and $\boldsymbol{x}'$ are allowed to be different [Gresele et al., 2020, Daunhawer et al., 2023]. The paired observations setting bears some similarity with the temporal setting covered in this work since the pairs $(\boldsymbol{x}^t, \boldsymbol{x}^{t-1})$ are observed jointly. However contrarily to the above works, Theorems 5.3 & 5.5 allow all latent variables to change between $t-1$ and t, only the temporal dependencies between them are assumed sparse. Morioka and Hyvarinen [2023] can also be seen as paired CRL in which the latents of different views can interact causally

in an restricted manner. Recently, Yao et al. [2023] generalized previous work by allowing more than two views.

Leveraging temporal dependencies or non-stationarity. Tong et al. [1990] proved identifiability of linear ICA when the latent factor $\boldsymbol{z}_i^t$ are correlated across time steps t but remain independent across components i, an idea that has been extended to nonlinear mixing [Hyvarinen and Morioka, 2017, Klindt et al., 2021, Schell and Oberhauser, 2023]. Using our notation, these works assume a diagonal adjacency matrix $\boldsymbol{G}^z$ which contrasts with Theorems 5.3 & 5.5 which allow for general $\boldsymbol{G}^z$ (although some graphs might not yield complete disentanglement). Yao et al. [2022a, Theorem 1] also allows for general $\boldsymbol{G}^z$, but do not rely on sparsity of $\boldsymbol{G}^z$ nor sparse interventions on the latent factors for identification. Instead, it relies on conditional independence of $\boldsymbol{z}_i^t$ given $\boldsymbol{z}^{t-1}$ and on a "sufficient variability" condition involving the third cross-derivatives $\frac{\partial^3}{(\partial \boldsymbol{z}_i^t)^2 \partial \boldsymbol{z}_j^{t-1}} \log p(\boldsymbol{z}_i^t \mid \boldsymbol{z}^{t-1})$ which excludes simple Gaussian models with homoscedastic variance like the ones we considered in Examples 5.8, 5.9, 5.11 and in our experiments of Section 5.8. General non-stationarity of the latent distribution, i.e. that are not sparse like the type of non-stationarity considered in this work, can also be used to identify the latent factors [Hyvarinen and Morioka, 2016, Hyvärinen et al., 2019, Khemakhem et al., 2020a, Hälvä and Hyvärinen, 2020, Morioka et al., 2021, Yao et al., 2022b,a], but these results require sufficient variability of higher-order derivatives/differences of the log-densities, which again typically exclude simple homoscedastic Gaussian models (see Appendix D.2 for more). Ahuja et al. [2022a] characterized the indeterminacies of the representation in dynamical latent models to be the set of equivariances of the transition mechanism. Apart from temporal dependencies, one can also consider latent factors structured according to a spatial topology [Hälvä et al., 2021].

Dynamical causal representation learning: The previous iteration of this work [Lachapelle et al., 2022] concurrently with Lippe et al. [2022] introduced latent variables identifiability guarantees for dynamical latent models based on sparse interventions. Lippe et al. [2023b] later proposed a generalization in which instantaneous causal connections are allowed. Key differences with the present work are (i) Lippe et al. [2023b] considers interventions with *known targets* while the present work (as well as Lachapelle et al. [2022]) consider interventions with *unknown targets*; (ii) Lachapelle et al. [2022, Theorem 5] and Theorems 5.3 & 5.5 do not need interventions to disentangle since they leverage sparsity of the temporal dependencies, contrarily to Lippe et al. [2023b]; (iii) Lippe et al. [2023b] allows for instantaneous causal connections, unlike the present work; and (iv) Lippe et al. [2023b] demonstrates their approach on image data. The concurrent work of Volodin [2021] independently proposed a very similar approach which also learns a sparse latent causal graph relating them together and to actions using binary masks, but focuses on testing various algorithmic variants and verifies empirically that the approach works on interactive environments rather than on formal identifiability guarantees. Lopez et al. [2023], Lei et al. [2023] found that

such models adapt to sparse interventions more quickly than their entangled counterparts. Keurti et al. [2023] discusses disentanglement in the temporal regimes through the lens of group theory but does not provide identifiability guarantees. Recently, Lippe et al. [2023a] proposed a model similar to ours with disentanglement guarantees based on the constraint that the effect of the variable $\boldsymbol{a}^{t-1}$ (analogous to R in their work) on each $\boldsymbol{z}_i^t$ is mediated by a deterministic binary variable.

Constraining the decoder function $\boldsymbol{f}$. It is worth noting that one can also obtain disentanglement guarantees by constraining the decoder function $\boldsymbol{f}$ in some way [Taleb and Jutten, 1999, Gresele et al., 2021, Buchholz et al., 2022, Leemann et al., 2023, Lachapelle et al., 2023b, Horan et al., 2021b]. In particular, this can be achieved by enforcing some form of sparsity on $\boldsymbol{f}$ [Moran et al., 2022, Zheng et al., 2022, Brady et al., 2023, Xi and Bloem-Reddy, 2023]. In contrast, the present work assumes only that $\boldsymbol{f}$ is a general diffeomorphism onto its image. Note that Zheng et al. [2022] reused many proof strategies of the shorter version of this work [Lachapelle et al., 2022].

Disentanglement with explicit supervision. Some works leverage more explicit supervision to disentangle. For example, Ahuja et al. [2022c] assumes labels are given by a linear transformation of mutually independent and non-Gaussian latent factors. Instead of relying on independence, Lachapelle et al. [2023a], Fumero et al. [2023] leverage the sparsity of the linear map to disentangle.

Other relevant works on sparsity. The assumption that high-level variables are sparsely related to one another and/or to actions was discussed by Bengio [2019], Goyal and Bengio [2021], Ke et al. [2021]. These ideas have been leveraged also by Goyal et al. [2021b,a], Madan et al. [2021] via attention mechanisms. Although these works are, in part, motivated by the same core assumption as ours, their focus is more on empirically verifying out-of-distribution generalization than it is on disentanglement (Definition 5.7) and formal identifiability results. The assumption that individual actions often affect only one factor of variation has been leveraged for disentanglement by Thomas et al. [2017]. Loosely speaking, the theory we developed in the present work can be seen as a formal justification for such approaches.

5.8. Experiments

To illustrate our identifiability results and the benefit of mechanism sparsity regularization for disentanglement, we apply the sparsity regularized VAE method of Section 5.5 on various synthetic datasets. Section 5.8.1 focuses on graphs satisfying the criterion of Assumption 5.5 which, as we saw, guarantees complete disentanglement. We also verify experimentally that the sufficient influence assumptions are indeed important for disentanglement and explore latent model with both homoscedastic and heteroscedastic variance. Section 5.8.2 explores graphs that do not satisfy the criterion. Details about our implementation are provided in Appendix C.2 and the code used to run these experiments can be found here: `https://github.com/slachapelle/disentanglement_via_mechanism_sparsity`.

Synthetic datasets. The datasets we considered are separated in two groups: *Action* & *Time* datasets. The former group has only auxiliary variables, which we interpret as actions, without temporal dependence, we thus fix $\hat{\boldsymbol{G}}^z = \mathbf{0}$. The latter group has only temporal dependence without actions, we thus fix $\hat{\boldsymbol{G}}^a = \mathbf{0}$. In each dataset, the ground-truth mixing function $\boldsymbol{f}$ is a randomly initialized neural network. The dimensionality of $\boldsymbol{z}$ and $\boldsymbol{x}$ are $d_z = 10$ and $d_x = 20$, respectively. In the action datasets, the dimensionality of $\boldsymbol{a}$ is $d_a = 10$, unless specified otherwise. The ground-truth transition model $p(\boldsymbol{z}^t \mid \boldsymbol{z}^{<t}, \boldsymbol{a}^{<t})$ is always a Gaussian with a mean outputted by some function $\boldsymbol{\mu}_{\boldsymbol{G}}(\boldsymbol{z}^{t-1}, a^{t-1})$ (the data is *Markovian*). For all datasets considered the covariance matrix is given by $\sigma_z^2 \boldsymbol{I}$, i.e. the variance if *homoscedastic*, except for the datasets ActionNonDiag$_{k=2}$ and TimeNonDiag$_{k=2}$ which have *heteroscedastic* variance. Appendix C.1 provides a more detailed descriptions of the datasets including the explicit form of $\boldsymbol{\mu}$ and $\boldsymbol{G}$ in each case. Note that the learned transition model $\hat{p}(\boldsymbol{z}^t \mid \boldsymbol{z}^{t-1}, \boldsymbol{a}^{t-1})$ is also an homoscedastic Gaussian where the mean function $\hat{\boldsymbol{\mu}}$ is an MLP.

Baselines. On the action datasets, we compare with TCVAE [Chen et al., 2018], iVAE [Khemakhem et al., 2020a]. Only iVAE leverages the action. On the temporal datasets, we compare our approach with TCVAE, PCL [Hyvarinen and Morioka, 2017] and SlowVAE [Klindt et al., 2021]. Only PCL and SlowVAE leverages the temporal dependencies. We also report the performance of a randomly initialized encoder (Random) and one trained via least-square regression directly on the ground-truth latent factors (Supervised). See Appendix C.3 for details on the baselines.

Unsupervised hyperparameter selection. In practice, the hyperparameters cannot be selected so as to optimize MCC, since this metric requires access to the ground-truth latent factors. Duan et al. [2020] introduced *unsupervised disentanglement ranking* (UDR) as a solution to unsupervised hyperparameter selection for disentanglement. Figures 5.5 & 5.7 shows the performance of all approaches using UDR to select the hyperparameter (when it has one). For our approach, we show a range of sparsity bounds β and indicate the hyperparameter selected by UDR with a black star. Note that, for our approach, we excluded from the UDR selection hyperparameters that yielded graphs with fewer edges than latent factors, as a heuristic to prevent UDR from selecting overly sparse graphs. Figures 5.5 & 5.7 show this *unsupervised* procedure selects a reasonable regularization level (as indicated by the black star), although not always the optimal one. See Appendix C.4 for details.

5.8.1. Graphs allowing complete disentanglement (satisfying Assumption 5.5)

Satisfying sufficient influence assumptions. Figure 5.5 reports the MCC and R scores of all methods on four datasets that satisfy both the graphical criterion and the sufficient influence assumption: the datasets ActionDiag and TimeDiag have "diagonal" graphs, i.e. $\boldsymbol{G}^a = \boldsymbol{I}$ and $\boldsymbol{G}^a = \boldsymbol{I}$, while ActionNonDiag and TimeNonDiag present more involved graphs (depicted in Figure 5.6).

Observations: We see that the sparsity constraint improves MCC and SHD on all datasets. Although most baselines obtain good R scores, which indicates their representation encodes all the information

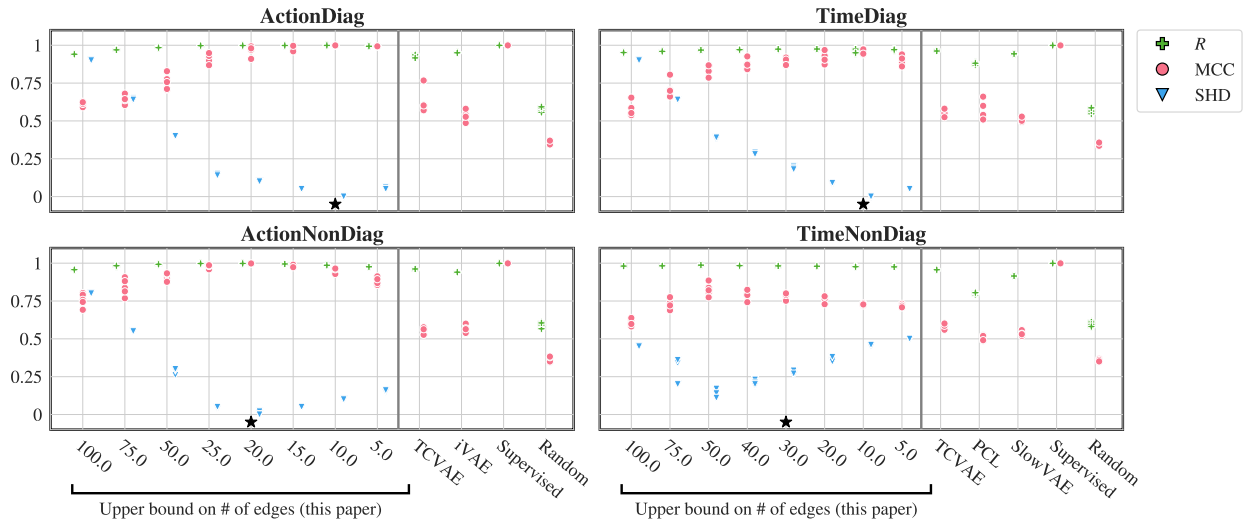
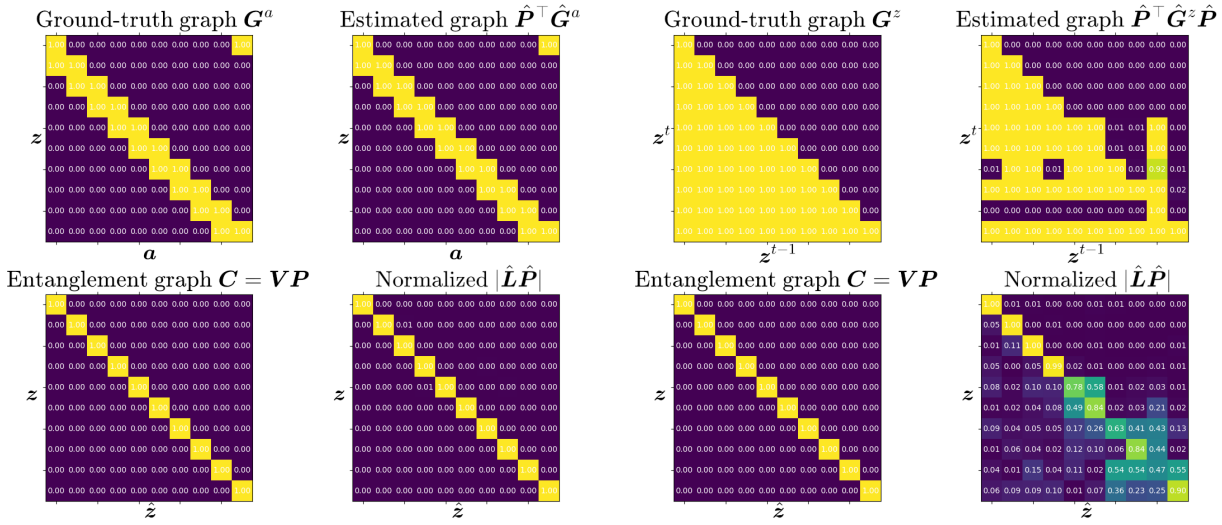


Figure 5.5. Graphical criterion holds: Datasets ActionDiag and TimeDiag have diagonal graphs while ActionNonDiag and TimeNonDiag have non-diagonal graphs. Sufficient influence is always satisfied. For our regularized VAE approach, we report performance for multiple sparsity levels β. In the left column, only $\hat{\boldsymbol{G}}^a$ is learned while in the right column, only $\hat{\boldsymbol{G}}^z$ is learned. For more details on the synthetic datasets, see Appendix C.1. The black star indicates which regularization parameter is selected by the filtered UDR procedure (see Appendix C.4). For R and MCC, higher is better. For SHD, lower is better. Performance is reported on 5 random seeds.

about the factors of variations, they obtain poor MCC in comparison to our approach with a properly selected sparsity level, which indicates they fail to disentangle. Moreover, the sparsity level selected by UDR (indicated by a black star) corresponds to the lowest SHD value for three out of four datasets and when it does not, it is still better than no sparsity at all. Figure 5.6 shows examples of estimated graphs. More details can be found in the caption.

Violating sufficient influence assumptions. The left column of Table 5.3 reports performance of all methods on the ActionNonDiag$_{\text{NoSuffInf}}$ and TimeNonDiag$_{\text{NoSuffInf}}$ datasets, which are essentially the same as ActionNonDiag and TimeNonDiag but do not satisfy the sufficient influence assumptions (see Appendix C.1 for details). **Observations:** For the ActionNonDiag$_{\text{NoSuffInf}}$ dataset, we still see an improvement in MCC and SHD when regularizing for sparsity, but not as important as for ActionNonDiag, which got MCC ≈ 1 and SHD ≈ 0. Still, our approach outperforms the baselines. For the TimeNonDiag$_{\text{NoSuffInf}}$ dataset, there is simply no improvement in MCC from sparsity regularization. In that case, SlowVAE (with hyperparameter selected to maximize MCC) and PCL have higher MCC. These observations confirm the importance of the sufficient influence assumptions.

Heteroscedastic variance ($k = 2$). The right column of Table 5.3 reports performance of all methods on the ActionNonDiag$_{k=2}$ and TimeNonDiag$_{k=2}$ datasets, which are essentially the same as ActionNonDiag and TimeNonDiag but presents heteroscedastic variance, i.e. var$(\boldsymbol{z}^t \mid \boldsymbol{z}^{t-1}, \boldsymbol{a}^{t-1})$ is not a



(a) ActionNonDiag dataset, $\beta = 20$ **(b)** TimeNonDiag dataset, $\beta = 50$

Figure 5.6. For each dataset, we visualize the median SHD run among the five randomly initialized runs of Figure 5.5 with the sparsity level β that is the closest to the ground-truth sparsity level $||\boldsymbol{G}||_0$. For each dataset, we visualize (i) the ground-truth graph, (ii) the permuted estimated graph, (iii) the entanglement graph predicted by our theory, and (iv) the permuted matrix of regression coefficients in absolute value normalized by the maximum coefficient i.e. $|\hat{\boldsymbol{L}}\hat{\boldsymbol{P}}|/\max_{i,j}|\hat{\boldsymbol{L}}_{i,j}|$. In Figure 5.6a, the estimated graph is exactly the ground-truth and $|\hat{\boldsymbol{L}}\hat{\boldsymbol{P}}|$ is perfectly diagonal, indicating complete disentanglement. In Figure 5.6b, the learned graph is close but not equal to the ground-truth. We can see that the off-diagonal nonzero values in $|\hat{\boldsymbol{L}}\hat{\boldsymbol{P}}|$ align with the poorly estimated parts of the graph.

constant function of $(\boldsymbol{z}^{t-1}, \boldsymbol{a}^{t-1})$. This setting is interesting since it is not covered by the exponential family theory of Lachapelle et al. [2022] which assumed a one-dimensional sufficient statistic $\boldsymbol{s}$ ($k = 1$) whereas here we have $k = 2$. Both datasets fall under the umbrella of our nonparametric theory. However, TimeNonDiag$_{k=2}$ cannot satisfy the sufficient influence assumption because the data is Markovian and does not present an auxiliary variable (see Remark 5.7). **Observations:** Both datasets benefit from sparsity and outperform the baselines. On ActionNonDiag$_{k=2}$ we obtain near perfect MCC and SHD while on TimeNonDiag$_{k=2}$ we obtain performance similar to TimeNonDiag. We hypothesize that the performance bottleneck in both TimeNonDiag and TimeNonDiag$_{k=2}$ is graph estimation, as in both cases SHD is always greater than $\approx 20\%$.

5.8.2. Graphs allowing only partial disentanglement (not satisfying Assumption 5.5)

In this section, we explore datasets with graphs that do not satisfy the criterion of Assumption 5.5. This means our theory can only guarantee a form a partial disentanglement. For this reason, we will

Datasets	ActionNonDiag$_{\text{NoSuffInf}}$			ActionNonDiag$_{k=2}$		
Metrics	SHD	MCC	R	SHD	MCC	R
iVAE	–	.61±.02	.97±.00	–	.59±.03	.94±.00
TCVAE (UDR)	–	.58±.03	.88±.01	–	.55±.02	.96±.00
TCVAE (MCC)	–	.61±.02	.96±.00	–	.55±.02	.96±.00
Ours (no sparsity)	.80±.00	.62±.02	.93±.00	.80±.00	.70±.03	.97±.00
Ours (sparsity)	.13±.03	.86±.04	1.0±.00	.03±.01	.98±.02	1.0±.00
Random	–	.37±.02	.63±.02	–	.37±.02	.60±.02
Supervised	–	1.0±.00	1.0±.00	–	1.0±.00	1.0±.00

Datasets	TimeNonDiag$_{\text{NoSuffInf}}$			TimeNonDiag$_{k=2}$		
Metrics	SHD	MCC	R	SHD	MCC	R
PCL	–	.66±.04	.96±.00	–	.58±.04	.83±.01
SlowVAE (UDR)	–	.59±.02	.98±.00	–	.57±.01	.93±.00
SlowVAE (MCC)	–	.71±.02	.98±.00	–	.58±.02	.95±.00
TCVAE (UDR)	–	.58±.03	.98±.00	–	.57±.01	.96±.00
TCVAE (MCC)	–	.58±.03	.98±.00	–	.57±.01	.96±.00
Ours (no sparsity)	.45±.00	.62±.04	.98±.00	.45±.00	.62±.01	.98±.00
Ours (sparsity)	.32±.05	.63±.03	.99±.00	.20±.07	.74±.04	.98±.00
Random	–	.40±.04	.67±.02	–	.36±.01	.59±.02
Supervised	–	1.0±.00	1.0±.00	–	1.0±.00	1.0±.00

Table 5.3. Datasets ActionNonDiag$_{\text{NoSuffInf}}$ and TimeNonDiag$_{\text{NoSuffInf}}$ do not satisfy their respective sufficient influence assumptions (Assumptions 5.6 & 5.11). Datasets ActionNonDiag$_{k=2}$ and TimeNonDiag$_{k=2}$ are such that var$(\boldsymbol{z}^t \mid \boldsymbol{z}^{t-1}, \boldsymbol{a}^{t-1})$ depends on $\boldsymbol{z}^{t-1}$ or $\boldsymbol{a}^{t-1}$ (which means the sufficient statistic has dimension $k = 2$, contrarily to all other datasets). For our method, we show performance both with and without the sparsity constraint. In the former case, the constraint is set to the number of edges in the ground-truth graph. For baselines that have hyperparameters, we report their performance with the hyperparameter configurations that maximize UDR and MCC.

report the R_{con} metric introduced in Section 5.6 which measures whether two representations are $\boldsymbol{a}$-consistent (Definition 5.13) or $\boldsymbol{z}$-consistent (Definition 5.14).

Satisfying sufficient influence assumptions. Figure 5.7 reports the MCC, R_{con} and R scores of all methods on four datasets that satisfy the sufficient influence assumption but not the graphical criterion: the datasets ActionBlockDiag and TimeBlockDiag have "block diagonal" graphs, while ActionBlockNonDiag and TimeBlockNonDiag have more intricate graphs (depicted in Figure 5.8). See Appendix C.1 for details about the datasets. **Observations:** In all four datasets, some sparsity level yields near perfect R_{con}, indicating the learned models are approximately $\boldsymbol{a}$-consistent or $\boldsymbol{z}$-consistent to the ground-truth. Morevover, SHD is correlated with R_{con}. Without surprise, MCC never comes close to one since complete disentanglement is not guaranteed by our theory. Analogously to Figure 5.5, the baselines have decent R values but very low MCC and R_{con}, indicating they cannot achieve partial disentanglement. Figure 5.8 shows examples of estimated

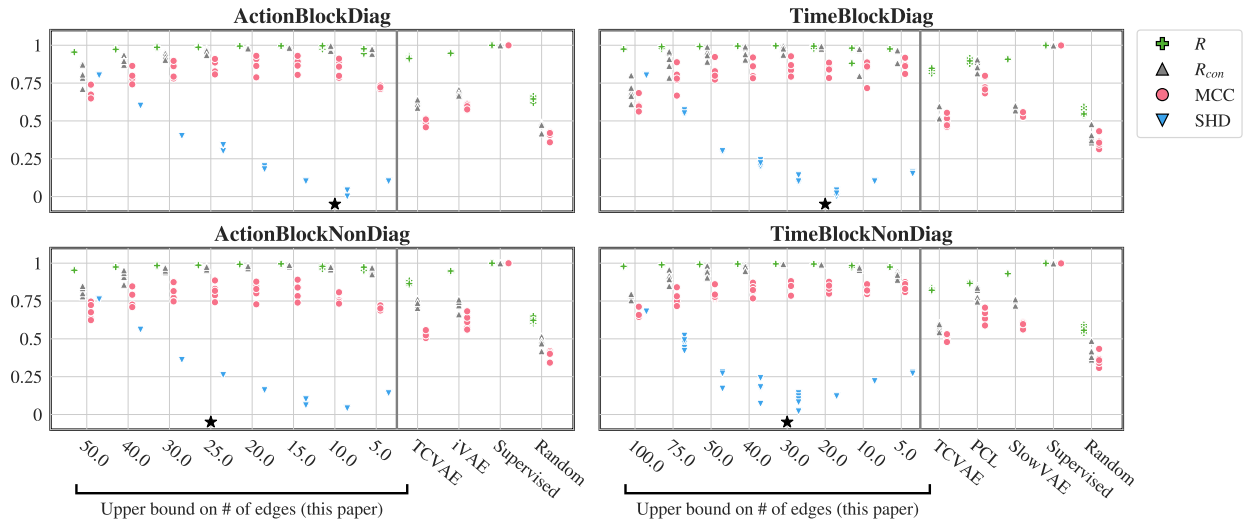
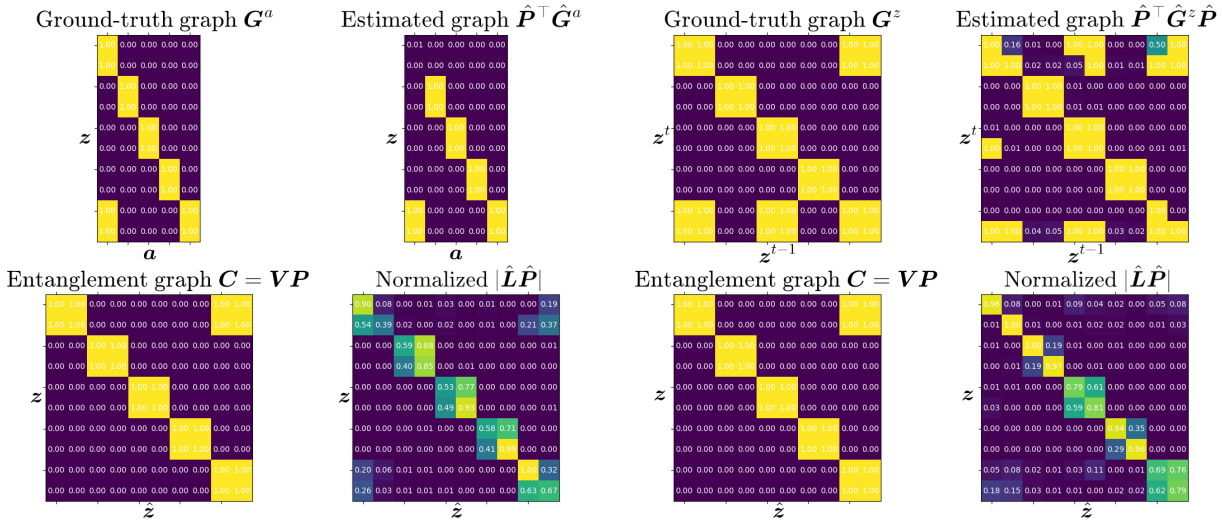


Figure 5.7. Graphical criterion does not hold: Datasets ActionBlockDiag and TimeBlockDiag have block-diagonal graphs while ActionBlockNonDiag and TimeBlockNonDiag have non-diagonal graphs. Sufficient influence is always satisfied. In the left column, only $\hat{\boldsymbol{G}}^a$ is learned and we vary β_a, and in the right column, only $\hat{\boldsymbol{G}}^z$ is learned and we vary β_z. For more details on the synthetic datasets, see Appendix C.1. The black star indicates which regularization parameter is selected by the filtered UDR procedure (see Appendix C.4). For R and MCC, higher is better. For SHD, lower is better. Performance is reported on 5 random seeds.

ActionRandomGraphs								
	Without sparsity			**With sparsity**				
p(edge)	MCC	R_{con}	R	MCC	R_{con}	R	SHD	$\mathbb{E}\|\|\boldsymbol{V}\|\|_0$
10%	.58±.13	.61±.11	.68±.13	.69±.14	.70±.12	.70±.12	.00±.00	45.0
20%	.67±.06	.69±.05	.83±.08	.85±.08	.86±.08	.86±.09	.01±.01	25.8
40%	.67±.03	.70±.03	.93±.04	.94±.05	.95±.05	.98±.04	.06±.05	15.8
60%	.69±.06	.73±.05	.96±.00	.88±.07	.91±.05	.99±.01	.14±.08	15.8
90%	.63±.04	.81±.08	.97±.00	.60±.01	.78±.07	.97±.00	.21±.07	45.0
TimeRandomGraphs								
	Without sparsity			**With sparsity**				
p(edge)	MCC	R_{con}	R	MCC	R_{con}	R	SHD	$\mathbb{E}\|\|\boldsymbol{V}\|\|_0$
10%	.66±.03	.66±.03	.98±.00	1.0±.00	1.0±.00	1.0±.00	.01±.02	10.2
20%	.63±.05	.63±.05	.98±.00	.99±.01	.99±.01	.99±.00	.08±.09	10.2
40%	.61±.02	.61±.02	.98±.00	.82±.16	.82±.16	.98±.01	.27±.13	10.2
60%	.58±.02	.58±.02	.98±.00	.71±.12	.71±.12	.98±.00	.33±.06	10.4
90%	.58±.03	.63±.08	.98±.00	.58±.02	.63±.08	.98±.00	.20±.07	26.1

Table 5.4. Experiments with randomly generated graphs. The probability of sampling an edge is p(edge). We report an estimation of $\mathbb{E}||\boldsymbol{V}||_0$ which is the average number of edges in the entanglement graph $\boldsymbol{V}$ entailed by the random ground-truth graph $\boldsymbol{G}$.

graph. When it comes to hyperparameter selection, UDR selects the hyperparameter with the lowest SHD on three out of four datasets, which indicates that UDR does reasonably well.



(a) ActionBlockNonDiag dataset, $\beta = 10$ **(b)** TimeBlockNonDiag dataset, $\beta = 30$

Figure 5.8. For each dataset, we visualize the median SHD run among the five randomly initialized runs of Figure 5.7 with the sparsity level β that is the closest to the ground-truth sparsity level $||\boldsymbol{G}||_0$. For each dataset, we visualize (i) the ground-truth graph, (ii) the permuted estimated graph, (iii) the entanglement graph predicted by our theory, and (iv) the permuted matrix of regression coefficients in absolute value normalized by the maximum coefficient i.e. $|\hat{\boldsymbol{L}}\hat{\boldsymbol{P}}|/\max_{i,j}|\hat{\boldsymbol{L}}_{i,j}|$. For both datasets, the learn graph is very close to the ground-truth. Furthermore, the match between the zero entries of $\hat{\boldsymbol{L}}\hat{\boldsymbol{P}}$ and those of the theoretical entanglement graph $\boldsymbol{C}$ is very good, although not perfect. Notice how certain blocks of latent factors remain entangled, as predicted by the theory.

Random graphs of varying sparsity levels. In Table 5.4, we consider the same $\boldsymbol{\mu}$ functions as in datasets ActionNonDiag and TimeNonDiag, but explore more diverse randomly generated ground-truth graphs with various degrees of sparsity. Edges are sampled i.i.d. with some probability $p(\text{edge})$. However note that, for TimeRandomGraphs dataset, the self-loops are presents with probability one. We report the performance of our approach both with and without sparsity regularization. When using sparsity, we set the β equal to the ground-truth number of edges $||\boldsymbol{G}||_0$. **Observations:** First, all datasets obtain an improvement in MCC and R_{con} from sparsity regularization, except for the very dense graphs with $p(\text{edge}) = 90\%$, in which case regularization does nothing or slightly degrades performance. Secondly, we can see that the SHD tends to be higher for larger graphs, suggesting these are harder to learn. Thirdly, in the ActionRandomGraphs datasets, we can see a negative correlation between MCC and $\mathbb{E}||\boldsymbol{V}||_0$, which is expected since $||\boldsymbol{V}||_0$ close to 10 means complete disentanglement is possible (assuming the graph is learned properly). This pattern also appears to some extent in the TimeRandomGraphs datasets. Notice how, among the TimeRandomGraphs datasets, all datasets sparser than $p(\text{edge}) = 90\%$ always have $\mathbb{E}||\boldsymbol{V}||_0 \approx 10$, indicating complete disentanglement should be possible.[7] This is confirmed by very high MCC, at least for sparser graphs which are learned properly. Finally, we note that the R score is low for the very sparse action

[7]We suspect this occurs because the self-loops which are present with probability one, unlike the Action dataset.

datasets. We suspect this is because very sparse graphs are less likely to satisfy the assumption of sufficient variability (Theorem 5.4) which guarantees quasi-linear equivalence (here it is actually *linear* equivalence, because of Gaussianity). Indeed, for very sparse graphs, some latent factors might end up without parents. This is not the case in the time datasets because of the self-loops which are always presents.

5.9. Conclusion

This work proposed a novel principle for disentanglement based on *mechanism sparsity regularization*. The idea is based on the assumption that the mechanisms that govern the dynamics of high-level concepts are often sparse: actions usually affect only a few entities and objects usually interact sparsely with each other. We provided novel nonparametric identifiability guarantees for this setting which gives sufficient conditions for disentanglement, whether complete or partial. Given the dependency structure between latent factors and auxiliary variables, our theory predicts the entanglement graph describing which estimated latent factors are expected to remain entangled. This constitutes a significant extension of the shorter version of this work [Lachapelle et al., 2022]. We further provide various examples to illustrates the consequences of our guarantees as well as the assumptions it relies on. For instance, we show that multi-node interventions with unknown targets fall under the umbrella of our framework. Finally, we demonstrate the theory experimentally by training a sparsity-constrained variational autoencoder on synthetic data, which allows us to explore various settings. Our work establishes a solid theoretical grounding for further empirical investigations in more realistic scenarios, such as single-cell data with gene perturbations [Lopez et al., 2023] and video [Lei et al., 2023]. Future works include relaxing assumptions such as conditional independence or considering more permissive settings such as "contextual sparsity", i.e., the assumption that objects only interact with each other in particular situations. We believe the latter could be formalized and leveraged for disentanglement using the tools developed in this work.

Appendices of Chapter 5

A. Identifiability theory - Nonparametric case

A.1. Useful Lemmas

Definition 5.18 (Regular closed set)**.** *A set $A \subseteq \mathbb{R}^n$ is regular closed when it is equal to the closure of its interior, i.e. $\overline{A^\circ} = A$.*

Lemma 5.4. *Let $A \subseteq \mathbb{R}^n$ and $\boldsymbol{f} : A \to \mathbb{R}^m$ be a C^k function. Then, its k first derivatives is uniquely defined on $\overline{A^\circ}$ in the sense that they do not depend on the specific choice of C^k extension.*

Proof Let $\boldsymbol{g} : U \to \mathbb{R}^n$ and $\boldsymbol{h} : V \to \mathbb{R}^n$ be two C^k extensions of $\boldsymbol{f}$ to $U \subseteq \mathbb{R}^n$ and $V \subseteq \mathbb{R}^n$ both open in $\mathbb{R}^n$. By definition,

$$\boldsymbol{g}(\boldsymbol{x}) = \boldsymbol{f}(\boldsymbol{x}) = \boldsymbol{h}(\boldsymbol{x}),\ \forall \boldsymbol{x} \in A\,. \tag{5.72}$$

The usual derivative is uniquely defined on the interior of the domain, so that

$$D\boldsymbol{g}(\boldsymbol{x}) = D\boldsymbol{f}(\boldsymbol{x}) = D\boldsymbol{h}(\boldsymbol{x}),\ \forall \boldsymbol{x} \in A^\circ\,. \tag{5.73}$$

Consider a point $\boldsymbol{x}_0 \in \overline{A^\circ}$. By definition of closure, there exists a sequence $\{\boldsymbol{x}_k\}_{k=1}^\infty \subseteq A^\circ$ s.t. $\lim_{k\to\infty} \boldsymbol{x}_k = \boldsymbol{x}_0$. We thus have that

$$\lim_{k\to\infty} D\boldsymbol{g}(\boldsymbol{x}_k) = \lim_{k\to\infty} D\boldsymbol{h}(\boldsymbol{x}_k) \tag{5.74}$$

$$D\boldsymbol{g}(\boldsymbol{x}_0) = D\boldsymbol{h}(\boldsymbol{x}_0)\,, \tag{5.75}$$

where we used the fact that the derivatives of $\boldsymbol{g}$ and $\boldsymbol{h}$ are continuous to go to the second line. Thus, all the C^k extensions of $\boldsymbol{f}$ must have equal derivatives on $\overline{A^\circ}$. This means we can unambiguously define the derivative of $\boldsymbol{f}$ everywhere on $\overline{A^\circ}$ to be equal to the derivative of one of its C^k extensions.

Since $\boldsymbol{f}$ is C^k, its derivative $D\boldsymbol{f}$ is C^{k-1}, we can thus apply the same argument to get that the second derivative of $\boldsymbol{f}$ is uniquely defined on $\overline{\overline{A^\circ}^\circ}$. It can be shown that $\overline{\overline{A^\circ}^\circ} = \overline{A^\circ}$. One can thus apply the same argument recursively to show that the first k derivatives of $\boldsymbol{f}$ are uniquely defined on $\overline{A^\circ}$. ■

		Calligraphic & indexing conventions
$[n]$	$:=$	$\{1, 2, \ldots, n\}$
x		Scalar (random or not, depending on context)
$\boldsymbol{x}$		Vector (random or not, depending on context)
$\boldsymbol{X}$		Matrix
$\mathcal{X}$		Set/Support
f		Scalar-valued function
$\boldsymbol{f}$		Vector-valued function
Df, $D\boldsymbol{f}$		Jacobian of f and $\boldsymbol{f}$
$D^2 f$		Hessian of f
$B \subseteq [n]$		Subset of indices
$\boldsymbol{x}_B$		Vector formed with the ith coordinates of $\boldsymbol{x}$, for all $i \in B$
$\boldsymbol{X}_{B,B'}$		Matrix formed with the entries $(i, j) \in B \times B'$ of $\boldsymbol{X}$.
		Recurrent notation
$\boldsymbol{x}^t \in \mathbb{R}^{d_x}$		Observation at time t
$\boldsymbol{x}^{\leq t} \in \mathbb{R}^{d_x \times t}$		Matrix of observations at times $1, \ldots, t$
$\boldsymbol{z}^t \in \mathbb{R}^{d_z}$		Vector of latent factors of variations at time t
$\boldsymbol{z}^{\leq t} \in \mathbb{R}^{d_z \times t}$		Matrix of latent vectors at times $1, \ldots, t$
$\boldsymbol{a}^t \in \mathbb{R}^{d_a}$		Vector of auxiliary variables at time t
$\boldsymbol{a}^{<t} \in \mathbb{R}^{d_a \times t}$		Matrix of auxiliary vectors at times $0, 1 \ldots, t-1$
$\mathcal{A} \subseteq \mathbb{R}^{d_a}$		Support of $\boldsymbol{a}^t$
$\boldsymbol{f} : \mathbb{R}^{d_z} \to \mathbb{R}^{d_x}$		Ground-truth decoder function
$\hat{\boldsymbol{f}} : \mathbb{R}^{d_z} \to \mathbb{R}^{d_x}$		Learned decoder function
$p(\boldsymbol{z}^t \mid \boldsymbol{z}^{<t}, \boldsymbol{a}^{<t})$		Ground-truth latent transition model
$\hat{p}(\boldsymbol{z}^t \mid \boldsymbol{z}^{<t}, \boldsymbol{a}^{<t})$		Learned latent transition model
$\boldsymbol{G}^a \in \{0, 1\}^{d_z \times d_a}$		Ground-truth adjacency matrix of graph connecting $\boldsymbol{a}^{<t}$ to $\boldsymbol{z}^t$
$\boldsymbol{G}^z \in \{0, 1\}^{d_z \times d_z}$		Ground-truth adjacency matrix of graph connecting $\boldsymbol{z}^{<t}$ to $\boldsymbol{z}^t$
$\hat{\boldsymbol{G}}^a, \hat{\boldsymbol{G}}^z$		Learned adjacency matrices
$\mathbf{Pa}_i^a \subseteq [d_a]$		Parents of $\boldsymbol{z}_i^t$ in $\boldsymbol{G}^a$
$\mathbf{Ch}_\ell^a \subseteq [d_z]$		Children of $\boldsymbol{a}_\ell^t$ in $\boldsymbol{G}^z$
$\mathbf{Pa}_i^z \subseteq [d_z]$		Parents of $\boldsymbol{z}_i^t$ in $\boldsymbol{G}^z$
$\mathbf{Ch}_i^z \subseteq [d_z]$		Children of $\boldsymbol{z}_i^{t-1}$ in $\boldsymbol{G}^z$
$D_z^t \log p \in \mathbb{R}^{1 \times d_z}$		Jacobian vector of $\log p(\boldsymbol{z}^t \mid \boldsymbol{z}^{<t}, \boldsymbol{a}^{<t})$ w.r.t. $\boldsymbol{z}^t$
$H_{z,a}^{t,\tau} \log p \in \mathbb{R}^{d_z \times d_a}$		Hessian matrix of $\log p(\boldsymbol{z}^t \mid \boldsymbol{z}^{<t}, \boldsymbol{a}^{<t})$ w.r.t. $\boldsymbol{z}^t$ and $\boldsymbol{a}^\tau$
$H_{z,z}^{t,\tau} \log p \in \mathbb{R}^{d_z \times d_z}$		Hessian matrix of $\log p(\boldsymbol{z}^t \mid \boldsymbol{z}^{<t}, \boldsymbol{a}^{<t})$ w.r.t. $\boldsymbol{z}^t$ and $\boldsymbol{z}^\tau$
$\sigma : [d_z] \to [d_z]$		A permutation
		Topology
$\overline{\mathcal{X}}$		Closure of the set $\mathcal{X} \subseteq \mathbb{R}^n$
$\mathcal{X}^\circ$		Interior of the set $\mathcal{X} \subseteq \mathbb{R}^n$

Table 5.5. Table of Notations.

Lemma 5.5. *Let X be some set. A family of functions $(f_i : X \to \mathbb{R})_{i=1}^n$ is linearly independent if and only if there exists $x_1, ..., x_n \in X$ such that the family of vectors $((f_1(x_i), ..., f_n(x_i)))_{i=1}^n$ is linearly independent.*

Proof We start by proving the "if" part. Assume the functions are linearly dependent. Then there exists $\boldsymbol{\alpha} \neq 0$ such that, for all $x \in X$, $\sum_{i=1}^n \boldsymbol{\alpha}_i f_i(x) = 0$. Choose distinct $x_1, ..., x_n \in X$. We have thus have that for all $j \in [n]$, $\sum_{i=1}^n \boldsymbol{\alpha}_i f_i(x_j) = 0$. This can be written in matrix form:

$$\begin{bmatrix} f_1(x_1) & \cdots & f_1(x_n) \\ \vdots & \ddots & \vdots \\ f_n(x_1) & \cdots & f_n(x_n) \end{bmatrix} \boldsymbol{\alpha} = \mathbf{0}\,, \tag{5.76}$$

which implies that the columns are linearly dependent.

We now show the "only if" part. Suppose that for all $\{x_1, \ldots, x_n\} \subseteq X$, the family of vectors $((f_1(x_i), ..., f_n(x_i)))_{i=1}^n$ is linearly dependent. This means that the set $U = \text{span}\{(f_1(x), ..., f_n(x)) \mid x \in X\}$ is a proper linear subspace of $\mathbb{R}^n$. This means that there is a nonzero $u \in U^\perp$, the orthogonal complement of U. By definition, u is orthogonal to all elements in $\{(f_1(x), ..., f_n(x)) \mid x \in X\}$. In other words, for all $x \in X$, $\sum_{i=1}^n u_i f_i(x) = 0$. Hence the $\boldsymbol{f}_i$ are linearly dependent. ■

A.2. Proof of Proposition 5.1

Proposition 5.1 (Linking dependency graph and Jacobian)**.** *Let $\boldsymbol{h}$ be a C^1 function, i.e. continuously differentiable, from $\mathbb{R}^n$ to $\mathbb{R}^m$ and let $\boldsymbol{H}$ be its dependency graph (Definition 5.2). Then,*

$$\boldsymbol{H}_{i,j} = 0 \iff \textit{For all } \boldsymbol{a} \in \mathbb{R}^n\,, D\boldsymbol{h}(\boldsymbol{a})_{i,j} = 0\,. \tag{5.7}$$

Proof The " $\Longrightarrow$ " direction holds because since we can simply differentiate $\boldsymbol{h}_i(\boldsymbol{a}) = \bar{\boldsymbol{h}}_i(\boldsymbol{a}_{-j})$ w.r.t. $\boldsymbol{a}_j$ to get zero.

We now show the " $\Longleftarrow$ " direction. Suppose that for all $\boldsymbol{a} \in \mathbb{R}^n$, $D\boldsymbol{h}(\boldsymbol{a})_{i,j} = 0$. We must now show that $\boldsymbol{h}_i(\boldsymbol{a})$ is constant in $\boldsymbol{a}_j$ for all $\boldsymbol{a}_{-j}$. Choose any $\boldsymbol{a}^0, \boldsymbol{a}^1 \in \mathbb{R}^n$ such that $\boldsymbol{a}^0_{-j} = \boldsymbol{a}^1_{-j}$. Thanks to the fundamental theorem of calculus, we can write

$$\boldsymbol{h}_i(\boldsymbol{a}^1) - \boldsymbol{h}_i(\boldsymbol{a}^0) = \int_{[0,1]} \frac{d}{d\alpha} \boldsymbol{h}_i((1-\alpha)\boldsymbol{a}^0 + \alpha \boldsymbol{a}^1)) d\alpha \tag{5.77}$$

$$= \int_{[0,1]} \underbrace{D\boldsymbol{h}((1-\alpha)\boldsymbol{a}^0 + \alpha\boldsymbol{a}^1)_{i,\cdot}}_{\text{zero at } j} \cdot \underbrace{(\boldsymbol{a}^1 - \boldsymbol{a}^0)}_{\text{zero except at } j} d\alpha \tag{5.78}$$

$$= 0\,. \tag{5.79}$$

Since $\boldsymbol{a}^0$ and $\boldsymbol{a}^1$ were arbitrary points such that $\boldsymbol{a}^0_{-j} = \boldsymbol{a}^1_{-j}$, this means the function $\boldsymbol{h}_i(\boldsymbol{a})$ is constant in $\boldsymbol{a}_j$ for all values of $\boldsymbol{a}_{-j}$. ■

A.3. Proof of Proposition 5.2

In this section, we prove Proposition 5.2. Before doing so, we first recall the definition of the support of a random variable (Definition 5.19) and prove a useful lemma (Lemma 5.6).

Definition 5.19. *(Support of a random variable) Let $\boldsymbol{x}$ be a random variable with values in $\mathbb{R}^n$ with distribution $\mathbb{P}_{\boldsymbol{x}}$. Let $\mathcal{O}_n$ be the standard topology of $\mathbb{R}^n$ (i.e. the set of open sets of $\mathbb{R}^n$). The support of $\boldsymbol{x}$ is defined as*

$$\text{supp}(\boldsymbol{x}) := \{\boldsymbol{x} \in \mathbb{R}^n \mid \boldsymbol{x} \in O \in \mathcal{O}_n \implies \mathbb{P}_{\boldsymbol{x}}(O) > 0\}\,. \tag{5.80}$$

Lemma 5.6. *Let $\boldsymbol{z}$ be a random variable with values in $\mathbb{R}^m$ with distribution $\mathbb{P}_{\boldsymbol{z}}$ and $\boldsymbol{y} := \boldsymbol{f}(\boldsymbol{z})$ where $\boldsymbol{f} : \text{supp}(\boldsymbol{z}) \to \mathbb{R}^n$ is a homeomorphism onto its image. Then*

$$\boldsymbol{f}(\text{supp}(\boldsymbol{z})) \subseteq \text{supp}(\boldsymbol{y}) \subseteq \overline{\boldsymbol{f}(\text{supp}(\boldsymbol{z}))}\,. \tag{5.81}$$

where the closure is taken w.r.t. to the topology of $\mathbb{R}^n$.

Proof We first prove that $f(\text{supp}(\boldsymbol{z})) \subseteq \text{supp}(\boldsymbol{y})$. Let $\boldsymbol{y}^0 \in \boldsymbol{f}(\text{supp}(\boldsymbol{z}))$ and N be an open neighborhood of $\boldsymbol{y}^0$, i.e. $\boldsymbol{y}^0 \in N \in \mathcal{O}_n$. Note that there exists $\boldsymbol{z}^0 \in \text{supp}(\boldsymbol{z})$ such that $\boldsymbol{f}(\boldsymbol{z}^0) = \boldsymbol{y}^0$. Note that $\boldsymbol{z}^0 \in \boldsymbol{f}^{-1}(\{\boldsymbol{y}^0\}) \subseteq \boldsymbol{f}^{-1}(N)$ and that, by continuity of $\boldsymbol{f}$, $\boldsymbol{f}^{-1}(N)$ is an open neighborhood of $\boldsymbol{z}^0$. Since $\boldsymbol{z}^0 \in \text{supp}(\boldsymbol{z})$, we have

$$\begin{aligned} 0 &< \mathbb{P}_{\boldsymbol{z}}(f^{-1}(N)) && (5.82)\\ &= \mathbb{P}_{\boldsymbol{z}} \circ f^{-1}(N) && (5.83)\\ &= \mathbb{P}_{\boldsymbol{y}}(N)\,. && (5.84) \end{aligned}$$

Hence $\boldsymbol{y}^0 \in \text{supp}(\boldsymbol{y})$, which concludes the "$\subseteq$" part.

We now prove the other inclusion. Let $\boldsymbol{y}^0 \in \text{supp}(\boldsymbol{y})$ and suppose, by contradiction, that $\boldsymbol{y}^0 \notin \overline{\boldsymbol{f}(\text{supp}(\boldsymbol{z}))}$. Since $\overline{\boldsymbol{f}(\text{supp}(\boldsymbol{z}))}$ is closed in $\mathbb{R}^n$, there exists N s.t. $\boldsymbol{y}^0 \in N \in \mathcal{O}_n$ with $N \cap \overline{\boldsymbol{f}(\text{supp}(\boldsymbol{z}))} = \emptyset$. Since $\boldsymbol{y}^0 \in \text{supp}(\boldsymbol{y})$,

$$\begin{aligned} 0 &< \mathbb{P}_{\boldsymbol{y}}(N) && (5.85)\\ &= \mathbb{P}_{\boldsymbol{z}}(\boldsymbol{f}^{-1}(N)) && (5.86)\\ &= \mathbb{P}_{\boldsymbol{z}}(\emptyset) = 0\,. && (5.87) \end{aligned}$$

The above contradiction implies that $\boldsymbol{y}^0 \in \overline{\boldsymbol{f}(\text{supp}(\boldsymbol{z}))}$. ■

Proposition 5.2 (Identifiability up to diffeomorphism)**.** *Let* $\boldsymbol{\theta} := (\boldsymbol{f}, p, \boldsymbol{G})$ *and* $\hat{\boldsymbol{\theta}} := (\hat{\boldsymbol{f}}, \hat{p}, \hat{\boldsymbol{G}})$ *be two models satisfying Assumption 5.1. If* $\boldsymbol{\theta} \sim_{\text{obs}} \hat{\boldsymbol{\theta}}$ *(Def. 5.4), then* $\boldsymbol{\theta} \sim_{\text{diff}} \hat{\boldsymbol{\theta}}$ *(Def. 5.5).*

Proof

Equality of Denoised Distributions. Given an arbitrary $\boldsymbol{a}^{<T} \in \mathcal{A}^T$ and a parameter $\boldsymbol{\theta} = (\boldsymbol{f}, p, \boldsymbol{G})$, let $\mathbb{P}_{\boldsymbol{x}^{\leq T}|\boldsymbol{a}^{<T};\boldsymbol{\theta}}$ be the conditional probability distribution of $\boldsymbol{x}^{\leq T}$, let $\mathbb{P}_{\boldsymbol{z}^{\leq T}|\boldsymbol{a}^{<T};\boldsymbol{\theta}}$ be the conditional probability distribution of $\boldsymbol{z}^{\leq T}$ and let $\mathbb{P}_{\boldsymbol{n}^{\leq T}}$ be the probability distribution of $\boldsymbol{n}^{\leq T}$ (the Gaussian noises added on $\boldsymbol{f}(\boldsymbol{z}^{\leq T})$, defined in Sec. 5.2.1). Let $\boldsymbol{y}^t := \boldsymbol{f}(\boldsymbol{z}^t)$ and $\mathbb{P}_{\boldsymbol{y}^{\leq T}|\boldsymbol{a}^{<T};\boldsymbol{\theta}}$ be its conditional probability distribution. First, notice that

$$\mathbb{P}_{\boldsymbol{x}^{\leq T}|\boldsymbol{a}^{<T};\boldsymbol{\theta}} = \mathbb{P}_{\boldsymbol{y}^{\leq T}|\boldsymbol{a}^{<T};\boldsymbol{\theta}} * \mathbb{P}_{\boldsymbol{n}^{\leq T}}\,, \tag{5.88}$$

where $*$ is the convolution operator between two measures. We now show that if two models agree on the observations, i.e. $\mathbb{P}_{\boldsymbol{x}^{\leq T}|\boldsymbol{a}^{<T};\boldsymbol{\theta}} = \mathbb{P}_{\boldsymbol{x}^{\leq T}|\boldsymbol{a}^{<T};\hat{\boldsymbol{\theta}}}$, then $\mathbb{P}_{\boldsymbol{y}^{\leq T}|\boldsymbol{a}^{<T};\boldsymbol{\theta}} = \mathbb{P}_{\boldsymbol{y}^{\leq T}|\boldsymbol{a}^{<T};\hat{\boldsymbol{\theta}}}$. The following argument makes use of the Fourier transform $\mathcal{F}$ generalized to arbitrary probability measures. This tool is necessary to deal with measures which do not have a density w.r.t either the Lebesgue or the counting measure, as is the case of $\mathbb{P}_{\boldsymbol{y}^{\leq T}|\boldsymbol{a}^{<T};\boldsymbol{\theta}}$ (all its mass is concentrated on the set $\boldsymbol{f}(\mathbb{R}^{d_z})$). See Pollard [2001, Chapter 8] for an introduction and useful properties.

$$\mathbb{P}_{\boldsymbol{x}^{\leq T}|\boldsymbol{a}^{<T};\boldsymbol{\theta}} = \mathbb{P}_{\boldsymbol{x}^{\leq T}|\boldsymbol{a}^{<T};\hat{\boldsymbol{\theta}}} \tag{5.89}$$

$$\mathbb{P}_{\boldsymbol{y}^{\leq T}|\boldsymbol{a}^{<T};\boldsymbol{\theta}} * \mathbb{P}_{\boldsymbol{n}^{\leq T}} = \mathbb{P}_{\boldsymbol{y}^{\leq T}|\boldsymbol{a}^{<T};\hat{\boldsymbol{\theta}}} * \mathbb{P}_{\boldsymbol{n}^{\leq T}} \tag{5.90}$$

$$\mathcal{F}(\mathbb{P}_{\boldsymbol{y}^{\leq T}|\boldsymbol{a}^{<T};\boldsymbol{\theta}} * \mathbb{P}_{\boldsymbol{n}^{\leq T}}) = \mathcal{F}(\mathbb{P}_{\boldsymbol{y}^{\leq T}|\boldsymbol{a}^{<T};\hat{\boldsymbol{\theta}}} * \mathbb{P}_{\boldsymbol{n}^{\leq T}}) \tag{5.91}$$

$$\mathcal{F}(\mathbb{P}_{\boldsymbol{y}^{\leq T}|\boldsymbol{a}^{<T};\boldsymbol{\theta}})\mathcal{F}(\mathbb{P}_{\boldsymbol{n}^{\leq T}}) = \mathcal{F}(\mathbb{P}_{\boldsymbol{y}^{\leq T}|\boldsymbol{a}^{<T};\hat{\boldsymbol{\theta}}})\mathcal{F}(\mathbb{P}_{\boldsymbol{n}^{\leq T}}) \tag{5.92}$$

$$\mathcal{F}(\mathbb{P}_{\boldsymbol{y}^{\leq T}|\boldsymbol{a}^{<T};\boldsymbol{\theta}}) = \mathcal{F}(\mathbb{P}_{\boldsymbol{y}^{\leq T}|\boldsymbol{a}^{<T};\hat{\boldsymbol{\theta}}}) \tag{5.93}$$

$$\mathbb{P}_{\boldsymbol{y}^{\leq T}|\boldsymbol{a}^{<T};\boldsymbol{\theta}} = \mathbb{P}_{\boldsymbol{y}^{\leq T}|\boldsymbol{a}^{<T};\hat{\boldsymbol{\theta}}}\,, \tag{5.94}$$

where (5.91) & (5.94) use the fact that the Fourier transform is invertible, (5.92) is an application of the fact that the Fourier transform of a convolution is the product of their Fourier transforms and (5.93) holds because the Fourier transform of a Normal distribution is nonzero everywhere. Note that the latter argument holds because we assume σ^2, the variance of the Gaussian noise added to $\boldsymbol{y}^t$, is the same for both models. Notice that, since $\boldsymbol{f}(\mathbb{R}^{d_z})$ and $\hat{\boldsymbol{f}}(\mathbb{R}^{d_z})$ are closed in $\mathbb{R}^{d_x}$ and the support of $\boldsymbol{z}^{\leq T}$ is $\mathbb{R}^{d_z \times T}$, Lemma 5.6 implies that

$$\boldsymbol{f}(\mathbb{R}^{d_z \times T}) = \text{supp}(\mathbb{P}_{\boldsymbol{y}^{\leq T}|\boldsymbol{a}^{<T};\boldsymbol{\theta}}) \qquad \& \qquad \text{supp}(\mathbb{P}_{\boldsymbol{y}^{\leq T}|\boldsymbol{a}^{<T};\hat{\boldsymbol{\theta}}}) = \hat{\boldsymbol{f}}(\mathbb{R}^{d_z \times T}) \tag{5.95}$$

where we overloaded the notation by defining $\boldsymbol{f}(\boldsymbol{z}^{\leq T}) := (\boldsymbol{f}(\boldsymbol{z}^1), ..., \boldsymbol{f}(\boldsymbol{z}^T))$ and analogously for $\hat{\boldsymbol{f}}(\boldsymbol{z}^{\leq T})$. Since both measure in (5.94) are equal, their supports must also be. This implies that $\boldsymbol{f}(\mathbb{R}^{d_z}) = \hat{\boldsymbol{f}}(\mathbb{R}^{d_z})$, which is part of the definition of equivalence up to diffeomorphism (Definition 5.5).

Equality of densities. Continuing with (5.94),

$$\mathbb{P}_{\boldsymbol{y}^{\leq T}|\boldsymbol{a}^{<T};\boldsymbol{\theta}} = \mathbb{P}_{\boldsymbol{y}^{\leq T}|\boldsymbol{a}^{<T};\hat{\boldsymbol{\theta}}} \tag{5.96}$$

$$\mathbb{P}_{\boldsymbol{z}^{\leq T}|\boldsymbol{a}^{<T};\boldsymbol{\theta}} \circ \boldsymbol{f}^{-1} = \mathbb{P}_{\boldsymbol{z}^{\leq T}|\boldsymbol{a}^{<T};\hat{\boldsymbol{\theta}}} \circ \hat{\boldsymbol{f}}^{-1} \tag{5.97}$$

$$\mathbb{P}_{\boldsymbol{z}^{\leq T}|\boldsymbol{a}^{<T};\boldsymbol{\theta}} \circ \boldsymbol{f}^{-1} \circ \hat{\boldsymbol{f}} = \mathbb{P}_{\boldsymbol{z}^{\leq T}|\boldsymbol{a}^{<T};\hat{\boldsymbol{\theta}}} \tag{5.98}$$

$$\mathbb{P}_{\boldsymbol{z}^{\leq T}|\boldsymbol{a}^{<T};\boldsymbol{\theta}} \circ \boldsymbol{v} = \mathbb{P}_{\boldsymbol{z}^{\leq T}|\boldsymbol{a}^{<T};\hat{\boldsymbol{\theta}}}\,, \tag{5.99}$$

where $\boldsymbol{v} := \boldsymbol{f}^{-1} \circ \hat{\boldsymbol{f}}$ is a composition of diffeomorphisms and thus a diffeomorphism from $\mathbb{R}^{d_z}$ to itself. Note that this composition is well defined because $\boldsymbol{f}(\mathbb{R}^{d_z}) = \hat{\boldsymbol{f}}(\mathbb{R}^{d_z})$. We chose to work directly with measures (functions on sets), as opposed to manifold integrals in Khemakhem et al. [2020a], because it simplifies the derivation of (5.99) and avoids having to define densities w.r.t. measures concentrated on a manifold.

The density of $\mathbb{P}_{\boldsymbol{z}^{\leq T}|\boldsymbol{a}^{<T};\boldsymbol{\theta}} \circ \boldsymbol{v}$ w.r.t. to the Lebesgue measure is given by the change-of-variable rule for random vectors (which can be applied because $\boldsymbol{v}$ is a diffeomorphism) and is given by $\prod_{t=1}^{T} p(\boldsymbol{v}(\boldsymbol{z}^t) \mid \boldsymbol{v}(\boldsymbol{z}^{<t}), \boldsymbol{a}^{<t})|\det D\boldsymbol{v}(\boldsymbol{z}^t)|$, where p refers to the density model with parameter $\boldsymbol{\theta}$ and $D\boldsymbol{v}(\boldsymbol{z}^t)$ is the Jacobian matrix of $\boldsymbol{v}$. Since $\mathbb{P}_{\boldsymbol{z}^{\leq T}|\boldsymbol{a}^{<T};\boldsymbol{\theta}} \circ \boldsymbol{v} = \mathbb{P}_{\boldsymbol{z}^{\leq T}|\boldsymbol{a}^{<T};\hat{\boldsymbol{\theta}}}$, their respective densities w.r.t. Lebesgue must also agree:

$$\prod_{t=1}^{T} \hat{p}(\boldsymbol{z}^t \mid \boldsymbol{z}^{<t}, \boldsymbol{a}^{<t}) = \prod_{t=1}^{T} p(\boldsymbol{v}(\boldsymbol{z}^t) \mid \boldsymbol{v}(\boldsymbol{z}^{<t}), \boldsymbol{a}^{<t})|\det D\boldsymbol{v}(\boldsymbol{z}^t)|\,, \tag{5.100}$$

where $\hat{p}$ refers to the conditional density of the model with parameter $\hat{\boldsymbol{\theta}}$.

For a given t_0, we have

$$\prod_{t=1}^{t_0} \hat{p}(\boldsymbol{z}^t \mid \boldsymbol{z}^{<t}, \boldsymbol{a}^{<t}) = \prod_{t=1}^{t_0} p(\boldsymbol{v}(\boldsymbol{z}^t) \mid \boldsymbol{v}(\boldsymbol{z}^{<t}), \boldsymbol{a}^{<t})|\det D\boldsymbol{v}(\boldsymbol{z}^t)|\,, \tag{5.101}$$

by integrating first $\boldsymbol{z}^T$, then $\boldsymbol{z}^{t\text{-}1}$, then ..., up to $\boldsymbol{z}^{t_0+1}$. Note that we can integrate $\boldsymbol{z}^{t_0}$ and get

$$\prod_{t=1}^{t_0-1} \hat{p}(\boldsymbol{z}^t \mid \boldsymbol{z}^{<t}, \boldsymbol{a}^{<t}) = \prod_{t=1}^{t_0-1} p(\boldsymbol{v}(\boldsymbol{z}^t) \mid \boldsymbol{v}(\boldsymbol{z}^{<t}), \boldsymbol{a}^{<t})|\det D\boldsymbol{v}(\boldsymbol{z}^t)|\,. \tag{5.102}$$

By dividing (5.101) by (5.102), we get

$$\hat{p}(\boldsymbol{z}^{t_0} \,|\boldsymbol{z}^{<t_0}, \boldsymbol{a}^{<t_0}) = p(\boldsymbol{v}(\boldsymbol{z}^{t_0}) \mid \boldsymbol{v}(\boldsymbol{z}^{<t_0}), \boldsymbol{a}^{<t_0})|\det D\boldsymbol{v}(\boldsymbol{z}^{t_0})|\,, \tag{5.103}$$

which completes the proof. ■

A.4. The consistency relations (Definitions 5.13 & 5.14) are equivalence relations

In this section, we demonstrate that the relations $\sim^{a}_{\text{con}}$ and $\sim^{z}_{\text{con}}$ are equivalence relations by leveraging the the fact that the set of $\boldsymbol{G}$-preserving diffeomorphisms form a group under composition (Proposition 5.5). We start by showing a fact that will be useful below.

Lemma 5.7. *Let $\boldsymbol{G} \in \{0,1\}^{m\times n}$.*

(1) A map $\boldsymbol{c} : \mathbb{R}^m \to \mathbb{R}^m$ is $\boldsymbol{G}$-preserving if and only if $\boldsymbol{c}$ is $\boldsymbol{GP}$-preserving, where $\boldsymbol{P}$ is an $n \times n$ permutation matrix.

(2) A map $\boldsymbol{c} : \mathbb{R}^m \to \mathbb{R}^m$ is $\boldsymbol{G}$-preserving if and only if $\boldsymbol{P} \circ \boldsymbol{c} \circ \boldsymbol{P}^\top$ is $\boldsymbol{PG}$-preserving, where $\boldsymbol{P}$ is a $m \times m$ permutation matrix.

(3) When m = n, a map $\boldsymbol{c} : \mathbb{R}^m \to \mathbb{R}^m$ is $\boldsymbol{G}$-preserving if and only if $\boldsymbol{P} \circ \boldsymbol{c} \circ \boldsymbol{P}^\top$ is $\boldsymbol{PGP}^\top$-preserving, where $\boldsymbol{P}$ is a $m \times m$ permutation matrix.

Proof Let $\boldsymbol{C}$ be the dependency graph of $\boldsymbol{c}$.

(1) $\boldsymbol{C}^\top \mathbb{R}^{m\times n}_{\boldsymbol{G}} \subseteq \mathbb{R}^{m\times n}_{\boldsymbol{G}} \iff \boldsymbol{C}^\top \mathbb{R}^{m\times n}_{\boldsymbol{G}} \boldsymbol{P} \subseteq \mathbb{R}^{m\times n}_{\boldsymbol{G}} \boldsymbol{P} \iff \boldsymbol{C}^\top \mathbb{R}^{m\times n}_{\boldsymbol{GP}} \subseteq \mathbb{R}^{m\times n}_{\boldsymbol{GP}}$

(2) First, notice that the dependency graph of $\boldsymbol{P} \circ \boldsymbol{c} \circ \boldsymbol{P}^\top$ is $\boldsymbol{PCP}^\top$.

$$(\boldsymbol{PCP}^\top)^\top \mathbb{R}^{m\times n}_{\boldsymbol{PG}} \subseteq \mathbb{R}^{m\times n}_{\boldsymbol{PG}} \iff \boldsymbol{PC}^\top \boldsymbol{P}^\top \boldsymbol{P} \mathbb{R}^{m\times n}_{\boldsymbol{G}} \subseteq \boldsymbol{P}\mathbb{R}^{m\times n}_{\boldsymbol{G}} \iff \boldsymbol{C}^\top \mathbb{R}^{m\times n}_{\boldsymbol{G}} \subseteq \mathbb{R}^{m\times n}_{\boldsymbol{G}}$$

(3) This is a consequence of the first two statements.

■

We are now ready to show that the relation $\sim^{a}_{\text{con}}$ (Definition 5.13) is an equivalence relation.

Proposition 5.9. *The consistency relation, $\sim^{a}_{\text{con}}$ (Def. 5.13), is an equivalence relation.*

Proof

Reflexivity. It is easy to see that $\boldsymbol{\theta} \sim^{a}_{\text{con}} \boldsymbol{\theta}$, by simply setting $\boldsymbol{v}(\boldsymbol{z}) := \boldsymbol{z}$ with $\boldsymbol{P} := \boldsymbol{I}$.

Symmetry. Assume $\boldsymbol{\theta} \sim^{a}_{\text{con}} \tilde{\boldsymbol{\theta}}$. Hence, we have $\boldsymbol{PG}^a = \tilde{\boldsymbol{G}}^a$ as well as

$$\boldsymbol{f}(\mathbb{R}^{d_z}) = \tilde{\boldsymbol{f}}(\mathbb{R}^{d_z}), \text{and} \tag{5.104}$$

$$\tilde{p}(\boldsymbol{z}^t \mid \boldsymbol{z}^{<t}, \boldsymbol{a}^{<t}) = p(\boldsymbol{v}(\boldsymbol{z}^t) \mid \boldsymbol{v}(\boldsymbol{z}^{<t}), \boldsymbol{a}^{<t}) |\det D\boldsymbol{v}(\boldsymbol{z}^t)|\,, \tag{5.105}$$

where $\boldsymbol{v} := \boldsymbol{f}^{-1} \circ \tilde{\boldsymbol{f}}$ can be written as $\boldsymbol{v} := \boldsymbol{c} \circ \boldsymbol{P}^\top$, where $\boldsymbol{c}$ is a $\boldsymbol{G}^a$-preserving diffeomorphism and $\boldsymbol{P}$ is a permutation. We can massage (5.105) to get

$$p(\boldsymbol{z}^t \mid \boldsymbol{z}^{<t}, \boldsymbol{a}^{<t}) = \tilde{p}(\boldsymbol{v}^{-1}(\boldsymbol{z}^t) \mid \boldsymbol{v}^{-1}(\boldsymbol{z}^{<t}), \boldsymbol{a}^{<t}) |\det D\boldsymbol{v}^{-1}(\boldsymbol{z}^t)|\,. \tag{5.106}$$

Of course, we also have that $\tilde{\boldsymbol{P}}\tilde{\boldsymbol{G}}^a = \boldsymbol{G}^a$, where $\tilde{\boldsymbol{P}} := \boldsymbol{P}^\top$. Now the only thing left to prove is that $\boldsymbol{v}^{-1}$ can be written as $\tilde{\boldsymbol{c}} \circ \tilde{\boldsymbol{P}}^\top$ where $\tilde{\boldsymbol{c}}$ is $\tilde{\boldsymbol{G}}^a$-preserving. We know that

$$\boldsymbol{v}^{-1} = \boldsymbol{P} \circ \boldsymbol{c}^{-1} = \underbrace{\boldsymbol{P} \circ \boldsymbol{c}^{-1} \circ \boldsymbol{P}^\top}_{\tilde{\boldsymbol{c}} :=} \circ \boldsymbol{P} = \tilde{\boldsymbol{c}} \circ \tilde{\boldsymbol{P}}^\top . \tag{5.107}$$

Note that $\boldsymbol{c}^{-1}$ is $\boldsymbol{G}^a$-preserving and thus, by Lemma 5.7, $\tilde{\boldsymbol{c}}$ is $\boldsymbol{P}\boldsymbol{G}^a$-preserving, i.e. $\tilde{\boldsymbol{G}}^a$-preserving. Hence, $\sim^a_{\text{con}}$ is symmetric.

Transitivity. Suppose $\boldsymbol{\theta} \sim^a_{\text{con}} \tilde{\boldsymbol{\theta}}$ and $\tilde{\boldsymbol{\theta}} \sim^a_{\text{con}} \hat{\boldsymbol{\theta}}$. This means

$$\boldsymbol{P}_1 \boldsymbol{G}^a = \tilde{\boldsymbol{G}}^a , \tag{5.108}$$

$$\boldsymbol{f}(\mathbb{R}^{d_z}) = \tilde{\boldsymbol{f}}(\mathbb{R}^{d_z}), \text{and} \tag{5.109}$$

$$\tilde{p}(\boldsymbol{z}^t \mid \boldsymbol{z}^{<t}, \boldsymbol{a}^{<t}) = p(\boldsymbol{v}_1(\boldsymbol{z}^t) \mid \boldsymbol{v}_1(\boldsymbol{z}^{<t}), \boldsymbol{a}^{<t}) |\det D\boldsymbol{v}_1(\boldsymbol{z}^t)| , \tag{5.110}$$

where $\boldsymbol{v}_1 := \boldsymbol{c}_1 \circ \boldsymbol{P}_1^\top$ with $\boldsymbol{c}_1$ being $\boldsymbol{G}^a$-preserving; and

$$\boldsymbol{P}_2 \tilde{\boldsymbol{G}}^a = \hat{\boldsymbol{G}}^a , \tag{5.111}$$

$$\tilde{\boldsymbol{f}}(\mathbb{R}^{d_z}) = \hat{\boldsymbol{f}}(\mathbb{R}^{d_z}), \text{and} \tag{5.112}$$

$$\hat{p}(\boldsymbol{z}^t \mid \boldsymbol{z}^{<t}, \boldsymbol{a}^{<t}) = \tilde{p}(\boldsymbol{v}_2(\boldsymbol{z}^t) \mid \boldsymbol{v}_2(\boldsymbol{z}^{<t}), \boldsymbol{a}^{<t}) |\det D\boldsymbol{v}_2(\boldsymbol{z}^t)| , \tag{5.113}$$

where $\boldsymbol{v}_2 := \boldsymbol{c}_2 \circ \boldsymbol{P}_2^\top$ with $\boldsymbol{c}_2$ being $\tilde{\boldsymbol{G}}^a$-preserving.

To show that $\boldsymbol{\theta} \sim^a_{\text{con}} \hat{\boldsymbol{\theta}}$, we first combine (5.108) with (5.111) to get

$$\underbrace{\boldsymbol{P}_2 \boldsymbol{P}_1}_{\boldsymbol{P} :=} \boldsymbol{G}^a = \hat{\boldsymbol{G}}^a . \tag{5.114}$$

Of course we also have that $\boldsymbol{f}(\mathbb{R}^{d_z}) = \tilde{\boldsymbol{f}}(\mathbb{R}^{d_z}) = \hat{\boldsymbol{f}}(\mathbb{R}^{d_z})$. By massaging both (5.110) and (5.113), we get:

$$\hat{p}(\boldsymbol{z}^t \mid \boldsymbol{z}^{<t}, \boldsymbol{a}^{<t}) = p(\boldsymbol{v}_1 \circ \boldsymbol{v}_2(\boldsymbol{z}^t) \mid \boldsymbol{v}_1 \circ \boldsymbol{v}_2(\boldsymbol{z}^{<t}), \boldsymbol{a}^{<t}) |\det D(\boldsymbol{v}_1 \circ \boldsymbol{v}_2)(\boldsymbol{z}^t)| . \tag{5.115}$$

Define $\boldsymbol{v} := \boldsymbol{v}_1 \circ \boldsymbol{v}_2$. We now want to show that $\boldsymbol{v}$ can be written has $\boldsymbol{v} = \boldsymbol{c} \circ \boldsymbol{P}^\top$ where $\boldsymbol{c}$ is $\boldsymbol{G}^a$-preserving. We have that

$$\boldsymbol{v}_1 \circ \boldsymbol{v}_2 = \boldsymbol{c}_1 \circ \boldsymbol{P}_1^\top \circ \boldsymbol{c}_2 \circ \boldsymbol{P}_2^\top \tag{5.116}$$

$$= \boldsymbol{c}_1 \circ \underbrace{\boldsymbol{P}_1^\top \circ \boldsymbol{P}_2^\top}_{\boldsymbol{P}^\top =} \circ \underbrace{\boldsymbol{P}_2 \circ \boldsymbol{c}_2 \circ \boldsymbol{P}_2^\top}_{\hat{\boldsymbol{c}} :=} \tag{5.117}$$

$$= \boldsymbol{c}_1 \circ \boldsymbol{P}^\top \circ \hat{\boldsymbol{c}} \tag{5.118}$$

where, by Lemma 5.7, $\hat{\boldsymbol{c}}$ is $\boldsymbol{P}_2\tilde{\boldsymbol{G}}^a$-preserving, i.e. $\hat{\boldsymbol{G}}^a$-preserving. We continue and get that

$$\boldsymbol{v}_1 \circ \boldsymbol{v}_2 = \boldsymbol{c}_1 \circ \boldsymbol{P}^\top \circ \hat{\boldsymbol{c}} \tag{5.119}$$

$$= \boldsymbol{c}_1 \circ \underbrace{\boldsymbol{P}^\top \circ \hat{\boldsymbol{c}} \circ \boldsymbol{P}}_{\boldsymbol{c}' :=} \circ \boldsymbol{P}^\top \tag{5.120}$$

$$= \boldsymbol{c}_1 \circ \boldsymbol{c}' \circ \boldsymbol{P}^\top , \tag{5.121}$$

where, by Lemma 5.7, $\boldsymbol{c}'$ is $\boldsymbol{P}^\top\hat{\boldsymbol{G}}^a$-preserving, i.e. $\boldsymbol{G}^a$-preserving (by (5.114)). Since both $\boldsymbol{c}_1$ and $\boldsymbol{c}'$ are $\boldsymbol{G}^a$-preserving, $\boldsymbol{c} := \boldsymbol{c}_1 \circ \boldsymbol{c}'$ is $\boldsymbol{G}^a$-preserving, which concludes the proof. ■

The same can be shown for $\sim^z_{\text{con}}$ (Definition 5.14).

Proposition 5.10. *The consistency relation, $\sim^z_{\text{con}}$ (Def. 5.14), is an equivalence relation.*

Proof The proof is exactly analogous to the proof that $\sim^a_{\text{con}}$ is an equivalence relation. Essentially, every statement of the form "$\boldsymbol{P}\boldsymbol{G}^a = \tilde{\boldsymbol{G}}^a$" becomes "$\boldsymbol{P}\boldsymbol{G}^z\boldsymbol{P}^\top = \tilde{\boldsymbol{G}}^z$" and statements of the form "$\boldsymbol{c}$ is $\boldsymbol{G}^a$-preserving" becomes "$\boldsymbol{c}$ is $\boldsymbol{G}^z$-preserving and $(\boldsymbol{G}^z)^\top$-preserving". The full proof is left as an exercise to the reader. ■

A.4.1. Combining equivalence relations.

Proposition 5.6. *Let $\boldsymbol{\theta} := (\boldsymbol{f}, p, \boldsymbol{G})$ and $\tilde{\boldsymbol{\theta}} := (\tilde{\boldsymbol{f}}, \tilde{p}, \tilde{\boldsymbol{G}})$ be two models satisfying Assumptions 5.1, 5.2 & 5.3. We have $\boldsymbol{\theta} \sim^{z,a}_{\text{con}} \tilde{\boldsymbol{\theta}}$ if and only if $\boldsymbol{\theta} \sim^a_{\text{con}} \tilde{\boldsymbol{\theta}}$ and $\boldsymbol{\theta} \sim^z_{\text{con}} \tilde{\boldsymbol{\theta}}$.*

Proof The "only if" part of the statement is trivial. We now show the "if" part.

Let $\boldsymbol{v} := \boldsymbol{f}^{-1} \circ \tilde{\boldsymbol{f}}$. Since $\boldsymbol{\theta} \sim^a_{\text{con}} \tilde{\boldsymbol{\theta}}$, we have that $\tilde{\boldsymbol{G}}^a = \boldsymbol{P}\boldsymbol{G}^a$ and $\boldsymbol{v} = \boldsymbol{c} \circ \boldsymbol{P}^\top$ where $\boldsymbol{P}$ a permutation matrix and $\boldsymbol{c}$ is $\boldsymbol{G}^a$-preserving. Since $\boldsymbol{\theta} \sim^z_{\text{con}} \tilde{\boldsymbol{\theta}}$, we have that $\tilde{\boldsymbol{G}}^z = \bar{\boldsymbol{P}}\boldsymbol{G}^z\bar{\boldsymbol{P}}^\top$ and $\boldsymbol{v} = \bar{\boldsymbol{c}} \circ \bar{\boldsymbol{P}}^\top$ where $\bar{\boldsymbol{P}}$ is a permutation matrix and $\bar{\boldsymbol{c}}$ is $\boldsymbol{G}^z$-preserving and $(\boldsymbol{G}^z)^\top$-preserving. Let $\boldsymbol{C}$ and $\bar{\boldsymbol{C}}$ be the dependency graphs $\boldsymbol{c}$ and $\bar{\boldsymbol{c}}$, respectively.

Choose an arbitrary $\boldsymbol{z}$. Since $D\boldsymbol{c}(\boldsymbol{z})$ is invertible, Lemma 5.2 implies that there exists a permutation $\boldsymbol{P}_0$ such that $\boldsymbol{P}_0^\top \subseteq D\boldsymbol{v}(\boldsymbol{z})$, which in turns implies that $\boldsymbol{P}_0^\top \subseteq \bar{\boldsymbol{C}}$. Because $\boldsymbol{P}_0^\top \subseteq \bar{\boldsymbol{C}}$, we have that $\boldsymbol{P}_0^\top$ is $\boldsymbol{G}^z$-preserving and $(\boldsymbol{G}^z)^\top$-preserving (Proposition 5.3). By closure under composition and inversion, $\bar{\boldsymbol{c}} \circ \boldsymbol{P}_0$ is $\boldsymbol{G}^z$- and $(\boldsymbol{G}^z)^\top$-preserving.

Note that

$$\boldsymbol{c} \circ \boldsymbol{P}^\top = \bar{\boldsymbol{c}} \circ \bar{\boldsymbol{P}}^\top \tag{5.122}$$

$$\implies \boldsymbol{C}\boldsymbol{P}^\top = \bar{\boldsymbol{C}}\bar{\boldsymbol{P}}^\top \tag{5.123}$$

$$\boldsymbol{C}\boldsymbol{P}^\top\bar{\boldsymbol{P}} = \bar{\boldsymbol{C}} \supseteq \boldsymbol{P}_0^\top \tag{5.124}$$

$$\implies \boldsymbol{C} \supseteq \boldsymbol{P}_0^\top\bar{\boldsymbol{P}}^\top\boldsymbol{P} . \tag{5.125}$$

This means the permutation $\boldsymbol{P}_0^\top\bar{\boldsymbol{P}}^\top\boldsymbol{P}$ must be $\boldsymbol{G}^a$-preserving since $\boldsymbol{C}$ is (Proposition 5.3).

This further implies that $\boldsymbol{C}\boldsymbol{P}^\top\bar{\boldsymbol{P}}\boldsymbol{P}_0 = \bar{\boldsymbol{C}}\boldsymbol{P}_0$ is $\boldsymbol{G}^a$-preserving by closure under multiplication (recall $\boldsymbol{C}$ is $\boldsymbol{G}^a$-preserving too). Hence $\bar{\boldsymbol{c}} \circ \boldsymbol{P}_0$ is $\boldsymbol{G}^a$-preserving.

We thus have that $\boldsymbol{v} = (\bar{\boldsymbol{c}} \circ \boldsymbol{P}_0)(\bar{\boldsymbol{P}}\boldsymbol{P}_0)^\top$ where $(\bar{\boldsymbol{c}} \circ \boldsymbol{P}_0)$ is $\boldsymbol{G}^a$-, $\boldsymbol{G}^z$- and $(\boldsymbol{G}^z)^\top$-preserving. The only thing left to show is that $(\bar{\boldsymbol{P}}\boldsymbol{P}_0)\boldsymbol{G}^a = \tilde{\boldsymbol{G}}^a$ and that $(\bar{\boldsymbol{P}}\boldsymbol{P}_0)\boldsymbol{G}^z(\bar{\boldsymbol{P}}\boldsymbol{P}_0)^\top = \tilde{\boldsymbol{G}}^z$. The former holds since

$$(\bar{\boldsymbol{P}}\boldsymbol{P}_0)\boldsymbol{G}^a = \boldsymbol{P}(\boldsymbol{P}^\top\bar{\boldsymbol{P}}\boldsymbol{P}_0)\boldsymbol{G}^a = \boldsymbol{P}\boldsymbol{G}^a = \tilde{\boldsymbol{G}}^a\,,$$

where the second equality leverages the fact that $\boldsymbol{P}^\top\bar{\boldsymbol{P}}\boldsymbol{P}_0$ is $\boldsymbol{G}^a$-preserving. Furthermore,

$$(\bar{\boldsymbol{P}}\boldsymbol{P}_0)\boldsymbol{G}^z(\bar{\boldsymbol{P}}\boldsymbol{P}_0)^\top = \bar{\boldsymbol{P}}\boldsymbol{G}^z(\bar{\boldsymbol{P}}\boldsymbol{P}_0)^\top = \bar{\boldsymbol{P}}\boldsymbol{G}^z\boldsymbol{P}_0^\top\bar{\boldsymbol{P}}^\top = \bar{\boldsymbol{P}}(\boldsymbol{P}_0(\boldsymbol{G}^z)^\top)^\top\bar{\boldsymbol{P}}^\top = \bar{\boldsymbol{P}}\boldsymbol{G}^z\bar{\boldsymbol{P}}^\top = \tilde{\boldsymbol{G}}^z\,,$$

where the first and fourth equalities leveraged the fact that $\boldsymbol{P}_0$ is $\boldsymbol{G}^z$- and $(\boldsymbol{G}^z)^\top$-preserving. ■

A.5. Technical lemmas in the proof of Theorems 5.1, 5.2 & 5.3

The goal of this section is to introduce and prove Lemma 5.12 which was crucial in proofs of Theorems 5.1, 5.2 & 5.3. To prove it, we need a few more results, which we present next.

The following two lemmas are standard, but we provide them with proofs for completeness.

Lemma 5.8. *Let $f : \mathbb{R}^n \to \mathbb{R}$ be continuous and $A \subseteq \mathbb{R}^n$. If, for all $\boldsymbol{x} \in A$, $\boldsymbol{f}(\boldsymbol{x}) = 0$, then the equality holds on $\overline{A}$.*

Proof We have that $A \subseteq \boldsymbol{f}^{-1}(\{0\})$. Since $\{0\}$ is closed, $\boldsymbol{f}^{-1}(\{0\})$ is also closed by continuity of f. This means $\overline{A} \subseteq \boldsymbol{f}^{-1}(\{0\})$ (since the closure of $\boldsymbol{A}$ is the smallest closed set containing $\boldsymbol{A}$). ■

Lemma 5.9. *Let μ be the Lebesgue measure on $\mathbb{R}^{d_z}$ and let $E_0 \subseteq \mathbb{R}^{d_z}$ be a zero measure set, i.e. $\mu(E_0) = 0$. Then, $\overline{\mathbb{R}^{d_z} \setminus E_0} = \mathbb{R}^{d_z}$.*

Proof Clearly, $\overline{\mathbb{R}^{d_z} \setminus E_0} \subseteq \mathbb{R}^{d_z}$.

We now show that $\mathbb{R}^{d_z} \subseteq \overline{\mathbb{R}^{d_z} \setminus E_0}$. Take $\boldsymbol{z}_0 \in \mathbb{R}^{d_z}$ and let U be an open set of $\mathbb{R}^{d_z}$ containing $\boldsymbol{z}_0$. Every open sets have nonzero Lebesgue measure, so

$$0 \neq \mu(U) = \mu(U \cap \mathbb{R}^{d_z}) = \mu(U \cap (\mathbb{R}^{d_z} \setminus E_0)) \implies U \cap (\mathbb{R}^{d_z} \setminus E_0) \neq \emptyset\,. \tag{5.126}$$

Since U was arbitrary, this means $\boldsymbol{z}_0 \in \overline{\mathbb{R}^{d_z} \setminus E_0}$. ■

This simple lemma will come in handy when proving Lemma 5.11.

Lemma 5.10. *If a permutation $\boldsymbol{P}$ is not $\boldsymbol{G}$-preserving and $\boldsymbol{C}$ is a $\boldsymbol{G}$-preserving matrix, we have that $\boldsymbol{C}\boldsymbol{P}^\top$ and $\boldsymbol{P}^\top\boldsymbol{C}$ have a zero on their diagonal.*

Proof Assume $\boldsymbol{P}$ is not $\boldsymbol{G}$-preserving, hence there exists i, j such that $\boldsymbol{G}_{i,\cdot} \not\subseteq \boldsymbol{G}_{j,\cdot}$, but $\boldsymbol{P}_{i,j} = 1$. Now note that

$$(\boldsymbol{C}\boldsymbol{P}^\top)_{i,i} = \boldsymbol{C}_{i,\cdot}(\boldsymbol{P}_{i\cdot})^\top = \boldsymbol{C}_{i,\cdot}\boldsymbol{e}_j = \boldsymbol{C}_{i,j}\,, \tag{5.127}$$

which is equal to zero because $\boldsymbol{C}$ is $\boldsymbol{G}$-preserving and $\boldsymbol{G}_{i,\cdot} \not\subseteq \boldsymbol{G}_{j,\cdot}$. Similarly,

$$(\boldsymbol{P}^\top\boldsymbol{C})_{j,j} = (\boldsymbol{P}_{\cdot,j})^\top\boldsymbol{C}_{\cdot,j} = \boldsymbol{e}_i^\top\boldsymbol{C}_{\cdot,j} = \boldsymbol{C}_{i,j} = 0\,. \tag{5.128}$$

which concludes the proof. ■

The following lemma is the same as Lemma 5.12 which is used to proved Theorems 5.1, 5.2 & 5.3, except it does not take into account the "almost everywhere" subtlety. Lemma 5.12 will extend it to deal with this difficulty.

Lemma 5.11. *Let $\boldsymbol{G} \in \{0,1\}^{m\times n}$, let $\mathcal{Z}$ be a connected subset of some topological space and let $\boldsymbol{L} : \mathcal{Z} \to \mathbb{R}^{m\times m}$ be a continuous function such that $\boldsymbol{L}(\boldsymbol{z})$ is invertible for all $\boldsymbol{z} \in \mathcal{Z}$. Suppose that, for all $\boldsymbol{z} \in \mathcal{Z}$, there exists a permutation matrix $\boldsymbol{P}(\boldsymbol{z})$ such that $\boldsymbol{L}(\boldsymbol{z})\boldsymbol{P}(\boldsymbol{z})$ is $\boldsymbol{G}$-preserving. Then, there exists a permutation matrix $\boldsymbol{P}$ such that, for all $\boldsymbol{z} \in \mathcal{Z}$, $\boldsymbol{L}(\boldsymbol{z})\boldsymbol{P}$ is $\boldsymbol{G}$-preserving.*

Proof The goal of this lemma is to show that in the statement above, one can change the order of the "for all $\boldsymbol{z} \in \mathcal{Z}$" and "there exists a permutation". To do that, we show that if $\mathcal{Z}$ is connected and the map $\boldsymbol{L}(\cdot)$ is continuous, then one can find a single permutation that works for all $\boldsymbol{z} \in \mathcal{Z}$.

Let $\mathcal{G}$ be the set of $\boldsymbol{G}^a$-preserving matrices. Recall that, by Proposition 5.3, $\mathcal{G}$ corresponds to all matrices that have some set of entries equal to zero.

First, since $\mathcal{Z}$ is connected and $\boldsymbol{L}$ is continuous, its image, $\boldsymbol{L}(\mathcal{Z})$, must be connected (by [Munkres, 2000, Theorem 23.5]).

Second, from the hypothesis of the lemma, we know that

$$\boldsymbol{L}(\mathcal{Z}) \subseteq \mathcal{L} := \left(\bigcup_{\pi\in\mathfrak{S}_m} \mathcal{G}\boldsymbol{P}_\pi\right) \setminus \{\text{singular matrices}\}\,, \tag{5.129}$$

where $\mathfrak{S}_m$ is the set of permutations and $\mathcal{G}\boldsymbol{P}_\pi = \{\boldsymbol{L}\boldsymbol{P}_\pi \mid \boldsymbol{L} \in \mathcal{G}\}$. We can rewrite the set $\mathcal{L}$ above as

$$\mathcal{L} = \left(\bigcup_{\pi\in\mathfrak{S}_m} \mathcal{G}\boldsymbol{P}_\pi \setminus \{\text{singular matrices}\}\right)\,. \tag{5.130}$$

We now define an equivalence relation $\sim$ over permutations: $\pi \sim \pi'$ iff $\boldsymbol{P}_\pi\boldsymbol{P}_{\pi'}^\top$ is $\boldsymbol{G}$-preserving. One can verify that the relation $\sim$ is indeed an equivalence relation by using the fact that invertible $\boldsymbol{G}$-preserving matrices form a group (Proposition 5.4). We notice that

$$\pi \sim \pi' \implies \mathcal{G} = \mathcal{G}\boldsymbol{P}_\pi\boldsymbol{P}_{\pi'}^\top \implies \mathcal{G}\boldsymbol{P}_{\pi'} = \mathcal{G}\boldsymbol{P}_\pi\,, \tag{5.131}$$

where the first implication holds because $\boldsymbol{G}$-preserving matrices are closed under matrix multiplication (Proposition 5.4). Let $\mathfrak{S}_m/\sim$ be the set of equivalence classes induce by $\sim$ and let Π stand for one such equivalence class. Thanks to (5.131), we can define, for all $\Pi \in \mathfrak{S}_m/\sim$, the following set:

$$V_\Pi := \mathcal{G}\boldsymbol{P}_\pi \setminus \{\text{singular matrices}\}, \text{ for some } \pi \in \Pi\,, \tag{5.132}$$

where the specific choice of $\pi \in \Pi$ is arbitrary (any $\pi' \in \Pi$ would yield the same definition, by (5.131)). This construction allows us to write

$$\mathcal{L} = \bigcup_{\Pi \in \mathfrak{S}_m/\sim} V_\Pi\,. \tag{5.133}$$

We now show that $\{V_\Pi\}_{\Pi\in\mathfrak{S}_m/\sim}$ forms a partition of $\mathcal{L}$. Choose two distinct equivalence classes of permutations Π and Π' and let $\pi \in \Pi$ and $\pi' \in \Pi'$ be representatives. We will now prove that

$$\mathcal{G}\boldsymbol{P}_\pi \cap \mathcal{G}\boldsymbol{P}_{\pi'} \subseteq \{\text{singular matrices}\}\,. \tag{5.134}$$

To achieve this, we proceed by contradiction: Suppose there exists an invertible matrix $\boldsymbol{A} \in \mathcal{G}\boldsymbol{P}_\pi \cap \mathcal{G}\boldsymbol{P}_{\pi'}$. By Lemma 5.2, there exists a permutation $\boldsymbol{P}$ s.t. $\boldsymbol{A}\boldsymbol{P}^\top$ has no zero on its diagonal. Of course, the permutation $\boldsymbol{P}$ belongs to only one equivalence class and thus either $\boldsymbol{P} \notin \Pi$ or $\boldsymbol{P} \notin \Pi'$. Without loss of generality, assume the former. We thus have that $\boldsymbol{P} \not\sim \pi$ and thus $\boldsymbol{P}_\pi\boldsymbol{P}^\top$ is not $\boldsymbol{G}$-preserving. We can thus write

$$\boldsymbol{A} \in \mathcal{G}\boldsymbol{P}_\pi \cap \mathcal{G}\boldsymbol{P}_{\pi'} \implies \boldsymbol{A}\boldsymbol{P}^\top \in \mathcal{G}\boldsymbol{P}_\pi\boldsymbol{P}^\top \cap \mathcal{G}\boldsymbol{P}_{\pi'}\boldsymbol{P}^\top\,. \tag{5.135}$$

By Lemma 5.10, all matrices in $\mathcal{G}\boldsymbol{P}_\pi\boldsymbol{P}^\top$ have a zero on their diagonal. This is a contradiction with $\boldsymbol{A}\boldsymbol{P}^\top \in \mathcal{G}\boldsymbol{P}_\pi\boldsymbol{P}^\top$, since, as we said, $\boldsymbol{A}\boldsymbol{P}^\top$ has no zero on its diagonal. We thus conclude that no invertible matrix is in the intersection $\mathcal{G}\boldsymbol{P}_\pi \cap \mathcal{G}\boldsymbol{P}_{\pi'}$ and thus (5.134) holds.

We thus have that

$$V_\Pi \cap V_{\Pi'} = \emptyset\,, \tag{5.136}$$

which shows that $\{V_\Pi\}_{\Pi\in\mathfrak{S}_m/\sim}$ is indeed a partition of $\mathcal{L}$.

Each V_Π is closed in $\mathcal{L}$ (w.r.t. the subset topology inherited from $\mathbb{R}^{m\times m}$) since

$$V_\Pi = \mathcal{G}\boldsymbol{P}_\pi \setminus \{\text{singular matrices}\} = \mathcal{L} \cap \underbrace{\mathcal{G}\boldsymbol{P}_\pi}_{\text{closed in } \mathbb{R}^{m\times m}}\,. \tag{5.137}$$

Moreover, V_Π is open in $\mathcal{L}$, since

$$V_\Pi = \mathcal{L} \setminus \underbrace{\bigcup_{\Pi' \neq \Pi} V_{\Pi'}}_{\text{closed in } \mathcal{L}}\,. \tag{5.138}$$

Thus, for any $\Pi \in \mathfrak{S}(\mathcal{B})/\sim$, the sets V_Π and $\bigcup_{\Pi' \neq \Pi} V_{\Pi'}$ forms a *separation* (see [Munkres, 2000, Section 23]). Since $\boldsymbol{L}(\mathcal{Z})$ is a connected subset of $\mathcal{L}$, it must lie completely in V_Π or $\bigcup_{\Pi' \neq \Pi} V_{\Pi'}$, by [Munkres, 2000, Lemma 23.2]. Since this is true for all Π, it must follow that there exists a Π^* such that $\boldsymbol{L}(\mathcal{Z}) \subseteq V_{\Pi^*}$. Choose any representative $\boldsymbol{P}_* \in \Pi^*$. We thus have that, for all $\boldsymbol{z} \in \mathcal{Z}$, $\boldsymbol{L}(\boldsymbol{z}) = \boldsymbol{C}(\boldsymbol{z})\boldsymbol{P}_*^\top$, where $\boldsymbol{C}(\boldsymbol{z})$ is $\boldsymbol{G}$-preserving, which completes the proof. ■

The goal of the next result is to relax the conditions of Lemma 5.11 so that $\boldsymbol{L}(\boldsymbol{z})\boldsymbol{P}(\boldsymbol{z})$ is $\boldsymbol{G}$-preserving for *almost all* $\boldsymbol{z} \in \mathbb{R}^{d_z}$.

Lemma 5.12. *Let $\boldsymbol{G} \in \{0,1\}^{m \times n}$ and let $\boldsymbol{L} : \mathbb{R}^{d_z} \to \mathbb{R}^{m \times m}$ be a continuous function such that $\boldsymbol{L}(\boldsymbol{z})$ is invertible for all $\boldsymbol{z} \in \mathbb{R}^{d_z}$. Suppose that, for almost all $\boldsymbol{z} \in \mathbb{R}^{d_z}$ (i.e. except on a set E_0 of Lebesgue measure zero), there exists a permutation matrix $\boldsymbol{P}(\boldsymbol{z})$ such that $\boldsymbol{L}(\boldsymbol{z})\boldsymbol{P}(\boldsymbol{z})$ is $\boldsymbol{G}$-preserving. Then, there exists a permutation matrix $\boldsymbol{P}$ such that, for all $\boldsymbol{z} \in \mathbb{R}^{d_z}$, $\boldsymbol{L}(\boldsymbol{z})\boldsymbol{P}$ is $\boldsymbol{G}$-preserving.*

Proof We know that for all $\boldsymbol{z} \in \mathbb{R}^{d_z} \setminus E_0$, where $\mu(E_0) = 0$ (Lebesgue measure zero), there exists a permutation matrix $\boldsymbol{P}(\boldsymbol{z})$ such that $\boldsymbol{L}(\boldsymbol{z})\boldsymbol{P}(\boldsymbol{z})$ is $\boldsymbol{G}$-preserving. For all permutations $\boldsymbol{P}$, define $\mathcal{Z}^{(\boldsymbol{P})} := \{\boldsymbol{z} \in \mathbb{R}^{d_z} \setminus E_0 \mid \boldsymbol{P}(\boldsymbol{z}) = \boldsymbol{P}\}$. The collection of all sets $\mathcal{Z}^{(\boldsymbol{P})}$ is finite (since there are finitely many permutations) and form a partition of $\mathbb{R}^{d_z} \setminus E_0$. Of course, for all $\boldsymbol{P}$, $\boldsymbol{L}(\boldsymbol{z})\boldsymbol{P}$ is $\boldsymbol{G}$-preserving for all $\boldsymbol{z} \in \mathcal{Z}^{(\boldsymbol{P})}$. By Lemma 5.8, we can extend this statement to the closure, i.e. for all $\boldsymbol{z} \in \overline{\mathcal{Z}^{(\boldsymbol{P})}}$, $\boldsymbol{L}(\boldsymbol{z})\boldsymbol{P}$ is $\boldsymbol{G}$-preserving.

Furthermore, we have that $\bigcup_{\boldsymbol{P}} \overline{\mathcal{Z}^{(\boldsymbol{P})}} = \overline{\bigcup_{\boldsymbol{P}} \mathcal{Z}^{(\boldsymbol{P})}} = \overline{\mathbb{R}^{d_z} \setminus E_0} = \mathbb{R}^{d_z}$, where the first equality is a standard property of closure (which holds only for finite unions), and the last equality holds by Lemma 5.9. We thus have that, for all $\boldsymbol{z} \in \mathbb{R}^{d_z}$, there exists a permutation $\boldsymbol{P}(\boldsymbol{z})$ such that $\boldsymbol{L}(\boldsymbol{z})\boldsymbol{P}(\boldsymbol{z})$ is $\boldsymbol{G}$-preserving. Since $\mathbb{R}^{d_z}$ is connected we can apply Lemma 5.11 to get the desired conclusion. ■

A.6. Connecting to the graphical criterion of Lachapelle et al. [2022]

The goal of this section is to prove Proposition 5.7 which states if some graphical criterion holds (Assumption 5.5), then $\boldsymbol{\theta} \sim_{\text{con}}^{z,a} \hat{\boldsymbol{\theta}}$ implies $\boldsymbol{\theta} \sim_{\text{perm}} \hat{\boldsymbol{\theta}}$, i.e. complete disentanglement. We recall Assumption 5.5.

Assumption 5.5 (Graphical criterion, Lachapelle et al. [2022])**.** *Let $\boldsymbol{G} = [\boldsymbol{G}^z \; \boldsymbol{G}^a]$ be a graph. For all $i \in \{1, ..., d_z\}$,*

$$\left(\bigcap_{j \in \mathbf{Ch}_i^z} \mathbf{Pa}_j^z\right) \cap \left(\bigcap_{j \in \mathbf{Pa}_i^z} \mathbf{Ch}_j^z\right) \cap \left(\bigcap_{\ell \in \mathbf{Pa}_i^a} \mathbf{Ch}_\ell^a\right) = \{i\}\,,$$

where $\mathbf{Pa}_i^z$ *and* $\mathbf{Ch}_i^z$ *are the sets of parents and children of node* $\boldsymbol{z}_i$ *in* $\boldsymbol{G}^z$*, respectively, while* $\mathbf{Ch}_\ell^a$ *is the set of children of* $\boldsymbol{a}_\ell$ *in* $\boldsymbol{G}^a$*.*

We note that the above assumption is slightly different from the original one from Lachapelle et al. [2022], since the intersections run over $\mathbf{Ch}_i^z$, $\mathbf{Pa}_i^z$ and $\mathbf{Pa}_i^a$ instead of over some sets of indexes $\mathcal{I}, \mathcal{J} \subseteq \{1, ..., d_z\}$ and $\mathcal{L} \subseteq \{1, ..., d_a\}$. This slightly simplified criterion is equivalent to the original one, which we now demonstrate for the interested reader.

Proposition 5.11. *Let* $\boldsymbol{G} = [\boldsymbol{G}^z \; \boldsymbol{G}^a] \in \{0,1\}^{d_z \times (d_z + d_a)}$*. The criterion of Assumption 5.5 holds for* $\boldsymbol{G}$ *if and only if the following holds for* $\boldsymbol{G}$*: For all* $i \in \{1, ..., d_z\}$*, there exist sets* $\mathcal{I}, \mathcal{J} \subseteq \{1, ..., d_z\}$ *and* $\mathcal{L} \subseteq \{1, ..., d_a\}$ *such that*

$$\left(\bigcap_{j \in \mathcal{I}} \mathbf{Pa}_j^{\boldsymbol{z}}\right) \cap \left(\bigcap_{j \in \mathcal{J}} \mathbf{Ch}_j^{\boldsymbol{z}}\right) \cap \left(\bigcap_{\ell \in \mathcal{L}} \mathbf{Ch}_\ell^{\boldsymbol{a}}\right) = \{i\}\,,$$

Proof The direction " $\implies$ " is trivial, since we can simply choose $\mathcal{I} := \mathbf{Ch}_i^z$, $\mathcal{J} := \mathbf{Pa}_i^z$ and $\mathcal{L} := \mathbf{Pa}_i^a$.

To show the other direction, we notice that we must have $\mathcal{I} \subseteq \mathbf{Ch}_i^{\boldsymbol{z}}$, $\mathcal{J} \subseteq \mathbf{Pa}_i^z$ and $\mathcal{L} \subseteq \mathbf{Pa}_i^a$, otherwise one of the sets in the intersection would not contain i, contradicting the criterion. Thus, the criterion of Def. 5.5 intersects the same sets or more sets. Moreover these potential additional sets must contain i because of the obvious facts that $j \in \mathbf{Ch}_i^z \iff i \in \mathbf{Pa}_j^{\boldsymbol{z}}$ and $\ell \in \mathbf{Pa}_i^a \iff i \in \mathbf{Ch}_\ell^{\boldsymbol{a}}$, thus they do not change the result of the intersection. ■

To prove Proposition 5.7, we will need the following lemma.

Lemma 5.13. *Let* $\boldsymbol{G} \in \{0,1\}^{m \times n}$ *and* $\mathbf{c}$ *be a diffeomorphism with dependency graph given by* $\boldsymbol{C} \in \{0,1\}^{m \times m}$ *(Definition 5.2). The function* $\mathbf{c}$ *is* $\boldsymbol{G}$*-preserving (Definition 5.12) if and only if*

$$\forall i, \boldsymbol{C}_{i,\cdot} \subseteq \bigcap_{k \in \boldsymbol{G}_{i,\cdot}} \boldsymbol{G}_{\cdot,k}\,.$$

Proof We leverage Proposition 5.3.

$$\boldsymbol{G}_{i,\cdot} \not\subseteq \boldsymbol{G}_{j,\cdot} \iff \exists k \text{ s.t. } \boldsymbol{G}_{i,k} = 1 \text{ and } \boldsymbol{G}_{j,k} = 0 \tag{5.139}$$

$$\iff \exists k \in \boldsymbol{G}_{i,\cdot} \text{ s.t. } j \notin \boldsymbol{G}_{\cdot,k} \tag{5.140}$$

$$\iff j \notin \bigcap_{k \in \boldsymbol{G}_{i,\cdot}} \boldsymbol{G}_{\cdot,k}\,. \tag{5.141}$$

■

Proposition 5.7 (Complete disentanglement as a special case)**.** *Let* $\boldsymbol{\theta} := (\boldsymbol{f}, p, \boldsymbol{G})$ *and* $\hat{\boldsymbol{\theta}} := (\hat{\boldsymbol{f}}, \hat{p}, \hat{\boldsymbol{G}})$ *be two models satisfying Assumptions 5.1, 5.2 & 5.3. If* $\boldsymbol{\theta} \sim_{\text{con}}^{\boldsymbol{z},\boldsymbol{a}} \hat{\boldsymbol{\theta}}$ *and* $\boldsymbol{G}$ *satisfies Assumption 5.5, then* $\boldsymbol{\theta} \sim_{\text{perm}} \hat{\boldsymbol{\theta}}$.

Proof By definition of $\sim_{\text{con}}^{\boldsymbol{z},\boldsymbol{a}}$, we know that the entanglement graph for $(\boldsymbol{f}, \hat{\boldsymbol{f}})$ is given by $\boldsymbol{V} = \boldsymbol{C}\boldsymbol{P}^\top$ where $\boldsymbol{P}$ is a permutation and $\boldsymbol{C}$ is a binary matrix that is $\boldsymbol{G}^a$-preserving, $\boldsymbol{G}^z$-preserving and $(\boldsymbol{G}^z)^\top$-preserving. Using Lemma 5.13, we have that, for all i,

$$\boldsymbol{C}_{i,\cdot} \subseteq \left(\bigcap_{j \in \boldsymbol{G}^{\boldsymbol{z}}_{i,\cdot}} \boldsymbol{G}^{\boldsymbol{z}}_{\cdot,j}\right) \cap \left(\bigcap_{j \in \boldsymbol{G}^{\boldsymbol{z}}_{\cdot,i}} \boldsymbol{G}^{\boldsymbol{z}}_{j,\cdot}\right) \cap \left(\bigcap_{j \in \boldsymbol{G}^{\boldsymbol{a}}_{i,\cdot}} \boldsymbol{G}^{\boldsymbol{a}}_{\cdot,j}\right) \tag{5.142}$$

$$= \left(\bigcap_{j \in \mathbf{Pa}^z_i} \mathbf{Ch}^z_j\right) \cap \left(\bigcap_{j \in \mathbf{Ch}^z_i} \mathbf{Pa}^z_j\right) \cap \left(\bigcap_{\ell \in \mathbf{Pa}^a_i} \mathbf{Ch}^a_\ell\right) \tag{5.143}$$

$$= \{i\}\,. \tag{5.144}$$

Thus $\boldsymbol{C}$ is in fact the identity matrix, and hence $\boldsymbol{\theta} \sim_{\text{perm}} \hat{\boldsymbol{\theta}}$. ■

B. Identifiability theory - Exponential family case

B.1. Technical Lemmas and definitions

We recall the definition of a minimal sufficient statistic in an exponential family, which can be found in Wainwright and Jordan [2008, p. 40].

Definition 5.20 (Minimal sufficient statistic)**.** *Given a parameterized distribution in the exponential family, as in (5.61), we say its sufficient statistic* $\boldsymbol{s}_i$ *is minimal when there is no* $v \neq 0$ *such that* $v^\top \boldsymbol{s}_i(z)$ *is constant for all* $z \in \mathcal{Z}$.

The following Lemma gives a characterization of minimality which will be useful in the proof of Thm. 5.4.

Lemma 5.14 (Characterization of minimal $\boldsymbol{s}$)**.** *A sufficient statistic of an exponential family distribution* $\boldsymbol{s} : \mathcal{Z} \to \mathbb{R}^k$ *is minimal if and only if there exists* $\boldsymbol{z}_{(0)}$, $\boldsymbol{z}_{(1)}$, ..., $\boldsymbol{z}_{(k)}$ *belonging to the support* $\mathcal{Z}$ *such that the following* k*-dimensional vectors are linearly independent:*

$$\boldsymbol{s}(\boldsymbol{z}_{(1)}) - \boldsymbol{s}(\boldsymbol{z}_{(0)}), ..., \boldsymbol{s}(\boldsymbol{z}_{(k)}) - \boldsymbol{s}(\boldsymbol{z}_{(0)})\,. \tag{5.145}$$

Proof. We start by showing the "if" part of the statement. Suppose there exist $\boldsymbol{z}_{(0)}, ..., \boldsymbol{z}_{(k)}$ in $\mathcal{Z}$ such that the vectors of (5.145) are linearly independent. By contradiction, suppose that $\boldsymbol{s}$ is not minimal, i.e. there exist a nonzero vector v and a scalar b such that $v^\top \boldsymbol{s}(z) = b$ for all $z \in \mathcal{Z}$. Notice that $b = v^\top \boldsymbol{s}(\boldsymbol{z}_{(0)})$. Hence, $v^\top(\boldsymbol{s}(\boldsymbol{z}_{(i)}) - \boldsymbol{s}(\boldsymbol{z}_{(0)})) = 0$ for all $i = 1, ..., k$. This can be rewritten in

matrix form as

$$v^\top [\boldsymbol{s}(\boldsymbol{z}_{(1)}) - \boldsymbol{s}(\boldsymbol{z}_{(0)}) \; ... \; \boldsymbol{s}(\boldsymbol{z}_{(k)}) - \boldsymbol{s}(\boldsymbol{z}_{(0)})] = 0\,, \tag{5.146}$$

which implies that the matrix in the above equation is not invertible. This is a contradiction.

We now show the "only if " part of the statement. Suppose that there is no $\boldsymbol{z}_{(0)}$, ..., $\boldsymbol{z}_{(k)}$ such that the vectors of (5.145) are linearly independent. Choose an arbitrary $\boldsymbol{z}_{(0)} \in \mathcal{Z}$. We thus have that $U := \text{span}\{\boldsymbol{s}(z) - \boldsymbol{s}(\boldsymbol{z}_{(0)}) \mid z \in \mathcal{Z}\}$ is a proper subspace of $\mathbb{R}^k$. This means the orthogonal complement of U, $U^\perp$, has dimension 1 or greater. We can thus pick a nonzero vector $v \in U^\perp$ such that $v^\top(\boldsymbol{s}(z) - \boldsymbol{s}(\boldsymbol{z}_0)) = 0$ for all $z \in \mathcal{Z}$, which is to say that $v^\top \boldsymbol{s}(z)$ is constant for all $z \in \mathcal{Z}$, and thus, $\boldsymbol{s}$ is not minimal. ■

B.2. Proof of linear identifiability (Theorem 5.4)

Theorem 5.4 (Conditions for linear identifiability - Adapted from Khemakhem et al. [2020a])**.** *Let $\boldsymbol{\theta} := (\boldsymbol{f}, \boldsymbol{\lambda}, \boldsymbol{G})$ and $\hat{\boldsymbol{\theta}} := (\hat{\boldsymbol{f}}, \hat{\boldsymbol{\lambda}}, \hat{\boldsymbol{G}})$ be two models satisfying Assumptions 5.1, 5.2 & 5.9. Further assume that*

*(1) **[Observational equivalence]** $\boldsymbol{\theta} \sim_{\text{obs}} \hat{\boldsymbol{\theta}}$ (Definition 5.4);*

*(2) **[Minimal sufficient statistics]** For all i, the sufficient statistic $\boldsymbol{s}_i$ is minimal (see below).*

*(3) **[Sufficient variability]** The natural parameter $\boldsymbol{\lambda}$ varies "sufficiently" as formalized by Assumption 5.10 (see below).*

Then, $\boldsymbol{\theta} \sim_{\text{lin}} \hat{\boldsymbol{\theta}}$ (Def. 5.17).

Proof First, we apply Proposition 5.2 to get that

$$\tilde{p}(\boldsymbol{z}^t \mid \boldsymbol{z}^{<t}, \boldsymbol{a}^{<t}) = p(\boldsymbol{v}(\boldsymbol{z}^t) \mid \boldsymbol{v}(\boldsymbol{z}^{<t}), \boldsymbol{a}^{<t}) |\det D\boldsymbol{v}(\boldsymbol{z}^t)|\,, \tag{5.147}$$

Linear relationship between $\boldsymbol{s}(\boldsymbol{f}^{-1}(x))$ and $\boldsymbol{s}(\hat{\boldsymbol{f}}^{-1}(x))$. By taking the logarithm on each sides of (5.147) and expliciting the exponential family form, we get

$$\begin{aligned}
&\sum_{i=1}^{d_z} \log h_i(\boldsymbol{z}_i^t) + \boldsymbol{s}_i(\boldsymbol{z}_i^t)^\top \boldsymbol{\lambda}_i(\boldsymbol{G}_i^z \odot \boldsymbol{z}^{<t}, \boldsymbol{G}_i^a \odot \boldsymbol{a}^{<t}) - \psi_i(\boldsymbol{z}^{<t}, \boldsymbol{a}^{<t}) \qquad (5.148)\\
&= \sum_{i=1}^{d_z} \log h_i(\boldsymbol{v}_i(\boldsymbol{z}^t)) + \boldsymbol{s}_i(\boldsymbol{v}_i(\boldsymbol{z}^t)))^\top \hat{\boldsymbol{\lambda}}_i(\hat{\boldsymbol{G}}_i^z \odot \boldsymbol{v}(\boldsymbol{z}^{<t}), \hat{\boldsymbol{G}}_i^a \odot \boldsymbol{a}^{<t}) - \hat{\psi}_i(\boldsymbol{v}(\boldsymbol{z}^{<t}), \boldsymbol{a}^{<t})\\
&\qquad\qquad + \log|\det D\boldsymbol{v}(\boldsymbol{z}^t)|
\end{aligned}$$

Note that (5.148) holds for all $\boldsymbol{z}^{<t}$ and $\boldsymbol{a}^{<t}$. In particular, we evaluate it at the points given in the assumption of sufficient variability of Thm. 5.4. We evaluate the equation at $(\boldsymbol{z}^t, \boldsymbol{z}_{(r)}, \boldsymbol{a}_{(r)})$ and

$(\boldsymbol{z}^t, \boldsymbol{z}_{(0)}, \boldsymbol{a}_{(0)})$ and take the difference which yields[8]

$$\begin{aligned}
&\sum_{i=1}^{d_z} \boldsymbol{s}_i(\boldsymbol{z}_i^t)^\top [\boldsymbol{\lambda}_i(\boldsymbol{G}_i^z \odot \boldsymbol{z}_{(r)}, \boldsymbol{G}_i^a \odot \boldsymbol{a}_{(r)}) - \boldsymbol{\lambda}_i(\boldsymbol{G}_i^z \odot \boldsymbol{z}_{(0)}, \boldsymbol{G}_i^a \odot \boldsymbol{a}_{(0)})] - \psi_i(\boldsymbol{z}_{(r)}, \boldsymbol{a}_{(r)}) + \psi_i(\boldsymbol{z}_{(0)}, \boldsymbol{a}_{(0)}) \\
=&\sum_{i=1}^{d_z} \boldsymbol{s}_i(\boldsymbol{v}_i(\boldsymbol{z}^t))^\top [\hat{\boldsymbol{\lambda}}_i(\hat{\boldsymbol{G}}_i^z \odot \boldsymbol{v}(\boldsymbol{z}_{(r)}), \hat{\boldsymbol{G}}_i^a \odot \boldsymbol{a}_{(r)}) - \hat{\boldsymbol{\lambda}}_i(\hat{\boldsymbol{G}}_i^z \odot \boldsymbol{v}(\boldsymbol{z}_{(0)}), \hat{\boldsymbol{G}}_i^a \odot \boldsymbol{a}_{(0)})] \\
&\qquad - \hat{\psi}_i(\boldsymbol{v}(\boldsymbol{z}_{(r)}), \boldsymbol{a}_{(r)}) + \hat{\psi}_i(\boldsymbol{v}(\boldsymbol{z}_{(0)}), \boldsymbol{a}_{(0)})
\end{aligned} \tag{5.149}$$

We regroup all normalization constants ψ into a term $d(\boldsymbol{z}_{(r)}, \boldsymbol{z}_{(0)}, \boldsymbol{a}_{(r)}, \boldsymbol{a}_{(0)})$ and write

$$\begin{aligned}
&\boldsymbol{s}(\boldsymbol{z}^t)^\top [\boldsymbol{\lambda}(\boldsymbol{z}_{(r)}, \boldsymbol{a}_{(r)}) - \boldsymbol{\lambda}(\boldsymbol{z}_{(0)}, \boldsymbol{a}_{(0)})] \\
=&\boldsymbol{s}(\boldsymbol{v}(\boldsymbol{z}^t))^\top [\hat{\boldsymbol{\lambda}}(\boldsymbol{v}(\boldsymbol{z}_{(r)}), \boldsymbol{a}_{(r)}) - \hat{\boldsymbol{\lambda}}(\boldsymbol{v}(\boldsymbol{z}_{(0)}), \boldsymbol{a}_{(0)})] + d(\boldsymbol{z}_{(r)}, \boldsymbol{z}_{(0)}, \boldsymbol{a}_{(r)}, \boldsymbol{a}_{(0)})\,.
\end{aligned} \tag{5.150}$$

Define

$$\boldsymbol{w}_{(r)} := \boldsymbol{\lambda}(\boldsymbol{z}_{(r)}, \boldsymbol{a}_{(r)}) - \boldsymbol{\lambda}(\boldsymbol{z}_{(0)}, \boldsymbol{a}_{(0)}) \tag{5.151}$$

$$\hat{\boldsymbol{w}}_{(r)} := \hat{\boldsymbol{\lambda}}(\boldsymbol{v}(\boldsymbol{z}_{(r)}), \boldsymbol{a}_{(r)}) - \hat{\boldsymbol{\lambda}}(\boldsymbol{v}(\boldsymbol{z}_{(0)}), \boldsymbol{a}_{(0)}) \tag{5.152}$$

$$d_{(r)} := d(\boldsymbol{z}_{(r)}, \boldsymbol{z}_{(0)}, \boldsymbol{a}_{(r)}, \boldsymbol{a}_{(0)})\,, \tag{5.153}$$

which yields

$$\boldsymbol{s}(\boldsymbol{z}^t)^\top \boldsymbol{w}_{(r)} = \boldsymbol{s}(\boldsymbol{v}(\boldsymbol{z}^t))^\top \hat{\boldsymbol{w}}_{(r)} + d_{(r)}\,. \tag{5.154}$$

We can regroup the $\boldsymbol{w}_{(r)}$ into a matrix and the $d_{(r)}$ into a vector:

$$\boldsymbol{W} := [\boldsymbol{w}_{(1)} \ldots \boldsymbol{w}_{(kd_z)}] \in \mathbb{R}^{kd_z \times kd_z} \tag{5.155}$$

$$\hat{\boldsymbol{W}} := [\hat{\boldsymbol{w}}_{(1)} \ldots \hat{\boldsymbol{w}}_{(kd_z)}] \in \mathbb{R}^{kd_z \times kd_z} \tag{5.156}$$

$$\boldsymbol{d} := [d_{(1)} \ldots d_{(kd_z)}] \in \mathbb{R}^{1 \times kd_z}\,. \tag{5.157}$$

Since (5.154) holds for all $1 \leq p \leq kd_z$, we can write

$$\boldsymbol{s}(\boldsymbol{z}^t)^\top \boldsymbol{W} = \boldsymbol{s}(\boldsymbol{v}(\boldsymbol{z}^t))^\top \hat{\boldsymbol{W}} + \boldsymbol{d}\,. \tag{5.158}$$

Note that $\boldsymbol{W}$ is invertible by the assumption of variability, hence

$$\boldsymbol{s}(\boldsymbol{z}^t)^\top = \boldsymbol{s}(\boldsymbol{v}(\boldsymbol{z}^t))^\top \hat{\boldsymbol{W}} \boldsymbol{W}^{-1} + \boldsymbol{d}\boldsymbol{W}^{-1}\,. \tag{5.159}$$

Let $\boldsymbol{b} := (\boldsymbol{d}\boldsymbol{W}^{-1})^\top$ and $\boldsymbol{L} := (\hat{\boldsymbol{W}}\boldsymbol{W}^{-1})^\top$. We can thus rewrite as

$$\boldsymbol{s}(\boldsymbol{z}^t) = \boldsymbol{L}\boldsymbol{s}(\boldsymbol{v}(\boldsymbol{z}^t)) + \boldsymbol{b}\,. \tag{5.160}$$

[8]Note that $\boldsymbol{z}_{(0)}$ and $\boldsymbol{z}_{(r)}$ can have different dimensionalities if they come from different time steps. It is not an issue to combine equations from different time steps, since (5.148) holds for all values of t, $\boldsymbol{z}^t$, $\boldsymbol{z}^{<t}$ and $\boldsymbol{a}^{<t}$.

Invertibility of $\boldsymbol{L}$. We now show that $\boldsymbol{L}$ is invertible. By Lemma 5.14, the fact that the $\boldsymbol{s}_i$ are minimal is equivalent to, for all $i \in \{1, ..., d_z\}$, having elements $\boldsymbol{z}_i^{(0)}$, ..., $\boldsymbol{z}_i^{(k)}$ in $\mathcal{Z}$ such that the family of vectors

$$\boldsymbol{s}_i(\boldsymbol{z}_i^{(1)}) - \boldsymbol{s}_i(\boldsymbol{z}_i^{(0)}),\ \dots\ , \boldsymbol{s}_i(\boldsymbol{z}_i^{(k)}) - \boldsymbol{s}_i(\boldsymbol{z}_i^{(0)}) \tag{5.161}$$

is linearly independent. Define

$$\boldsymbol{z}^{(0)} := [\boldsymbol{z}_1^{(0)} \dots \boldsymbol{z}_{d_z}^{(0)}]^\top \in \mathbb{R}^{d_z} \tag{5.162}$$

For all $i \in \{1, ..., d_z\}$ and all $p \in \{1, ..., k\}$, define the vectors

$$\boldsymbol{z}^{(p,i)} := [\boldsymbol{z}_1^{(0)} \dots \boldsymbol{z}_{i-1}^{(0)}\ \boldsymbol{z}_i^{(p)}\ \boldsymbol{z}_{i+1}^{(0)} \dots \boldsymbol{z}_{d_z}^{(0)}]^\top \in \mathbb{R}^{d_z}\,. \tag{5.163}$$

For a specific $1 \leq p \leq k$ and $i \in \{1, ..., d_z\}$, we can take the following difference based on (5.160)

$$\boldsymbol{s}(\boldsymbol{z}^{(p,i)}) - \boldsymbol{s}(\boldsymbol{z}^{(0)}) = \boldsymbol{L}[\boldsymbol{s}(\boldsymbol{v}(\boldsymbol{z}^{(p,i)})) - \boldsymbol{s}(\boldsymbol{v}(\boldsymbol{z}^{(0)}))]\,, \tag{5.164}$$

where the left hand side is a vector filled with zeros except for the block corresponding to $\boldsymbol{s}_i(\boldsymbol{z}_i^{(p,i)}) - \boldsymbol{s}_i(\boldsymbol{z}_i^{(0)})$. Let us define

$$\begin{aligned}
\Delta\boldsymbol{s}^{(i)} &:= [\boldsymbol{s}(\boldsymbol{z}^{(1,i)}) - \boldsymbol{s}(\boldsymbol{z}^{(0)})\ \dots\ \boldsymbol{s}(\boldsymbol{z}^{(k,i)}) - \boldsymbol{s}(\boldsymbol{z}^{(0)})] \in \mathbb{R}^{kd_z \times k} \\
\Delta\hat{\boldsymbol{s}}^{(i)} &:= [\boldsymbol{s}(\boldsymbol{v}(\boldsymbol{z}^{(1,i)})) - \boldsymbol{s}(\boldsymbol{v}(\boldsymbol{z}^{(0)}))\ \dots\ \boldsymbol{s}(\boldsymbol{v}(\boldsymbol{z}^{(k,i)})) - \boldsymbol{s}(\boldsymbol{v}(\boldsymbol{z}^{(0)}))] \in \mathbb{R}^{kd_z \times k}\,.
\end{aligned}$$

Note that the columns of $\Delta\boldsymbol{s}^{(i)}$ are linearly independent and all rows are filled with zeros except for the block of rows $\{(i-1)k+1, ..., ik\}$. We can thus rewrite (5.164) in matrix form

$$\Delta\boldsymbol{s}^{(i)} = \boldsymbol{L}\Delta\hat{\boldsymbol{s}}^{(i)}\,. \tag{5.165}$$

We can regroup these equations for every i by doing

$$[\Delta\boldsymbol{s}^{(1)}\ \dots\ \Delta\boldsymbol{s}^{(d_z)}] = \boldsymbol{L}[\Delta\hat{\boldsymbol{s}}^{(1)}\ \dots\ \Delta\hat{\boldsymbol{s}}^{(d_z)}]\,. \tag{5.166}$$

Notice that the newly formed matrix on the left hand side has size $kd_z \times kd_z$ and is block diagonal. Since every block is invertible, the left hand side of (5.166) is an invertible matrix, which in turn implies that $\boldsymbol{L}$ is invertible. This completes the proof. ■

B.3. Proof of Theorem 5.5

Lemma 5.15. *Let $\boldsymbol{\theta} := (\boldsymbol{f}, p, \boldsymbol{G})$ satisfy Assumptions 5.1, 5.2, 5.3, 5.4 & 5.9 and let $q := \log p$. Then*

$$D_z^t q(\boldsymbol{z}^t \mid \boldsymbol{z}^{<t}, \boldsymbol{a}^{<t}) = \boldsymbol{\lambda}(\boldsymbol{z}^{<t}, \boldsymbol{a}^{<t})^\top D\boldsymbol{s}(\boldsymbol{z}^t) + D(\log h)(\boldsymbol{z}^t) \tag{5.167}$$

$$H_{z,a}^{t,\tau} q(\boldsymbol{z}^t \mid \boldsymbol{z}^{<t}, \boldsymbol{a}^{<t}) = D\boldsymbol{s}(\boldsymbol{z}^t)^\top D_a^\tau \boldsymbol{\lambda}(\boldsymbol{z}^{<t}, \boldsymbol{a}^{<t}) \tag{5.168}$$

$$H_{z,z}^{t,\tau} q(\boldsymbol{z}^t \mid \boldsymbol{z}^{<t}, \boldsymbol{a}^{<t}) = D\boldsymbol{s}(\boldsymbol{z}^t)^\top D_z^\tau \boldsymbol{\lambda}(\boldsymbol{z}^{<t}, \boldsymbol{a}^{<t})\,. \tag{5.169}$$

Proof We have

$$\log p(\boldsymbol{z}^t \mid \boldsymbol{z}^{<t}, \boldsymbol{a}^{<t}) := \log h(\boldsymbol{z}^t) + \boldsymbol{s}(\boldsymbol{z}^t)^\top \boldsymbol{\lambda}(\boldsymbol{z}^{<t}, \boldsymbol{a}^{<t}) - \psi(\boldsymbol{z}^{<t}, \boldsymbol{a}^{<t}) \tag{5.170}$$

$$\log h(\boldsymbol{z}^t) + \boldsymbol{\lambda}(\boldsymbol{z}^{<t}, \boldsymbol{a}^{<t})^\top \boldsymbol{s}(\boldsymbol{z}^t) - \psi(\boldsymbol{z}^{<t}, \boldsymbol{a}^{<t})\,. \tag{5.171}$$

We can differentiate the above w.r.t. $\boldsymbol{z}^t$ to get

$$D_z^t q(\boldsymbol{z}^t \mid \boldsymbol{z}^{<t}, \boldsymbol{a}^{<t}) = \boldsymbol{\lambda}(\boldsymbol{z}^{<t}, \boldsymbol{a}^{<t})^\top D\boldsymbol{s}(\boldsymbol{z}^t) + D(\log h)(\boldsymbol{z}^t) \tag{5.172}$$

Differentiating the above w.r.t. $\boldsymbol{z}^\tau$ or $\boldsymbol{a}^\tau$ yields the desired result. ■

Theorem 5.5 (Disentanglement via sparse temporal dependencies in exponential families)**.** *Let $\boldsymbol{\theta} := (\boldsymbol{f}, \boldsymbol{\lambda}, \boldsymbol{G})$ and $\hat{\boldsymbol{\theta}} := (\hat{\boldsymbol{f}}, \hat{\boldsymbol{\lambda}}, \hat{\boldsymbol{G}})$ be two models satisfying Assumptions 5.1, 5.2, 5.3, 5.4, 5.9 as well as all assumptions of Theorem 5.4. Further suppose that*

(1) The sufficient statistic $\boldsymbol{s}$ is d_z-dimensional ($k = 1$) and is a diffeomorphism from $\mathbb{R}^{d_z}$ to $\boldsymbol{s}(\mathbb{R}^{d_z})$;

*(2) **[Sufficient influence of $\boldsymbol{z}$]** The Jacobian of the ground-truth transition function $\boldsymbol{\lambda}$ with respect to $\boldsymbol{z}$ varies "sufficiently", as formalized in Assumption 5.11;*

Then, there exists a permutation matrix $\boldsymbol{P}$ such that $\boldsymbol{P}\boldsymbol{G}^z\boldsymbol{P}^\top \subseteq \hat{\boldsymbol{G}}^z$. Further assume that

*(3) **[Sparsity regularization]** $||\hat{\boldsymbol{G}}^z||_0 \leq ||\boldsymbol{G}^z||_0$;*

Then, $\boldsymbol{\theta} \sim_{\text{con}}^z \hat{\boldsymbol{\theta}}$ (Def. 5.14) & $\boldsymbol{\theta} \sim_{\text{lin}} \hat{\boldsymbol{\theta}}$ (Def. 5.17), which together implies that

$$\boldsymbol{v}(\boldsymbol{z}) = \boldsymbol{s}^{-1}(\boldsymbol{C}\boldsymbol{P}^\top \boldsymbol{s}(\boldsymbol{z}) + \boldsymbol{b})\,,$$

where $\boldsymbol{b} \in \mathbb{R}^{d_z}$ and $\boldsymbol{C} \in \mathbb{R}^{d_z \times d_z}$ is invertible, $\boldsymbol{G}^z$- and $(\boldsymbol{G}^z)^\top$-preserving (Definition 5.11).

Proof Recall the equation we derived in Section 5.3.1:

$$H_{z,z}^{t,\tau} \hat{q}(\boldsymbol{z}^t \mid \boldsymbol{z}^{<t}, \boldsymbol{a}^{<t}) = D\boldsymbol{v}(\boldsymbol{z}^t)^\top H_{z,z}^{t,\tau} q(\boldsymbol{v}(\boldsymbol{z}^t) \mid \boldsymbol{v}(\boldsymbol{z}^{<t}), \boldsymbol{a}^{<t}) D\boldsymbol{v}(\boldsymbol{z}^\tau)\,. \tag{5.173}$$

Using Lemma 5.15, we get that

$$D\boldsymbol{s}(\boldsymbol{z}^t)^\top D_z^\tau \hat{\boldsymbol{\lambda}}(\boldsymbol{z}^{<t}, \boldsymbol{a}^{<t}) = D\boldsymbol{v}(\boldsymbol{z}^t)^\top D\boldsymbol{s}(\boldsymbol{v}(\boldsymbol{z}^t))^\top D_z^\tau \boldsymbol{\lambda}(\boldsymbol{v}(\boldsymbol{z}^{<t}), \boldsymbol{a}^{<t}) D\boldsymbol{v}(\boldsymbol{z}^\tau)\,. \tag{5.174}$$

Note that Assumption 5.3 requires that $D_z^\tau \boldsymbol{\lambda}(\boldsymbol{z}^{<t}, \boldsymbol{a}^{<t}) \subseteq \boldsymbol{G}^z$ and that $D_z^\tau \hat{\boldsymbol{\lambda}}(\boldsymbol{z}^{<t}, \boldsymbol{a}^{<t}) \subseteq \hat{\boldsymbol{G}}^z$. Theorem 5.4 implies that there exist an invertible matrix $\boldsymbol{L} \in \mathbb{R}^{d_z \times d_z}$ and a vector $\boldsymbol{b} \in \mathbb{R}^{d_z}$ such that

$$\boldsymbol{v}(\boldsymbol{z}) = \boldsymbol{s}^{-1}(\boldsymbol{L}\boldsymbol{s}(\boldsymbol{z}) + \boldsymbol{b})\,. \tag{5.175}$$

Taking the derivative of the above w.r.t. $\boldsymbol{z}$, we obtain

$$\begin{aligned} D\boldsymbol{v}(\boldsymbol{z}) &= D\boldsymbol{s}^{-1}(\boldsymbol{L}\boldsymbol{s}(\boldsymbol{z}) + \boldsymbol{b})\boldsymbol{L}D\boldsymbol{s}(\boldsymbol{z}) && (5.176)\\ &= D\boldsymbol{s}^{-1}(\boldsymbol{s}(\boldsymbol{v}(\boldsymbol{z})))\boldsymbol{L}D\boldsymbol{s}(\boldsymbol{z}) && (5.177)\\ &= D\boldsymbol{s}(\boldsymbol{v}(\boldsymbol{z}))^{-1}\boldsymbol{L}D\boldsymbol{s}(\boldsymbol{z})\,, && (5.178) \end{aligned}$$

where we used $\boldsymbol{s}(\boldsymbol{v}(\boldsymbol{z})) = \boldsymbol{L}\boldsymbol{s}(\boldsymbol{z}) + \boldsymbol{b}$ to go from the first to the second line and used the inverse function theorem to go from the second to the third line. Plugging (5.178) into (5.174) yields

$$\begin{aligned} &D\boldsymbol{s}(\boldsymbol{z}^t)^\top D_z^\tau \hat{\boldsymbol{\lambda}}(\boldsymbol{z}^{<t}, \boldsymbol{a}^{<t}) && (5.179)\\ &= D\boldsymbol{s}(\boldsymbol{z}^t)^\top \boldsymbol{L}^\top D\boldsymbol{s}(\boldsymbol{v}(\boldsymbol{z}^t))^{-\top} D\boldsymbol{s}(\boldsymbol{v}(\boldsymbol{z}^t))^\top D_z^\tau \boldsymbol{\lambda}(\boldsymbol{v}(\boldsymbol{z}^{<t}), \boldsymbol{a}^{<t}) D\boldsymbol{s}(\boldsymbol{v}(\boldsymbol{z}^\tau))^{-1}\boldsymbol{L}D\boldsymbol{s}(\boldsymbol{z}^\tau) && (5.180)\\ &= D\boldsymbol{s}(\boldsymbol{z}^t)^\top \boldsymbol{L}^\top D_z^\tau \boldsymbol{\lambda}(\boldsymbol{v}(\boldsymbol{z}^{<t}), \boldsymbol{a}^{<t}) D\boldsymbol{s}(\boldsymbol{v}(\boldsymbol{z}^\tau))^{-1}\boldsymbol{L}D\boldsymbol{s}(\boldsymbol{z}^\tau)\,, && (5.181) \end{aligned}$$

which implies

$$\begin{aligned} &D\boldsymbol{s}(\boldsymbol{z}^t)^\top D_z^\tau \hat{\boldsymbol{\lambda}}(\boldsymbol{z}^{<t}, \boldsymbol{a}^{<t}) = D\boldsymbol{s}(\boldsymbol{z}^t)^\top \boldsymbol{L}^\top D_z^\tau \boldsymbol{\lambda}(\boldsymbol{v}(\boldsymbol{z}^{<t}), \boldsymbol{a}^{<t}) D\boldsymbol{s}(\boldsymbol{v}(\boldsymbol{z}^\tau))^{-1}\boldsymbol{L}D\boldsymbol{s}(\boldsymbol{z}^\tau) && (5.182)\\ &D_z^\tau \hat{\boldsymbol{\lambda}}(\boldsymbol{z}^{<t}, \boldsymbol{a}^{<t}) D\boldsymbol{s}(\boldsymbol{z}^\tau)^{-1} = \boldsymbol{L}^\top D_z^\tau \boldsymbol{\lambda}(\boldsymbol{v}(\boldsymbol{z}^{<t}), \boldsymbol{a}^{<t}) D\boldsymbol{s}(\boldsymbol{v}(\boldsymbol{z}^\tau))^{-1}\boldsymbol{L}\,, && (5.183) \end{aligned}$$

where we right- and left-multiplied by $D\boldsymbol{s}(\boldsymbol{z}^t)^{-\top}$ and $D\boldsymbol{s}(\boldsymbol{z}^\tau)$, respectively. Let us define

$$\Lambda(\gamma) := D_z^\tau \boldsymbol{\lambda}(\boldsymbol{v}(\boldsymbol{z}^{<t}), \boldsymbol{a}^{<t}) D\boldsymbol{s}(\boldsymbol{v}(\boldsymbol{z}^\tau))^{-1} \quad \hat{\Lambda}(\gamma) := D_z^\tau \hat{\boldsymbol{\lambda}}(\boldsymbol{v}(\boldsymbol{z}^{<t}), \boldsymbol{a}^{<t}) D\boldsymbol{s}(\boldsymbol{v}(\boldsymbol{z}^\tau))^{-1}\,,$$

where $\gamma = (t, \tau, \boldsymbol{z}^{<t}, \boldsymbol{a}^{<t})$. Note that because $D\boldsymbol{s}$ is diagonal, we have that $\Lambda(\gamma) \subseteq \boldsymbol{G}^z$ and $\hat{\Lambda}(\gamma) \subseteq \hat{\boldsymbol{G}}^z$. Using this notation, we can rewrite (5.183) as

$$\underbrace{\hat{\Lambda}(\gamma)}_{\subseteq \hat{\boldsymbol{G}}^z} = \boldsymbol{L}^\top \underbrace{\Lambda(\gamma)}_{\subseteq \boldsymbol{G}^z} \boldsymbol{L}\,. \tag{5.184}$$

Thanks to Assumption 5.11, we can apply the same argument as in Theorem 5.5 to show that $\boldsymbol{L} = \boldsymbol{C}\boldsymbol{P}^\top$ where $\boldsymbol{C}$ is a matrix that is both $\boldsymbol{G}^z$-preserving and $(\boldsymbol{G}^z)^\top$-preserving, as desired. ■

B.4. Relating with sufficient influence assumptions of Lachapelle et al. [2022]

In this section, we relate the nonparametric sufficient influence assumptions of this work, i.e. Assumptions 5.7 & 5.8, to the analogous assumptions of Lachapelle et al. [2022] for exponential families, i.e. Assumptions 5.11 & 5.12, the latter of which we recall below.

Assumption 5.12 (Sufficient influence of $\boldsymbol{a}$ [Lachapelle et al., 2022])**.** *Assume $k = 1$, i.e. the sufficient statistics $\boldsymbol{s}_i$ are one-dimensional. For all $\ell \in \{1, ..., d_a\}$, there exist $\{(\boldsymbol{z}_{(r)}, \boldsymbol{a}_{(r)}, \epsilon_{(r)}, \tau_{(r)})\}_{r=1}^{|\mathbf{Ch}_\ell^{\boldsymbol{a}}|}$ belonging to their respective support such that*

$$\operatorname{span}\left\{\boldsymbol{\lambda}(\boldsymbol{z}_{(r)}, \boldsymbol{a}_{(r)} + \epsilon_{(r)}\boldsymbol{E}^{(\ell,\tau)}) - \boldsymbol{\lambda}(\boldsymbol{z}_{(r)}, \boldsymbol{a}_{(r)})\right\}_{r=1}^{|\mathbf{Ch}_\ell^{\boldsymbol{a}}|} = \mathbb{R}^{d_z}_{\mathbf{Ch}_\ell^{\boldsymbol{a}}},$$

where $\epsilon \in \mathbb{R}$ and $\boldsymbol{E}^{(\ell,\tau)} \in \mathbb{R}^{d_a \times t}$ is the one-hot matrix with the entry (ℓ, τ) set to one.

The following proposition shows that, when the exponential family holds with $k = 1$, we have that (i) for the "sufficient influence of $\boldsymbol{a}$" assumptions, the nonparametric and exponential family versions are actually equivalent, and (ii) for the "sufficient influence of $\boldsymbol{z}$" assumptions, the nonparametric version implies the exponential family version.

Proposition 5.12 (Sufficient influence assumptions: nonparametric v.s. exponential)**.** *Let the parameter $\boldsymbol{\theta} := (\boldsymbol{f}, p, \boldsymbol{G})$ satisfy Assumptions 5.1, 5.2, 5.3 & 5.9. Further assume that $k = 1$ and that $D\boldsymbol{s}(\boldsymbol{z}) \in \mathbb{R}^{d_z \times d_z}$ is invertible everywhere. Then,*

Sufficient influence of $\boldsymbol{a}$: Assumption 5.7 (nonparametric) $\iff$ Assumption 5.12 (exponential family)

Sufficient influence of $\boldsymbol{z}$: Assumption 5.8 (nonparametric) $\implies$ Assumption 5.11 (exponential family)

Proof We start by proving the first equivalence for the sufficient influence of $\boldsymbol{a}$ assumptions. By using Lemma 5.15 we see that

$$\operatorname{span}\left\{D_z^{t_{(r)}} \log p(\boldsymbol{z} \mid \boldsymbol{z}_{(r)}, \boldsymbol{a}_{(r)} + \epsilon_{(r)}\boldsymbol{E}^{(\ell,\tau_{(r)})}) - D_z^{t_{(r)}} \log p(\boldsymbol{z} \mid \boldsymbol{z}_{(r)}, \boldsymbol{a}_{(r)})\right\}_{r=1}^{|\mathbf{Ch}_\ell^{\boldsymbol{a}}|} \tag{5.185}$$

$$= \operatorname{span}\left\{D\boldsymbol{s}(\boldsymbol{z})^\top \boldsymbol{\lambda}(\boldsymbol{z}_{(r)}, \boldsymbol{a}_{(r)} + \epsilon_{(r)}\boldsymbol{E}^{(\ell,\tau_{(r)})}) - D\boldsymbol{s}(\boldsymbol{z})^\top \boldsymbol{\lambda}(\boldsymbol{z}_{(r)}, \boldsymbol{a}_{(r)})\right\}_{r=1}^{|\mathbf{Ch}_\ell^{\boldsymbol{a}}|} \tag{5.186}$$

$$= D\boldsymbol{s}(\boldsymbol{z})^\top \operatorname{span}\left\{\boldsymbol{\lambda}(\boldsymbol{z}_{(r)}, \boldsymbol{a}_{(r)} + \epsilon_{(r)}\boldsymbol{E}^{(\ell,\tau_{(r)})}) - \boldsymbol{\lambda}(\boldsymbol{z}_{(r)}, \boldsymbol{a}_{(r)})\right\}_{r=1}^{|\mathbf{Ch}_\ell^{\boldsymbol{a}}|}. \tag{5.187}$$

We start by showing " $\impliedby$ ". Assumption 5.12 implies that (5.187) is equal to $D\boldsymbol{s}(\boldsymbol{z}^t)^\top \mathbb{R}^{d_z}_{\mathbf{Ch}_\ell^{\boldsymbol{a}}}$ which is equal to $\mathbb{R}^{d_z}_{\mathbf{Ch}_\ell^{\boldsymbol{a}}}$ since $D\boldsymbol{s}(\boldsymbol{z}^t)$ is invertible everywhere and is diagonal. To show " $\implies$ ", we can apply the same argument.

We now show that Assumption 5.8 implies Assumption 5.11. we again use Lemma 5.15 and see that

$$\mathbb{R}^{d_z}_{\boldsymbol{G}^z} = \operatorname{span}\left\{H_{z,z}^{t_{(r)},\tau_{(r)}} \log p(\boldsymbol{z} \mid \boldsymbol{z}_{(r)}, \boldsymbol{a}_{(r)})\right\}_{r=1}^{||\boldsymbol{G}^z||_0} \tag{5.188}$$

$$= \operatorname{span}\left\{D\boldsymbol{s}(\boldsymbol{z})^\top D_z^{\tau_{(r)}} \boldsymbol{\lambda}(\boldsymbol{z}_{(r)}, \boldsymbol{a}_{(r)})\right\}_{r=1}^{||\boldsymbol{G}^z||_0} \tag{5.189}$$

$$= D\boldsymbol{s}(\boldsymbol{z})^\top \operatorname{span}\left\{D_z^{\tau_{(r)}} \boldsymbol{\lambda}(\boldsymbol{z}_{(r)}, \boldsymbol{a}_{(r)}) D\boldsymbol{s}(\boldsymbol{z})^{-1}\right\}_{r=1}^{||\boldsymbol{G}^z||_0} D\boldsymbol{s}(\boldsymbol{z}). \tag{5.190}$$

Now recall that, in Assumption 5.8, we had that $\boldsymbol{z} = \boldsymbol{z}^{\tau_{(r)}}$ for all $r = 1, ..., ||\boldsymbol{G}^z||_0$, which allows us to write

$$\mathbb{R}^{d_z}_{\boldsymbol{G}^z} = D\boldsymbol{s}(\boldsymbol{z})^\top \operatorname{span}\left\{D_z^{\tau_{(r)}} \boldsymbol{\lambda}(\boldsymbol{z}_{(r)}, \boldsymbol{a}_{(r)}) D\boldsymbol{s}(\boldsymbol{z}^{\tau_{(r)}})^{-1}\right\}_{r=1}^{||\boldsymbol{G}^z||_0} D\boldsymbol{s}(\boldsymbol{z}), \tag{5.191}$$

which implies

$$\text{span}\left\{D_z^{\tau(r)}\boldsymbol{\lambda}(\boldsymbol{z}_{(r)},\boldsymbol{a}_{(r)})D\boldsymbol{s}(\boldsymbol{z}^{\tau(r)})^{-1}\right\}_{r=1}^{||\boldsymbol{G}^z||_0} = D\boldsymbol{s}(\boldsymbol{z})^{-\top}\mathbb{R}^{d_z}_{\boldsymbol{G}^z}D\boldsymbol{s}(\boldsymbol{z})^{-1} = \mathbb{R}^{d_z}_{\boldsymbol{G}^z}\,, \tag{5.192}$$

where the last equality holds because $D\boldsymbol{s}(\boldsymbol{z})$ is diagonal and invertible everywhere. ■

C. Experiments

C.1. Synthetic datasets

We now provide a detailed description of the synthetic datasets used in experiments of Section 5.8.

For all experiments, the dimensionality of $\boldsymbol{x}^t$ is $d_x = 20$ and the ground-truth $\boldsymbol{f}$ is a random neural network with three hidden layers of 20 units with Leaky-ReLU activations with negative slope of 0.2. The weight matrices are sampled according to a 0-1 Gaussian distribution and, to make sure $\boldsymbol{f}$ is injective as assumed in all theorems of this paper, we orthogonalize its columns. Inspired by typical weight initialization in NN [Glorot and Bengio, 2010b], we rescale the weight matrices by $\sqrt{\frac{2}{1+0.2^2}}\sqrt{\frac{2}{d_{in}+d_{out}}}$. The standard deviation of the Gaussian noise added to $\boldsymbol{f}(\boldsymbol{z}^t)$ is set to $\sigma = 10^{-2}$ throughout. Since the goal of the experiments is to validate our identifiability results, which assume infinite data, all datasets considered here are very large: 1 million examples.

We now present the different choices of ground-truth $p(\boldsymbol{z}^t \mid \boldsymbol{z}^{<t}, \boldsymbol{a}^{<t})$ we explored in our experiments. In all cases considered (except the experiment with $k = 2$ of Table 5.3), it is a Gaussian with covariance $0.0001I$ independent of $(\boldsymbol{z}^{<t}, \boldsymbol{a}^{<t})$ and a mean given by some function $\mu(\boldsymbol{z}^{t-1}, \boldsymbol{a}^{t-1})$. Notice that we hence are in the case where $k = 1$ with monotonic sufficient statistics, which is not covered by the theory of Khemakhem et al. [2020a]. Throughout, we set $d_z = 10$ and, unless explicitly specified otherwise, we set $d_a = 10$. In all *Time* datasets, sequences have length $T = 2$. In *Action* datasets, the value of T has no consequence since we assume there is no time dependence.

C.1.1. Datasets satisfying graphical criterion. The datasets of this section satisfy the graphical criterion of Section 5.3.6. This means our theory predicts complete disentanglement (Definition 5.7). Unless specified otherwise, all datasets satisfy their respective sufficient influence assumptions (Section 5.3.7). These can be checked using Remark 5.5 combined with standard facts about independence of the sine and cosine functions.

ActionDiag (Figure 5.5). In this dataset, $d_a = d_x$ and the connectivity matrix between $\boldsymbol{a}^{t-1}$ and $\boldsymbol{z}^t$ is diagonal, which trivially implies that the graphical criterion of Section 5.3.6 is satisfied. The

mean function is given by

$$\mu(\boldsymbol{z}^{t-1}, \boldsymbol{a}^{t-1}) := \sin(\boldsymbol{a}^{t-1}) \,,$$

where $\sin$ is applied element-wise. Moreover, the components of the action vector $\boldsymbol{a}^{t-1}$ are sampled independently and uniformly between -2 and 2. The same sampling scheme is used for all following datasets. One can check that the sufficient influence assumption (Assumption 5.6) holds.

ActionNonDiag (Figure 5.5). We consider a case where the graphical criterion of Section 5.3.6 is satisfied non-trivially. Let

$$\boldsymbol{G}^a := \begin{pmatrix} 1 & & & & 1 \\ 1 & 1 & & & \\ & 1 & \ddots & & \\ & & \ddots & 1 & \\ & & & 1 & 1 \end{pmatrix} \tag{5.193}$$

be the adjacency matrix between $\boldsymbol{a}^{t-1}$ and $\boldsymbol{z}^t$. The ith row, denoted by $\boldsymbol{G}^a_i$, corresponds to parents of $\boldsymbol{z}^t_i$ in $\boldsymbol{a}^{t-1}$. Note that it is analogous to the graph depicted in Figure 5.4, which satisfies the graphical criterion. The mean function is given by

$$\mu(\boldsymbol{z}^{t-1}, \boldsymbol{a}^{t-1}) := \begin{bmatrix} \boldsymbol{G}^{\boldsymbol{a}}_1 \cdot \sin(\frac{3}{\pi}\boldsymbol{a}^{t-1}) \\ \boldsymbol{G}^{\boldsymbol{a}}_2 \cdot \sin(\frac{4}{\pi}\boldsymbol{a}^{t-1} + 1) \\ \vdots \\ \boldsymbol{G}^{\boldsymbol{a}}_{d_z} \cdot \sin(\frac{d_z+2}{\pi}\boldsymbol{a}^{t-1} + d_z - 1) \end{bmatrix} . \tag{5.194}$$

One can check that the sufficient influence assumption (Assumption 5.6) holds, thanks to the independence of sines with different frequencies.

ActionNonDiag$_{\text{NoSuffInf}}$ (Table 5.3). This dataset has the same ground truth adjacency matrix as the above dataset (5.193), but a different transition function which does not satisfy the assumption of sufficient influence (Section 5.6). We sampled a matrix $\boldsymbol{W}$ with independent Normal 0-1 entries. The mean function is thus

$$\mu(\boldsymbol{z}^{t-1}, \boldsymbol{a}^{t-1}) := (\boldsymbol{G}^a \odot \boldsymbol{W})\boldsymbol{a}^{t-1} \,, \tag{5.195}$$

where $\odot$ is the Hadamard product (a.k.a. element-wise product).

ActionNonDiag$_{k=2}$ (Table 5.3). This dataset has the "double diagonal" adjacency matrix of (5.193) and the same mean function of (5.194), but the variance of $\boldsymbol{z}^t$ (we assume diagonal covariance)

depends on $\boldsymbol{a}^{t-1}$ via

$$\sigma^2(\boldsymbol{z}^{t-1}, \boldsymbol{a}^{t-1}) := \frac{1}{10 d_a} \begin{bmatrix} \exp\left(\boldsymbol{G}_1^{\boldsymbol{a}} \cdot \cos(\frac{3}{\pi}\boldsymbol{a}^{t-1})\right) \\ \exp(\boldsymbol{G}_2^{\boldsymbol{a}} \cdot \cos(\frac{4}{\pi}\boldsymbol{a}^{t-1} + 1)) \\ \vdots \\ \exp(\boldsymbol{G}_{d_z}^{\boldsymbol{a}} \cdot \cos(\frac{d_z+2}{\pi}\boldsymbol{a}^{t-1} + d_z - 1)) \end{bmatrix}. \tag{5.196}$$

TimeDiag (Figure 5.5). In this dataset, each $\boldsymbol{z}_i^t$ has only $\boldsymbol{z}_i^{t-1}$ as parent. This trivially satisfies the graphical criterion of Section 5.3.6. The mean function is given by

$$\mu(\boldsymbol{z}^{t-1}, \boldsymbol{a}^{t-1}) := \boldsymbol{z}^{t-1} + 0.5 \sin(\boldsymbol{z}^{t-1}),$$

where the sin function is applied element-wise. Notice that no auxiliary variables are required. One can check that the sufficient variability assumption (Assumption 5.10) and sufficient influence assumption (Assumption 5.10) of Theorem 5.5 (exponential family) holds.

TimeNonDiag (Figure 5.5). We consider a case where the graphical criterion of Section 5.3.6 is satisfied non-trivially. Let

$$\boldsymbol{G}^z := \begin{pmatrix} 1 & & & & \\ 1 & 1 & & & \\ \vdots & & \ddots & & \\ 1 & & & 1 & \\ 1 & 1 & \dots & 1 & 1 \end{pmatrix} \tag{5.197}$$

be the adjacency matrix between $\boldsymbol{z}^t$ and $\boldsymbol{z}^{t-1}$. The ith row of $\boldsymbol{G}^z$, denoted by $\boldsymbol{G}_i^z$, corresponds to the parents of $\boldsymbol{z}_i^t$. Notice that this connectivity matrix has no 2-cycles and all self-loops are present. Thus, by Proposition 5.8, it satisfies the graphical criterion of Section 5.3.6. The mean function in this case is given by

$$\mu(\boldsymbol{z}^{t-1}, \boldsymbol{a}^{t-1}) := \boldsymbol{z}^{t-1} + 0.5 \begin{bmatrix} \boldsymbol{G}_1^{\boldsymbol{z}} \cdot \sin(\frac{3}{\pi}\boldsymbol{z}^{t-1}) \\ \boldsymbol{G}_2^{\boldsymbol{z}} \cdot \sin(\frac{4}{\pi}\boldsymbol{z}^{t-1} + 1) \\ \vdots \\ \boldsymbol{G}_{d_z}^{\boldsymbol{z}} \cdot \sin(\frac{d_z+2}{\pi}\boldsymbol{z}^{t-1} + d_z - 1) \end{bmatrix}, \tag{5.198}$$

which is analogous to (5.194). One can verify that this transition model satisfies the sufficient variability assumption (Assumption 5.10) and sufficient influence assumption (Assumption 5.10) of Theorem 5.5 (exponential family) holds.

TimeNonDiag$_{\text{NoSuffInf}}$ (Table 5.3). This dataset has the same ground truth adjacency matrix as in (5.197), but a different transition function that does not satisfy the assumption of sufficient influence. We sampled a transition matrix W with independent Normal 0-1 entries. The transition

function is thus

$$\mu(\boldsymbol{z}^{t-1}, \boldsymbol{a}^{t-1}) := \boldsymbol{z}^{t-1} + 0.5(\boldsymbol{G}^z \odot W)\boldsymbol{z}^{t-1}\,. \tag{5.199}$$

TimeNonDiag$_{k=2}$ (Table 5.3). This dataset has the lower triangular adjacency matrix of (5.197) and the same mean function of (5.198), but the variance of $\boldsymbol{z}^t$ (we assume diagonal covariance) depends on $\boldsymbol{z}^{t-1}$ via

$$\sigma^2(\boldsymbol{z}^{t-1}, \boldsymbol{a}^{t-1}) := \frac{1}{10d_z}\begin{bmatrix} \exp\left(\boldsymbol{G}^{\boldsymbol{z}}_1 \cdot \cos(\frac{3}{\pi}\boldsymbol{z}^{t-1})\right) \\ \exp(\boldsymbol{G}^{\boldsymbol{z}}_2 \cdot \cos(\frac{4}{\pi}\boldsymbol{z}^{t-1} + 1)) \\ \vdots \\ \exp(\boldsymbol{G}^{\boldsymbol{z}}_{d_z} \cdot \cos(\frac{d_z+2}{\pi}\boldsymbol{z}^{t-1} + d_z - 1)) \end{bmatrix}. \tag{5.200}$$

C.1.2. Datasets that do not satisfy the graphical criterion. The transition mechanisms for the action and temporal datasets of Figure 5.7 and Table 5.4 are (5.194) and (5.198), respectively, except for the graphs which are different.

ActionBlockDiag and ActionBlockNonDiag (Figure 5.7). The left graph corresponds to ActionBlockDiag while the right one corresponds to ActionBlockNonDiag.

$$\boldsymbol{G}^a_{(1)} := \begin{bmatrix} 1 & & & & & \\ 1 & & & & & \\ & 1 & & & & \\ & 1 & & & & \\ & & 1 & & & \\ & & 1 & & & \\ & & & 1 & & \\ & & & 1 & & \\ & & & & 1 & \\ & & & & 1 & \end{bmatrix} \quad \boldsymbol{G}^a_{(2)} := \begin{bmatrix} 1 & & & & & \\ 1 & & & & & \\ & 1 & & & & \\ & 1 & & & & \\ & & 1 & & & \\ & & 1 & & & \\ & & & 1 & & \\ & & & 1 & & \\ 1 & & & & 1 & \\ 1 & & & & 1 & \end{bmatrix}$$

TimeBlockDiag and TimeBlockNonDiag (Figure 5.7). The left graph corresponds to TimeBlockDiag while the right one corresponds to TimeBlockNonDiag.

$$\boldsymbol{G}^z_{(1)} := \begin{bmatrix} 1 & 1 & & & & & & & & \\ 1 & 1 & & & & & & & & \\ & & 1 & 1 & & & & & & \\ & & 1 & 1 & & & & & & \\ & & & & 1 & 1 & & & & \\ & & & & 1 & 1 & & & & \\ & & & & & & 1 & 1 & & \\ & & & & & & 1 & 1 & & \\ & & & & & & & & 1 & 1 \\ & & & & & & & & 1 & 1 \end{bmatrix} \quad \boldsymbol{G}^z_{(2)} := \begin{bmatrix} 1 & 1 & & & & & & & 1 & 1 \\ 1 & 1 & & & & & & & 1 & 1 \\ & & 1 & 1 & & & & & & \\ & & 1 & 1 & & & & & & \\ & & & & 1 & 1 & & & & \\ & & & & 1 & 1 & & & & \\ & & & & & & 1 & 1 & & \\ & & & & & & 1 & 1 & & \\ 1 & 1 & & & 1 & 1 & & & 1 & 1 \\ 1 & 1 & & & 1 & 1 & & & 1 & 1 \end{bmatrix}$$

ActionRandomGraphs and TimeRandomGraphs (Table 5.4). The transition mechanisms are the same as in the ActionNonDiag and TimeNonDiag datasets, i.e. they are given by (5.194) & (5.198), respectively. However, the graphs are sampled randomly, with various levels of sparsity. For the ActionRandomGraphs dataset, we have $\boldsymbol{G}^a_{i,j} \sim \text{Ber}(p)$ and independent. For the TimeRandomGraphs datasets, it is the same except for the diagonal elements, which are forced to be active, i.e. $\boldsymbol{G}^z_{i,i} = 1$.

C.2. Implementation details of our regularized VAE approach

All details of our implementation matches those of Lachapelle et al. [2022] (except for the constrained optimization which was introduced in Section 5.5).

Learned mechanisms. Every coordinate $\boldsymbol{z}_i$ of the latent vector has its own mechanism $\hat{p}(\boldsymbol{z}^t_i \mid \boldsymbol{z}^{<t}, \boldsymbol{a}^{<t})$ that is Gaussian with mean outputted by $\hat{\boldsymbol{\mu}}_i(\boldsymbol{z}^{t-1}, \boldsymbol{a}^{t-1})$ (a multilayer perceptron with 5 layers of 512 units) and a learned variance which does not depend on the previous time steps. For learning, we use the typical parameterization of the Gaussian distribution with μ and σ^2 and not its exponential family parameterization. Throughout, the dimensionality of $\boldsymbol{z}^t$ in the learned model always match the dimensionality of the ground-truth (same for baselines). Learning the dimensionality of $\boldsymbol{z}^t$ is left for future work.

Prior of $\boldsymbol{z}^1$ in time-sparsity experiments. In *time-sparsity* experiments, the prior of the first latent $\hat{p}(\boldsymbol{z}^1)$ (when $t = 1$) is modelled separately as a Gaussian with learned mean and learned diagonal covariance. Note that this learned covariance at time $t = 1$ is different from the subsequent learned conditional covariance at time $t > 1$.

Encoder/Decoder. In all experiments, including baselines, both the encoder and the decoder is modelled by a neural network with 6 fully connected hidden layers of 512 units with LeakyReLU activation with negative slope 0.2. For all VAE-based methods, the encoder outputs the mean and

a diagonal covariance. Moreover, $p(\boldsymbol{x}|\boldsymbol{z})$ has a *learned* isotropic covariance $\sigma^2 I$. Note that $\sigma^2 I$ corresponds to the covariance of the independent noise $\boldsymbol{n}^t$ in the equation $\boldsymbol{x}^t = \boldsymbol{f}(\boldsymbol{z}^t) + \boldsymbol{n}^t$.

C.3. Baselines

In synthetic experiments of Sec. 5.8, all methods used a minibatch size of 1024 and the same encoder and decoder architecture: A MLP with 6 layers of 512 units with LeakyReLU activations (negative slope of 0.2). We tuned manually the learning rate of each method to ensure proper convergence. For VAE-based methods, i.e. TCVAE, SlowVAE and iVAE, we are always choosing $p(x|z)$ Gaussian with a covariance $\sigma^2 I$ and learn σ^2.

β-TCVAE. We used the implementation provided in the original paper by Chen et al. [2018] which is available at `https://github.com/rtqichen/beta-tcvae`. We used a learning rate of 1e-4.

iVAE. We used the implementation available at `https://github.com/ilkhem/icebeem` from Khemakhem et al. [2020a]. In it, the mean of the prior $p(z|a)$ is fixed to zero while its diagonal covariance is allowed to depend on a through an MLP. We change this to allow the mean to also depend on a through the neural network (with 5 layers and width 512). We also lower bounded its variance as well as the variance of $q(z \mid x, a)$ to improve the stability of learning. In the original implementation, the covariance of $p(x|z)$ was not learned. We found that learning it (analogously to what we do in our method) improved performance. We used a learning rate of 1e-4.

SlowVAE. We used the implementation provided in `https://github.com/bethgelab/slow_disentanglement` [Klindt et al., 2021]. Like for other VAE-based methods, we modelled $p(x|z)$ as a Gaussian with covariance $\sigma^2 I$ and learned σ^2.

PCL. We used the implementation provided here: `https://github.com/bethgelab/slow_disentanglement/tree/baselines`. PCL [Hyvarinen and Morioka, 2017] stands for "permutation contrastive learning" and works as follows: Given sequential data $\{\boldsymbol{x}^t\}_{t=1}^T$, PCL trains a regression function $r((x', x))$ to discriminate between pairs of adjacent observations (positive pairs) and randomly matched pairs (negative pairs). The regression function has the form

$$r((x, x')) = \sum_{i=1}^{d_z} B_i(h_i(x), h_i(x')) \,, \tag{5.201}$$

where $h : \mathbb{R}^{d_x} \to \mathbb{R}^{d_z}$ is the encoder and $B_i : \mathbb{R}^2 \to \mathbb{R}$ are learned functions. In our implementation, the B_i functions are fully connected neural networks with 5 layers and 512 hidden units. We experimented with the less expressive function suggested in the original work, but found that the extra capacity improved performance across all datasets we considered.

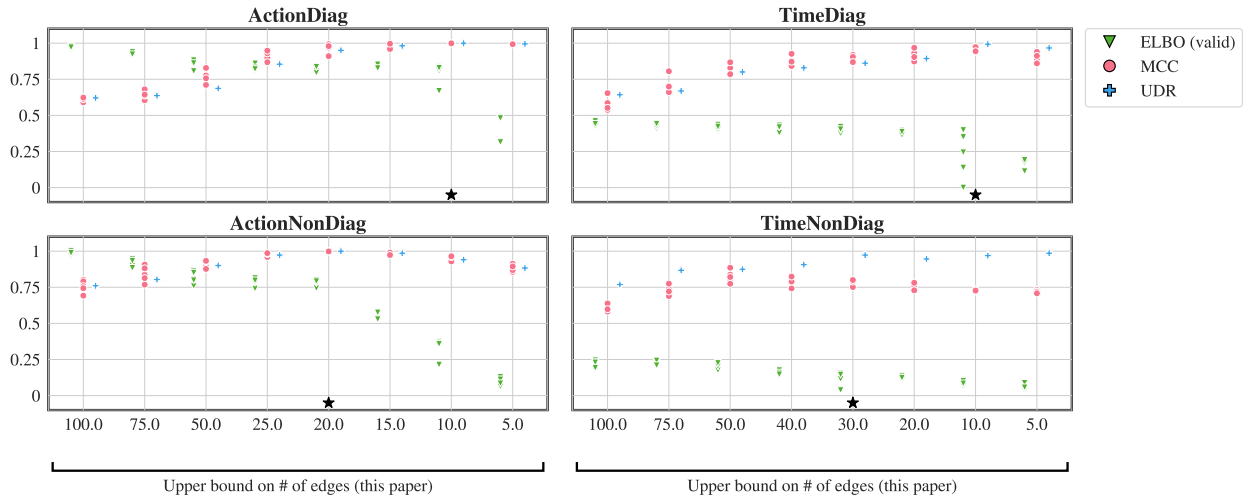


Figure 5.9. Investigating the link between goodness of fit (ELBO), disentanglement (MCC) and UDR. The ELBO is normalized so that it remains between 0 and 1.

C.4. Unsupervised hyperparameter selection

In practice, one cannot measure MCC since the ground-truth latent variables are not observed. Unlike in standard machine learning setting, hyperparameter selection for disentanglement cannot be performed simply by evaluating goodness of fit on a validation set and selecting the highest scoring model since there is usually a trade-off between goodness of fit and disentanglement [Locatello et al., 2019, Sec. 5.4]. To circumvent this problem, Duan et al. [2020] introduced *unsupervised disentanglement ranking* (UDR) which, for every hyperparameter combinations, measures how consistent are different random intializations of the algorithm. The authors argue that hyperparameters yielding disentangled representation typically yields consistent representations. In our experiments, the consistency of a given hyperparameter combination is measured as follows: for every pair of models, we compute the MCC between their representations. Then, we report the median of all pairwise MCC. This gives a UDR score for every hyperparameter values considered. Figure 5.9 reports the ELBO (normalized between zero and one), the MCC and the UDR score for the experiments of Figure 5.5. We can visualize the trade-off between ELBO and MCC. That being said, MCC and UDR correlates nicely except for the TimeNonDiag dataset, in which this correlation breaks for stronger regularization. We noticed that these specific runs correspond to excessively sparse graph, with fewer than 10 edges (out of 100 possible edges). The black star indicates the hyperparameter selected by UDR when excluding coefficient values which yields graphs with less than 10 edges (on average).

Baselines. Two of the baselines considered had hyperparameters to tune, SlowVAE [Klindt et al., 2021] and TCVAE [Chen et al., 2018]. For SlowVAE, we did a grid search on the following values, $\gamma \in \{1.0, 2.0, 4.0, 8.0, 16.0\}$ and $\alpha \in \{1, 3, 6, 10\}$. For TCVAE, we explored $\beta \in \{1, 2, 3, 4, 5\}$ but

the optimal value in terms of disentanglement was almost always 1. Values of β larger than 5 led to instabilities during training. The hyperparameters were selected using UDR, as described in the paragraph above.

D. Miscellaneous

D.1. On the invertibility of the mixing function

Throughout this work as well as many others [Hyvarinen and Morioka, 2016, 2017, Hyvärinen et al., 2019, Khemakhem et al., 2020a, Locatello et al., 2020a, Klindt et al., 2021], it is assumed that the mixing function mapping the latent factors to the observation is a diffeomorphism onto its image. In this section, we briefly discuss the practical implications of this assumption.

Recall that a diffeomorphism is a differentiable bijective function with a differentiable inverse. We start by adressing the bijective part of the assumption. To understand it, we consider a plausible situation where the mapping $\boldsymbol{f}$ is not invertible. Consider the minimal example of Fig. 5.1 consisting of a tree, a robot and a ball. Assume that the ball can be hidden behind either the tree or the robot. Then, the mixing function $\boldsymbol{f}$ is not invertible because, given only the image, it is impossible to know whether the ball is behind the tree or the robot. Thus, this situation is not covered by our theory. Intuitvely, one could infer, at least approximately, where the ball is hidden based on previous time frames. Allowing for this form of occlusion is left as future work. See also Mansouri et al. [2022] for further discussion about how one can relax this assumption.

We believe the differentiable part of this assumption is only a technicality that could probably be relaxed to being piecewise differentiable. Our experiments were performed with data generated with a piecewise linear $\boldsymbol{f}$, which in not differentiable only on a set of (Lebesgue) measure zero, but this was not an issue in practice.

D.2. Contrasting with the assumptions of Khemakhem et al. [2020a] & Yao et al. [2022b]

In this section, we discuss two identifiability results previously proposed in the literature that do not leverage sparsity [Khemakhem et al., 2020a, Yao et al., 2022a]. We show that these results do not apply to the simple homoscedastic Gaussian latent models of the form $p(\boldsymbol{z}_i^t \mid \boldsymbol{z}^{t-1}) = \mathcal{N}(\boldsymbol{z}_i^t \mid \boldsymbol{\mu}_i(\boldsymbol{z}^{t-1}), \sigma_i^2)$, contrarily to our theory, as we saw in Examples 5.8, 5.9 and 5.11. We will see that in the context of a Gaussian latent model, both results require the variance to vary sufficiently strongly. We believe that such a requirement is not well suited for nearly deterministic environments such as the one depicted in Figure 5.1.

Khemakhem et al. [2020a]. The most significant distinction between the theory of Khemakhem et al. [2020a] (iVAE) and ours is how identifiability up to permutation is obtained: Theorems 2 & 3

from iVAE shows that if the assumptions of their Theorem 1 (which is essentially Theorem 5.4) are satisfied and $\boldsymbol{s}_i$ has dimension $k > 1$ **or** is non-monotonic, then the model is not just identifiable up to linear transformation but up to permutations (and rescalings). In contrast, our theory covers the case where $k = 1$ and $\boldsymbol{s}_i$ is monotonic, like in the homoscedastic Gaussian case. Interestingly, Khemakhem et al. [2020a] mentioned this specific case as a counterexample to their theory in their Proposition 3. The extra power of our theory comes from the extra *structure* in the dependencies of the latent factors coupled with sparsity regularization. We note that, assuming the latent factors are Gaussian, the variability assumption of Theorem 5.4 combined with $k > 1$ requires the variance to vary sufficiently, which is implausible in the nearly deterministic environment of Figure 5.1.
Yao et al. [2022a]. This work (Theorem 1) requires that, for each value of $\boldsymbol{z}^t$, the $2d_z$ functions

$$\frac{\partial^2}{\partial \boldsymbol{z}_i^t \partial \boldsymbol{z}^{t-1}} \log p(\boldsymbol{z}_i^t \mid \boldsymbol{z}^{t-1}) \text{ and } \frac{\partial^3}{(\partial \boldsymbol{z}_i^t)^2 \partial \boldsymbol{z}^{t-1}} \log p(\boldsymbol{z}_i^t \mid \boldsymbol{z}^{t-1}) \text{ for } i = 1 \dots d_z \,,$$

seen as functions from $\mathbb{R}^{d_z}$ to $\mathbb{R}^{d_z}$ are linearly independent. Indeed, if $p(\boldsymbol{z}_i^t \mid \boldsymbol{z}^{t-1}) = \mathcal{N}(\boldsymbol{z}_i^t \mid \boldsymbol{\mu}_i(\boldsymbol{z}^{t-1}), \sigma_i^2)$, one can easily derive that

$$\frac{\partial}{\partial \boldsymbol{z}_i^t} \log p(\boldsymbol{z}_i^t \mid \boldsymbol{z}_i^{t-1}) = -(\boldsymbol{z}_i - \boldsymbol{\mu}_i(\boldsymbol{z}^{t-1}))/\sigma_i^2 \tag{5.202}$$

$$\frac{\partial^2}{(\partial \boldsymbol{z}_i^t)^2} \log p(\boldsymbol{z}_i^t \mid \boldsymbol{z}_i^{t-1}) = -1/\sigma_i^2 \tag{5.203}$$

$$\frac{\partial^2}{(\partial \boldsymbol{z}_i^t)^2 \partial \boldsymbol{z}^{t-1}} \log p(\boldsymbol{z}_i^t \mid \boldsymbol{z}_i^{t-1}) = \mathbf{0} \,, \tag{5.204}$$

which shows that the assumption of Yao et al. [2022a, Theorem 1] does not hold for homoscedastic Gaussian latent models. We further notice that, had the variance σ_i^2 depend on $\boldsymbol{z}^{t-1}$, the identifiability result of Yao et al. [2022b] could have applied.

D.3. Derivation of the ELBO

In this section, we derive the evidence lower bound presented in Sec. 5.5.

$$\log p(\boldsymbol{x}^{\leq T} \mid \boldsymbol{a}^{<T}) = \tag{5.205}$$

$$\mathbb{E}_{q(\boldsymbol{z}^{\leq T} \mid \boldsymbol{x}^{\leq T}, \boldsymbol{a}^{<T})} \left[\log \frac{q(\boldsymbol{z}^{\leq T} \mid \boldsymbol{x}^{\leq T}, \boldsymbol{a}^{<T})}{p(\boldsymbol{z}^{\leq T} \mid \boldsymbol{x}^{\leq T}, \boldsymbol{a}^{<T})} \right. \tag{5.206}$$

$$\left. + \log \frac{p(\boldsymbol{z}^{\leq T}, \boldsymbol{x}^{\leq T} \mid \boldsymbol{a}^{<T})}{q(\boldsymbol{z}^{\leq T} \mid \boldsymbol{x}^{\leq T}, \boldsymbol{a}^{<T})} \right] \tag{5.207}$$

$$\geq \mathbb{E}_{q(\boldsymbol{z}^{\leq T} \mid \boldsymbol{x}^{\leq T}, \boldsymbol{a}^{<T})} \left[\log \frac{p(\boldsymbol{z}^{\leq T}, \boldsymbol{x}^{\leq T} \mid \boldsymbol{a}^{<T})}{q(\boldsymbol{z}^{\leq T} \mid \boldsymbol{x}^{\leq T}, \boldsymbol{a}^{<T})} \right] \tag{5.208}$$

$$= \mathbb{E}_{q(\boldsymbol{z}^{\leq T} \mid \boldsymbol{x}^{\leq T}, \boldsymbol{a}^{<T})} \left[\log p(\boldsymbol{x}^{\leq T} \mid \boldsymbol{z}^{\leq T}, \boldsymbol{a}^{<T}) \right] \tag{5.209}$$

$$- KL(q(\boldsymbol{z}^{\leq T} \mid \boldsymbol{x}^{\leq T}, \boldsymbol{a}^{<T}) || p(\boldsymbol{z}^{\leq T} \mid \boldsymbol{a}^{<T})) \tag{5.210}$$

where the inequality holds because the term at (5.206) is a Kullback-Leibler divergence, which is greater or equal to 0. Notice that

$$p(\boldsymbol{x}^{\leq T} \mid \boldsymbol{z}^{\leq T}, \boldsymbol{a}^{<T}) = p(\boldsymbol{x}^{\leq T} \mid \boldsymbol{z}^{\leq T}) = \prod_{t=1}^{T} p(\boldsymbol{x}^t \mid \boldsymbol{z}^t)\,. \tag{5.211}$$

Recall that we are considering a variational posterior of the following form:

$$q(\boldsymbol{z}^{\leq T} \mid \boldsymbol{x}^{\leq T}, \boldsymbol{a}^{<T}) := \prod_{t=1}^{T} q(\boldsymbol{z}^t \mid \boldsymbol{x}^t)\,. \tag{5.212}$$

Equations (5.211) & (5.212) allow us to rewrite the term in (5.209) as

$$\sum_{t=1}^{T} \mathop{\mathbb{E}}_{\boldsymbol{z}^t \sim q(\cdot|\boldsymbol{x}^t)} [\log p(\boldsymbol{x}^t \mid \boldsymbol{z}^t)] \tag{5.213}$$

Notice further that

$$p(\boldsymbol{z}^{\leq T} \mid \boldsymbol{a}^{<T}) = \prod_{t=1}^{T} p(\boldsymbol{z}^t \mid \boldsymbol{z}^{<t}, \boldsymbol{a}^{<t})\,. \tag{5.214}$$

Using (5.212) & (5.214), the KL term (5.210) can be broken down as a sum of KL as:

$$\sum_{t=1}^{T} \mathop{\mathbb{E}}_{\boldsymbol{z}^{<t} \sim q(\cdot|\boldsymbol{x}^{<t})} KL(q(\boldsymbol{z}^t \mid \boldsymbol{x}^t) || p(\boldsymbol{z}^t \mid \boldsymbol{z}^{<t}, \boldsymbol{a}^{<t})) \tag{5.215}$$

Putting all together yields the desired ELBO:

$$\begin{aligned} \log p(\boldsymbol{x}^{\leq T} | \boldsymbol{a}^{<T}) \geq & \sum_{t=1}^{T} \mathop{\mathbb{E}}_{\boldsymbol{z}^t \sim q(\cdot|\boldsymbol{x}^t)} [\log p(\boldsymbol{x}^t \mid \boldsymbol{z}^t)] \\ & - \mathop{\mathbb{E}}_{\boldsymbol{z}^{<t} \sim q(\cdot|\boldsymbol{x}^{<t})} KL(q(\boldsymbol{z}^t \mid \boldsymbol{x}^t) || p(\boldsymbol{z}^t \mid \boldsymbol{z}^{<t}, \boldsymbol{a}^{<t}))\,. \end{aligned} \tag{5.216}$$

Prologue to the Fourth Contribution

Article Details

Synergies between Disentanglement and Sparsity: Generalization and Identifiability in Multi-Task Learning

by *Sébastien Lachapelle*, Tristan Deleu*, Divyat Mahajan, Ioannis Mitliagkas, Yoshua Bengio, Simon Lacoste-Julien and Quentin Bertrand.* This work was published at the 40th International Conference on Machine Learning (ICML 2023).

*Equal contributions.

Contributions of the Authors

Sébastien Lachapelle developed the ideas and proofs for how sparse multi-task learning can yield disentanglement as well as how disentangled representations combined with sparsity regularization can improve generalization. He also contributed to the writing and led the disentanglement experiments. **Tristan Deleu** developed the code based in JAX guided by his experience in meta-learning, contributed to the implementation of the non-smooth bilevel optimization, led the experiments on miniImageNet and contributed to the writing. **Divyat Mahajan** led the generalization experiments. **Simon Lacoste-Julien** provided supervision, contributed to the writing and provided guidance for the theory. **Quentin Bertrand** supervised the project, brought his expertise on non-smooth bilevel optimization to the project, led the implementation of the non-smooth bilevel optimization, derived and implemented the dual of the group Lasso penalized multiclass SVM used in the miniImageNet and generally helped with the writing and experiments.

Context and Limitations

Bengio et al. [2013, Section 3.5] explain the difference between the goal of learning an *invariant representation*, which is about *what* information is captured by the representation, and that of learning a *disentangled representation*, which is about *how* the information is represented. The

authors further argue that disentanglement is crucial since the learner does not know which features will be important for a given task ahead of time:

> *It is important to distinguish between the related but distinct goals of learning invariant features and learning to disentangle explanatory factors. The central difference is the preservation of information. Invariant features, by definition, have reduced sensitivity in the direction of invariance. This is the goal of building features that are insensitive to variation in the data that are uninformative to the task at hand. Unfortunately, it is often difficult to determine a priori which set of features and variations will ultimately be relevant to the task at hand. Further, as is often the case in the context of deep learning methods, the feature set being trained may be destined to be used in multiple tasks that may have distinct subsets of relevant features. Considerations such as these lead us to the conclusion that the most robust approach to feature learning is to disentangle as many factors as possible, discarding as little information about the data as is practical.* — Bengio et al. [2013, Section 3.5]

However, the precise mechanism by which a disentangled representation leads to more robustness is left rather open. It is implied that disentanglement should allow the learner to choose the right features for the task it is confronted to in a way that would not be possible had the representation been entangled. But how exactly? And is disentanglement necessary to achieve this? Answering these questions is important, especially given the fact that empirical works investigating whether disentanglement improves downstream performance reach conflicting conclusions (see Section 6.4).

The following contribution proposes the first theoretical principle explaining how disentangled representations have an advantage over entangled ones in a few-shot setting when the features that "will ultimately be relevant to the task at hand" are unknown in advance. The key assumption is that, when using a disentangled representation, all possible future tasks can be solved using a sparse predictor[9], i.e. one which depends only on a small subset of features. Under this assumption, it is clear that fitting a predictor with sparse regularization on top of a representation will improve sample complexity (sparsity regularization reduces the hypothesis class) without introducing any bias *given that the representation used is disentangled*. Indeed, if the representation is entangled, the optimal predictor for the task and that representation will not be sparse, which means sparsity regularization will induce bias. We emphasize that this conclusion holds even when the entangled representation contains exactly the same information as the disentangled one, which makes clear that the magic happens because of *how* the information is encoded and not *what* information is encoded. In some sense, we want to "align" the representation learned (disentanglement) with the inductive bias of the predictor (sparsity) to learn with fewer samples in an unbiased way when confronted to a diverse set of tasks. Although this argument is formalized in Section 6.2, note that

[9]The analysis is carried out with linear predictors, but could be extended to general nonlinear predictors, as long as they only use a sparse subset of features to predict.

we do not quantify the gains in sample complexity (with generalization bounds) arising from having a disentangled representation combined with sparsity regularization and left this for future work.

We further propose a novel identifiability result based on sparse multi-task learning, which adds to the now growing literature on identifiable representation learning (see Sections 5.7 & 6.4 for reviews). The proposed result has some nice properties such as allowing for dependent factors $\boldsymbol{z}_i$, for a non-invertible relationship between the observations $\boldsymbol{x}$ and factors $\boldsymbol{z}$ as well as making next to no assumptions on $p(\boldsymbol{x})$ (see next paragraph). However, the methodology also suffers from some limitations. For instance, the result assumes we observe an uncountably infinite number of tasks in order to prove disentanglement, which adds on top of the usual "infinite data" assumptions found in identifiability analyses. Although this assumption appears difficult to get rid of, there might be ways to avoid it. For instance, there is a body of literature showing identifiability of sparse dictionary learning up to permutation and rescaling in the finite-sample regime [Georgiev et al., 2005, Aharon et al., 2006, Hu and Huang, 2023]. These results could form a basis to move to finite-sample identifiability analyses in the nonlinear case.

Discriminative v.s. generative assumptions. Interestingly, the identifiability result of this contribution makes assumptions only about $p(y \mid \boldsymbol{x}, \boldsymbol{w})$, where $\boldsymbol{w}$ is a task-specific parameter, and none about $p(\boldsymbol{x})$. This is in stark contrast with the results of Chapters 5 & 7 which imposes restrictions on $p(\boldsymbol{x})$. Avoiding making assumptions about $p(\boldsymbol{x})$ is a good thing, since coming up with a good model for the distribution of images $p(\boldsymbol{x})$ is difficult.

Recent developments

Briefly after the publication of our work, Fumero et al. [2023] introduced a very similar approach with the key difference being that they add an additional regularization to encourage sharing features across tasks, to prevent duplication of features. This additional regularizer appears to be especially important when the number of latent factors is unknown, as is the case in practice. The authors also show empirically that, on more realistic domain generalization tasks such as PACS [Li et al., 2017], VLCS [Fang et al., 2013], OfficeHome [Venkateswara et al., 2017] and Waterbirds [Sagawa et al., 2020], this sparse multitask learning approach can learn features that allow for better few-shot performance than features learn via ERM. Interestingly, to achieve these results, they apply a sparsity penalty when fitting the linear head, which corroborates our own theoretical analysis on how disentangled representations combined with sparsity regularization can improve sample complexity.

The proof strategy we developed has been reused by Bing et al. [2023] to show causal representation learning is possible with multi-target interventions with linear mixing and by Xu et al. [2023] to show identifiability with sparse latent factors.

Chapter 6

Synergies Between Disentanglement and Sparsity: Generalization and Identifiability in Multi-Task Learning

Abstract

Although disentangled representations are often said to be beneficial for downstream tasks, current empirical and theoretical understanding is limited. In this work, we provide evidence that disentangled representations coupled with sparse task-specific predictors improve generalization. In the context of multi-task learning, we prove a new identifiability result that provides conditions under which maximally sparse predictors yield disentangled representations. Motivated by this theoretical result, we propose a practical approach to learn disentangled representations based on a sparsity-promoting bi-level optimization problem. Finally, we explore a meta-learning version of this algorithm based on group Lasso multiclass SVM predictors, for which we derive a tractable dual formulation. It obtains competitive results on standard few-shot classification benchmarks, while each task is using only a fraction of the learned representations.

6.1. Introduction

The recent literature on self-supervised learning has provided evidence that learning a representation on large corpuses of data can yield strong performances on a wide variety of downstream tasks [Devlin et al., 2018, Chen et al., 2020], especially in few-shot learning scenarios where the training data for these tasks is limited [Brown et al., 2020, Dosovitskiy et al., 2021b, Radford et al., 2021]. Beyond transferring across multiple tasks, these learned representations also lead to improved robustness against distribution shifts [Wortsman et al., 2022] as well as stunning text-conditioned image generation [Ramesh et al., 2022]. However, preliminary assessments of the

latter have highlighted shortcomings related to compositionality [Marcus et al., 2022], suggesting new algorithmic innovations are needed.

Another line of work has argued for the integration of ideas from causality to make progress towards more robust and transferable machine learning systems [Pearl, 2019, Schölkopf, 2019, Goyal and Bengio, 2021]. *Causal representation learning* has emerged recently as a field aiming to define and learn representations suited for causal reasoning [Schölkopf et al., 2021]. This set of ideas is strongly related to learning *disentangled representations* [Bengio et al., 2013]. Informally, a representation is considered disentangled when its components are in one-to-one correspondence with natural and interpretable factors of variations, such as object positions, colors or shapes. Although a plethora of works have investigated theoretically under which conditions disentanglement is possible through the lens of identifiability [Hyvarinen and Morioka, 2016, 2017, Hyvärinen et al., 2019, Khemakhem et al., 2020a, Locatello et al., 2020a, Klindt et al., 2021, Von Kügelgen et al., 2021, Gresele et al., 2021, Lachapelle et al., 2022, Lippe et al., 2022, Ahuja et al., 2022c], fewer works have tackled *how a disentangled representation could be beneficial for downstream tasks*. Those who did mainly provide empirical rather than theoretical evidence for or against its usefulness [Locatello et al., 2019, van Steenkiste et al., 2019, Miladinović et al., 2019, Dittadi et al., 2021, Montero et al., 2021]. We believe our work can bring some theoretical insights as to when and why disentanglement can help.

In this work, we explore synergies between disentanglement and sparse task-specific predictors in the context of multi-task learning. At the heart of our contributions is the assumption that only a small subset of all factors of variations are useful for each downstream task, and this subset might change from one task to another. We will refer to such tasks as *sparse tasks*, and their corresponding sets of useful factors as their *supports*. This assumption was initially suggested by Bengio et al. [2013, Section 3.5]: "the feature set being trained may be destined to be used in multiple tasks that may have distinct [and unknown] subsets of relevant features. Considerations such as these lead us to the conclusion that the most robust approach to feature learning is to disentangle as many factors as possible, discarding as little information about the data as is practical". This strategy is in line with the current self-supervised learning trend [Radford et al., 2021], except for its focus on disentanglement.

6.1.1. Contributions

(1) We formalize this "sparse task assumption" and argue theoretically and empirically how, when it holds, a disentangled representation coupled with a sparsity-regularized task-specific predictor can generalize better than their entangled counterparts (Section 6.2).

(2) We introduce a novel identifiability result (Theorem 6.1) which shows how one can leverage multiple sparse supervised tasks to learn a shared disentangled representation by regularizing

the task-specific predictors to be maximally sparse (Section 6.3.2). We note that the usage of supervision is in line with many recent results which leverages more or less weak forms of supervision to guarantee identifiability. Contrary to many existing identifiability results, ours allows for statistically dependent latent factors and a non-invertible map between observations and latents.

(3) Motivated by this result, we propose a tractable bi-level optimization (Problem (6.6)) to learn the shared representation while regularizing the task-specific predictors to be sparse (Section 6.3.4). We validate our theory by showing our approach can indeed disentangle latent factors on tasks constructed from the 3D Shapes dataset [Burgess and Kim, 2018].

(4) Finally, we draw a connection between this bi-level optimization problem and formulations from the meta-learning literature. Inspired by our identifiability result, we enhance an existing method [Lee et al., 2019], where the task-specific predictors are now group-sparse SVMs. We show that this new meta-learning algorithm achieves competitive performance on the *mini*ImageNet benchmark [Vinyals et al., 2016], while only using a fraction of the representation.

We emphasize that, although related, the theoretical contributions of Sections 6.2 & 6.3 are distinct and stand of their own. Indeed, Section 6.2 shows how disentangled representations combined with sparsity regularization can improve generalization, while Section 6.3 shows how regularizing task-specific predictors to be sparse can induce disentanglement in a multi-task learning setting.

6.1.2. Background

We start by introducing formally the notion of entangled and disentangled representations.

First, we assume the existence of some ground-truth encoder function $\boldsymbol{f_\theta} : \mathbb{R}^d \to \mathbb{R}^m$ that maps observations $\boldsymbol{x} \in \mathcal{X} \subseteq \mathbb{R}^d$, *e.g.*, images, to its corresponding interpretable and usually lower dimensional representation $\boldsymbol{f_\theta}(\boldsymbol{x}) \in \mathbb{R}^m$, $m \leq d$. The exact form of this ground-truth encoder depends on the task at hand, but also on what the machine learning practitioner considers as interpretable. The learned encoder function is denoted by $\boldsymbol{f_{\hat{\theta}}} : \mathbb{R}^d \to \mathbb{R}^m$, and should not be conflated with the ground-truth representation $\boldsymbol{f_\theta}$. For example, $\boldsymbol{f_{\hat{\theta}}}$ can be parametrized by a neural network. Throughout, we are going to use the following definition of disentanglement.

Definition 6.1 (Disentangled Representation, Khemakhem et al. 2020a, Lachapelle et al. 2022)**.** *A learned encoder function* $\boldsymbol{f_{\hat{\theta}}} : \mathbb{R}^d \to \mathbb{R}^m$ *is said to be* disentangled w.r.t. the ground-truth representation $\boldsymbol{f_\theta}$ *when there exists an invertible diagonal matrix* $\boldsymbol{D}$ *and a permutation matrix* $\boldsymbol{P}$ *such that, for all* $\boldsymbol{x} \in \mathcal{X}$, $\boldsymbol{f_{\hat{\theta}}}(\boldsymbol{x}) = \boldsymbol{D}\boldsymbol{P}\boldsymbol{f_\theta}(\boldsymbol{x})$.

Intuitively, a representation is disentangled when there is a one-to-one correspondence between its components and those of the ground-truth representation, up to rescaling. When an encoder $\boldsymbol{f_{\hat{\theta}}}$ is

not disentangled, we say it is *entangled*. Note that there exist less stringent notions of disentanglement which allow for component-wise nonlinear invertible transformations of the factors [Hyvarinen and Morioka, 2017, Hyvärinen et al., 2019].

Notation. Capital bold letters denote matrices and lowercase bold letters denote vectors. The set of integers from 1 to n is denoted by $[n]$. We write $\|\cdot\|$ for the Euclidean norm on vectors and the Frobenius norm on matrices. For a matrix $\boldsymbol{A} \in \mathbb{R}^{k\times m}$, $\|\boldsymbol{A}\|_{2,1} = \sum_{j=1}^{m} \|\boldsymbol{A}_{:j}\|$, and $\|\boldsymbol{A}\|_{2,0} = \sum_{j=1}^{m} \mathbb{1}_{\|\boldsymbol{A}_{:j}\|\neq 0}$, where $\mathbb{1}$ is the indicator function. The ground-truth parameter of the encoder function is $\boldsymbol{\theta}$, while that of the learned representation is $\hat{\boldsymbol{\theta}}$. We follow this convention for all the parameters throughout. Table 6.1 in Appendix A summarizes all the notation.

6.2. Disentanglement and Sparse Task-Specific Predictors Improve Generalization

In this section, we show that for any *linearly equivalent* representation (entangled or disentangled), the maximum likelihood estimator defined in Problem (6.1) yields the same model (Proposition 6.1). However, we also show that disentangled representations have better generalization properties when the task-specific predictor is regularized to be sparse. (Proposition 6.2 & Figure 6.1). Our analysis is centred around the following assumption.

Assumption 6.1 (Linear equivalence)**.** *The learned encoder $\boldsymbol{f}_{\hat{\theta}}$ is* linearly equivalent *to the ground-truth encoder $\boldsymbol{f}_{\boldsymbol{\theta}}$,* i.e., *there exists an invertible matrix $\boldsymbol{L}$ such that, for all $\boldsymbol{x} \in \mathcal{X}$, $\boldsymbol{f}_{\hat{\theta}}(\boldsymbol{x}) = \boldsymbol{L}\boldsymbol{f}_{\boldsymbol{\theta}}(\boldsymbol{x})$.*

Note that similar notions of linear equivalence were used e.g. by Hyvärinen et al. [2019], Khemakhem et al. [2020a], Roeder et al. [2021]

Despite being assumed linearly equivalent, the learned representation $\boldsymbol{f}_{\hat{\theta}}$ might not be disentangled (Definition 6.1); in that case, we say the representation is *linearly entangled*. When we refer to a disentangled representation, we write $\boldsymbol{L} := \boldsymbol{D}\boldsymbol{P}$. Roeder et al. [2021] have shown that many common methods learn representations identifiable up to linear equivalence, such as deep neural networks for classification, contrastive learning [Oord et al., 2018, Radford et al., 2021] and autoregressive language models [Mikolov et al., 2010, Brown et al., 2020].

6.2.1. MLE invariance to linear feature transformations

Consider the following maximum likelihood estimator (MLE):[1]

$$\hat{\boldsymbol{W}}_n^{(\hat{\boldsymbol{\theta}})} := \arg\max_{\tilde{\boldsymbol{W}}} \sum_{(\boldsymbol{x},y)\in\mathcal{D}} \log p(y; \boldsymbol{\eta} = \tilde{\boldsymbol{W}}\boldsymbol{f}_{\hat{\boldsymbol{\theta}}}(\boldsymbol{x}))\,, \tag{6.1}$$

[1] We assume the solution is unique.

where y denotes the label, $\mathcal{D} := \{(\boldsymbol{x}^{(i)}, y^{(i)})\}_{i=1}^n$ is the dataset, $p(y; \boldsymbol{\eta})$ is a distribution over labels[2] parameterized by $\boldsymbol{\eta} \in \mathbb{R}^k$, and $\hat{\boldsymbol{W}} \in \mathbb{R}^{k \times m}$ is the *task-specific predictor*. The following result shows that the model estimated via maximum likelihood defined in Problem (6.1) is invariant to invertible linear transformations of the features. Note that it is an almost direct consequence of the invariance of MLE to reparametrization [Casella and Berger, 2001, Thm. 7.2.10]. See Appendix A for a proof.

Proposition 6.1. *Let $\hat{\boldsymbol{W}}_n^{(\hat{\boldsymbol{\theta}})}$ and $\hat{\boldsymbol{W}}_n^{(\boldsymbol{\theta})}$ be the solutions to Problem (6.1) with the representations $\boldsymbol{f}_{\hat{\boldsymbol{\theta}}}$ and $\boldsymbol{f}_{\boldsymbol{\theta}}$, respectively (which we assume are unique). If $\boldsymbol{f}_{\hat{\boldsymbol{\theta}}}$ and $\boldsymbol{f}_{\boldsymbol{\theta}}$ are linearly equivalent (Assumption 6.1), then we have, $\forall \boldsymbol{x} \in \mathcal{X}$, $\hat{\boldsymbol{W}}_n^{(\hat{\boldsymbol{\theta}})} \boldsymbol{f}_{\hat{\boldsymbol{\theta}}}(\boldsymbol{x}) = \hat{\boldsymbol{W}}_n^{(\boldsymbol{\theta})} \boldsymbol{f}_{\boldsymbol{\theta}}(\boldsymbol{x})$.*

Proposition 6.1 shows that the model $p(y; \hat{\boldsymbol{W}}_n^{(\hat{\boldsymbol{\theta}})} \boldsymbol{f}_{\hat{\boldsymbol{\theta}}}(\boldsymbol{x}))$ learned by Problem (6.1) is independent of $\boldsymbol{L}$, *i.e., the learned model is the same for disentangled and linearly entangled representations*. We thus expect both disentangled and linearly entangled representations to perform identically on downstream tasks.

6.2.2. An advantage of disentangled representations

We are now going to see how adding sparsity regularization to Problem (6.1) favors the disentangled representation when the ground-truth data generating process is truly sparse.

Assumption 6.2 (Data generation process)**.** *The input-label pairs are i.i.d. samples from the distribution $p(\boldsymbol{x}, y) := p(y; \boldsymbol{W} \boldsymbol{f}_{\boldsymbol{\theta}}(\boldsymbol{x}))p(\boldsymbol{x})$, where $\boldsymbol{W} \in \mathbb{R}^{k \times m}$ is the ground-truth coefficient matrix such that $\|\boldsymbol{W}\|_{2,0} = \ell$.*

To formalize the hypothesis that *only a subset of the features $\boldsymbol{f}_{\boldsymbol{\theta}}(\boldsymbol{x})$ are actually useful to predict the target y*, we assume that the ground-truth coefficient matrix $\boldsymbol{W}$ is column sparse, *i.e.,* $\|\hat{\boldsymbol{W}}\|_{2,0} = \ell < m$. Under this assumption, it is natural to constrain the MLE as such:

$$\hat{\boldsymbol{W}}_n^{(\hat{\boldsymbol{\theta}},\ell)} := \operatorname*{arg\,max}_{\|\tilde{\boldsymbol{W}}\|_{2,0} \leq \ell} \sum_{(\boldsymbol{x},y) \in \mathcal{D}} \log p(y; \tilde{\boldsymbol{W}} \boldsymbol{f}_{\hat{\boldsymbol{\theta}}}(\boldsymbol{x})) \ . \tag{6.2}$$

To analyze the impact of this additional constraint on the generalization error, we consider both the estimation error (a.k.a. variance) and the approximation error (a.k.a. bias) separately [Mohri et al., 2018, Chapter 4].

Estimation error. The sparsity constraint of Problem (6.2) decreases the size of the hypothesis class considered to minimize the negative log-likelihood and should thus yield a decrease in estimation error for both entangled and disentangled representations (*i.e.,* reduce overfitting). Sparsity regularization is a well-understood approach to control the complexity of a predictor, see for example Bickel et al. [2009], Lounici et al. [2011a], Mohri et al. [2018].

[2] $p(y; \boldsymbol{\eta})$ could be a Gaussian density (regression) or a categorical distribution (classification).

Approximation error. Disentangled and entangled representations differ in how the sparsity constraint of Problem (6.2) impacts their approximation errors. The following proposition will help us see how this regularization favors disentangled representations over entangled ones.

Proposition 6.2. *Let $\hat{\boldsymbol{W}}_\infty^{(\hat{\boldsymbol{\theta}})}$ be the (assumed unique) solution of the population-based MLE, $\arg\max_{\tilde{\boldsymbol{W}}} \mathbb{E}_{p(\boldsymbol{x},y)} \log p(y; \tilde{\boldsymbol{W}} \boldsymbol{f}_{\hat{\boldsymbol{\theta}}}(\boldsymbol{x}))$. If Assumption 6.1 (linear equivalence) & Assumption 6.2 (data generating process) hold, $\hat{\boldsymbol{W}}_\infty^{(\hat{\boldsymbol{\theta}})} = \boldsymbol{W}\boldsymbol{L}^{-1}$.*

From Proposition 6.2, one can see that if the representation $\boldsymbol{f}_{\hat{\boldsymbol{\theta}}}$ is disentangled ($\boldsymbol{L} = \boldsymbol{D}\boldsymbol{P}$), then

$$\|\hat{\boldsymbol{W}}_\infty^{(\hat{\boldsymbol{\theta}})}\|_{2,0} = \|\boldsymbol{W}(\boldsymbol{D}\boldsymbol{P})^{-1}\|_{2,0} = \|\boldsymbol{W}\|_{2,0} = \ell\,.$$

Thus, the sparsity constraint in Problem (6.2) does not exclude the population MLE estimator from its hypothesis class which means no approximation error is entailed (no bias). Contrarily, when $\boldsymbol{f}_{\hat{\boldsymbol{\theta}}}$ is linearly entangled, the population MLE might have more nonzero columns than the ground-truth (since $\boldsymbol{L}^{-1}$ might destroy the sparsity of $\boldsymbol{W}$), and thus would be excluded from the hypothesis space of Problem (6.2), which means an approximation error is introduced.

Conclusion. The above points suggest that *if the ground-truth task is sufficiently sparse, the disentangled representation should benefit from sparsity regularization (assuming the number of samples is low) because it reduces the estimation error (variance) without increasing the approximation error (bias).* In contrast, an entangled representation might not benefit from sparsity regularization if the increase in approximation error is more important than the reduction in estimation error.

Empirical validation (Figure 6.1). We now present a simple simulated experiment that illustrates the above claim that *disentangled representations coupled with sparsity regularization can yield better generalization.* Figure 6.1 compares the generalization performances of L_1 and L_2-penalized linear regressions [Tibshirani, 1996, Hoerl and Kennard, 1970], computed on the top of both disentangled and linearly entangled representations, which are frozen during training. L_1-penalized linear regression coupled with the disentangled representation yields better generalization than other alternatives when $\ell/m = 5\%$ and when the number of samples is very small. One can also see that disentanglement, sparsity regularization, and sufficient sparsity in the ground-truth data generating process are necessary for significant improvements, in line with our discussion. Lastly, all methods yield similar performance when the number of samples grows. More details and discussions can be found in Appendix D.1.

6.3. Sparse Multi-Task Learning for Disentanglement

In Section 6.2, we argued that disentangled representations can improve generalization when combined with sparse task-specific predictors, but we did not mention how to obtain a disentangled representation in the first place. In this section, we first provide a new identification result

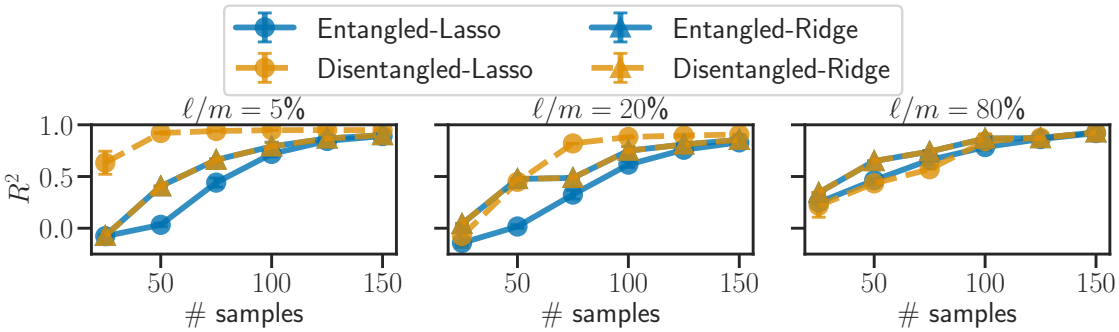


Figure 6.1. Test performance for the entangled and disentangled representation using Lasso and Ridge regression. All the results are averaged over 10 seeds, with standard error shown in error bars.

(Theorem 6.1, Section 6.3.2), which states that in the multi-task learning setting, regularizing the task-specific predictors to be sparse can yield disentangled representations. Then, in Section 6.3.4, we provide a practical way to learn disentangled representations motivated by our identifiability result.

6.3.1. Task & data generating process

Throughout this section, we assume the learner is given a set of T datasets $\{\mathcal{D}_1, \dots, \mathcal{D}_T\}$ where each dataset $\mathcal{D}_t := \{(\boldsymbol{x}^{(t,i)}, y^{(t,i)})\}_{i=1}^n$ consists of n couples of input $\boldsymbol{x} \in \mathbb{R}^d$ and label $y \in \mathcal{Y}$. The set of labels $\mathcal{Y}$ might contain either class indices or real values, depending on whether we are concerned with classification or regression tasks.

Our theory relies on the assumption that, for each task t, the dataset $\mathcal{D}_t$ is made of i.i.d. samples from the distribution

$$p(\boldsymbol{x}, y \mid \boldsymbol{W}^{(t)}) := p(y; \boldsymbol{W}^{(t)} \boldsymbol{f}_{\boldsymbol{\theta}}(\boldsymbol{x})) p(\boldsymbol{x} \mid \boldsymbol{W}^{(t)}) \,, \tag{6.3}$$

where $\boldsymbol{W}^{(t)} \in \mathbb{R}^{k \times m}$ is the task-specific ground-truth coefficient matrix. We emphasize that the representation $\boldsymbol{f}_{\boldsymbol{\theta}}$ is shared across all the tasks while the coefficient matrices $\boldsymbol{W}^{(t)}$ are task-specific. Also note that the distribution over $\boldsymbol{x}$ is allowed to change from one task to another. However, we assume that its support, $\mathcal{X}$, is fixed across tasks.

We further assume that the task-specific matrices $\boldsymbol{W}^{(t)}$ are i.i.d. samples from some probability measure $\mathbb{P}_{\boldsymbol{W}}$ with support $\mathcal{W}$. We will see in Section 6.3.3 that the most critical assumptions of our theory concern $\mathbb{P}_{\boldsymbol{W}}$.

6.3.2. Main identifiability result

We are now ready to show the main theoretical result of this work, which provides a bi-level optimization problem for which the optimal representations are guaranteed to be disentangled. It assumes infinitely many tasks are observed, with task-specific ground-truth matrices $\boldsymbol{W}$ sampled

from $\mathbb{P}_{\boldsymbol{W}}$. We denote by $\hat{\boldsymbol{W}}^{(\boldsymbol{W})}$ the task-specific estimator of $\boldsymbol{W}$. We delay the presentation of its technical assumptions to Section 6.3.3. See Appendix B.2 for a proof.

Theorem 6.1 (Sparse multi-task learning for disentanglement)**.** *Let $\hat{\boldsymbol{\theta}}$ be a minimizer of*

$$\min_{\hat{\boldsymbol{\theta}}} \mathbb{E}_{\mathbb{P}_{\boldsymbol{W}}} \mathbb{E}_{p(\boldsymbol{x},y|\boldsymbol{W})} - \log p(y; \hat{\boldsymbol{W}}^{(\boldsymbol{W})} \boldsymbol{f}_{\hat{\boldsymbol{\theta}}}(\boldsymbol{x})) \tag{6.4}$$
$$\text{s.t.} \quad \hat{\boldsymbol{W}}^{(\boldsymbol{W})} \in \underset{\substack{\tilde{\boldsymbol{W}} \text{ s.t.} \\ ||\tilde{\boldsymbol{W}}||_{2,0} \leq ||\boldsymbol{W}||_{2,0}}}{\arg\min} \mathbb{E}_{p(\boldsymbol{x},y|\boldsymbol{W})} - \log p(y; \tilde{\boldsymbol{W}} \boldsymbol{f}_{\hat{\boldsymbol{\theta}}}(\boldsymbol{x})) \ ,$$

where the constraint holds for all $\boldsymbol{W} \in \mathcal{W}$ and where $\mathbb{P}_{\boldsymbol{W}}$ and $p(\boldsymbol{x}, y \mid \boldsymbol{W})$ are described in Section 6.3.1. Under Assumptions 6.3, 6.4, 6.5, 6.6, 6.7 and if $\boldsymbol{f}_{\tilde{\boldsymbol{\theta}}}$ is continuous for all $\tilde{\boldsymbol{\theta}}$, $\boldsymbol{f}_{\hat{\boldsymbol{\theta}}}$ is disentangled w.r.t. $\boldsymbol{f}_{\boldsymbol{\theta}}$ (Definition 6.1).

Intuitively, this optimization problem effectively selects a representation $\boldsymbol{f}_{\hat{\boldsymbol{\theta}}}$ that (i) allows a perfect fit of the data distribution, and (ii) allows the task-specific estimators $\hat{\boldsymbol{W}}^{(\boldsymbol{W})}$ to be as sparse as the ground-truth $\boldsymbol{W}$. The theorem guarantees that such a representation must be disentangled.

Under the same assumptions and with the same disentanglement guarantees, Theorem 6.4 in Appendix B presents a variation of Problem (6.4) which enforces the weaker constraint $\mathbb{E}_{\mathbb{P}_{\boldsymbol{W}}} \|\hat{\boldsymbol{W}}^{(\boldsymbol{W})}\|_{2,0} \leq \mathbb{E}_{\mathbb{P}_{\boldsymbol{W}}} \|\boldsymbol{W}\|_{2,0}$, instead of $\|\hat{\boldsymbol{W}}^{(\boldsymbol{W})}\|_{2,0} \leq \|\boldsymbol{W}\|_{2,0}$ for each task $\boldsymbol{W}$ individually.

Characteristic features of our theory. (i) Contrary to most identifiability results for disentanglement (Section 6.4), we do not assume the observations $\boldsymbol{x}$ are generated by transforming a latent random vector $\boldsymbol{z}$ through a bijective decoder $\boldsymbol{g}$. Instead, we assume the existence of a **not necessarily invertible ground-truth feature extractor** $\boldsymbol{f}_{\boldsymbol{\theta}}(\boldsymbol{x})$ from which the labels can be predicted using only a subset of its components in every task. (ii) Most previous works make assumptions about the distribution of latent factors, e.g., (conditional) independence, exponential family or other parametric assumptions. In contrast, we make no such assumption except a rather weak assumption on the support of the ground-truth features (Assumption 6.4). Crucially, this allows for **statistically dependent latent factors**, which we explore empirically in Section 6.5.1.

6.3.3. Assumptions of Theorem 6.1

We now present the technical assumptions of Theorem 6.1. Perhaps unsurprisingly, the parameters $\boldsymbol{\eta}$ have to be identifiable from $p(y; \boldsymbol{\eta})$ in order for $\boldsymbol{f}_{\boldsymbol{\theta}}$ to be identifiable.

Assumption 6.3 (Identifiability of $\boldsymbol{\eta}$ from $p(y; \boldsymbol{\eta})$)**.** $\mathrm{KL}(p(y; \boldsymbol{\eta}) \,||\, p(y; \tilde{\boldsymbol{\eta}})) = 0 \implies \boldsymbol{\eta} = \tilde{\boldsymbol{\eta}}$*, where* KL *denotes the Kullback-Leibler divergence.*

This property holds, *e.g.*, when $p(y; \boldsymbol{\eta})$ is a Gaussian in the usual μ, σ^2 parameterization. Generally, it also holds for minimal parameterizations of exponential families [Wainwright and Jordan, 2008].

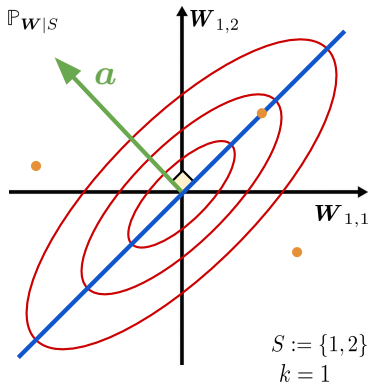


Figure 6.2. Illustration of Assumption 6.6 showing three examples of distribution $\mathbb{P}_{\boldsymbol{W}|S}$. The red distribution satisfies the assumption, but the blue and orange distributions do not. The red lines are level sets of a Gaussian distribution with full rank covariance. The blue line represents the support of a Gaussian distribution with a low-rank covariance. The orange dots represent a distribution with finite support. The green vector $\boldsymbol{a}$ shows that the condition is violated for both the blue and the orange distribution, since, in both cases, $\boldsymbol{W}_{1,S}$ and $\boldsymbol{a}$ are orthogonal ($\boldsymbol{W}_{1,S}\boldsymbol{a} = 0$) with probability greater than zero.

The following assumption requires the ground-truth representation $\boldsymbol{f}_\theta(\boldsymbol{x})$ to vary enough such that its image cannot be trapped inside a proper subspace.

Assumption 6.4 (Sufficient representation variability)**.** *There exists $\boldsymbol{x}^{(1)}, \ldots, \boldsymbol{x}^{(m)} \in \mathcal{X}$ such that the matrix $\boldsymbol{F} := [\boldsymbol{f}_{\boldsymbol{\theta}}(\boldsymbol{x}^{(1)}), \ldots, \boldsymbol{f}_{\boldsymbol{\theta}}(\boldsymbol{x}^{(m)})]$ is invertible.*

The following assumption requires that the support of the distribution $\mathbb{P}_{\boldsymbol{W}}$ is sufficiently rich.

Assumption 6.5 (Sufficient task variability)**.** *There exists $\boldsymbol{W}^{(1)}, \ldots, \boldsymbol{W}^{(m)} \in \mathcal{W}$ and indices $i_1, \ldots, i_m \in [k]$ such that the rows $\boldsymbol{W}^{(1)}_{i_1,:}, \ldots, \boldsymbol{W}^{(m)}_{i_m,:}$ are linearly independent.*

Under Assumptions 6.3, 6.4 and 6.5, the representation $\boldsymbol{f}_{\boldsymbol{\theta}}$ is identifiable up to linear equivalence (see Theorem 6.2 in Appendix B). Similar results were shown by Roeder et al. [2021], Ahuja et al. [2022c]. The next assumptions will guarantee disentanglement.

In order to formalize the intuitive idea that most tasks do not require all features, we will denote by $S^{(t)}$ the support of the matrix $\boldsymbol{W}^{(t)}$, *i.e.,*

$$S^{(t)} := \{j \in [m] \mid \boldsymbol{W}^{(t)}_{:j} \neq \boldsymbol{0}\}\,.$$

In other words, $S^{(t)}$ is the set of features which are useful to predict y in the t-th task; note that it is unknown to the learner. For our analysis, we decompose $\mathbb{P}_{\boldsymbol{W}}$ as

$$\mathbb{P}_{\boldsymbol{W}} = \sum_{S \in \mathcal{P}([m])} p(S)\mathbb{P}_{\boldsymbol{W}|S}\,, \tag{6.5}$$

where $\mathcal{P}([m])$ is the collection of all subsets of $[m]$, $p(S)$ is the probability that the support of $\boldsymbol{W}$ is S and $\mathbb{P}_{\boldsymbol{W}|S}$ is the conditional distribution of $\boldsymbol{W}$ given that its support is S. Let $\mathcal{S}$ be the support of

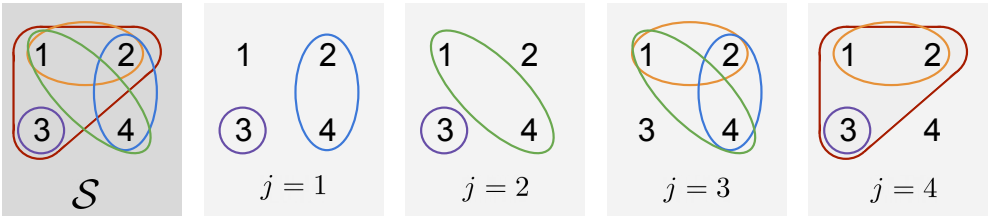


Figure 6.3. The leftmost figure represents $\mathcal{S}$, the set of task supports observed under the ground-truth distribution $p(S)$. The other figures form a verification that Assumption 6.7 holds for $\mathcal{S}$.

the distribution $p(S)$, *i.e.*, $\mathcal{S} := \{S \in \mathcal{P}([m]) \mid p(S) > 0\}$. The set $\mathcal{S}$ will have an important role in Assumption 6.7.

The following assumption requires that $\mathbb{P}_{\boldsymbol{W}|S}$ does not concentrate mass on certain proper subspaces.

Assumption 6.6 (Intra-support sufficient task variability)**.** *For all $S \in \mathcal{S}$ and all $\boldsymbol{a} \in \mathbb{R}^{|S|} \setminus \{0\}$,*

$$\mathbb{P}_{\boldsymbol{W}|S}\{\boldsymbol{W} \in \mathbb{R}^{k \times m} \mid \boldsymbol{W}_{:S}\boldsymbol{a} = \boldsymbol{0}\} = 0\,.$$

We illustrate the above assumption in the simpler case where $k = 1$. For instance, Assumption 6.6 holds when the distribution of $\boldsymbol{W}_{1,S} \mid S$ has a density w.r.t. the Lebesgue measure on $\mathbb{R}^{|S|}$, which is true for example when $\boldsymbol{W}_{1,S} \mid S \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{\Sigma})$ and the covariance matrix $\boldsymbol{\Sigma}$ is full rank (red distribution in Figure 6.2). However, if $\boldsymbol{\Sigma}$ is not full rank, the probability distribution of $\boldsymbol{W}_{1,S} \mid S$ concentrates its mass on a proper linear subspace $V \subsetneq \mathbb{R}^{|S|}$, which violates Assumption 6.6 (blue distribution in Figure 6.2). Another important counter-example is when $\mathbb{P}_{\boldsymbol{W}|S}$ concentrates some of its mass on a point $\boldsymbol{W}^{(0)}$, *i.e.*, $\mathbb{P}_{\boldsymbol{W}|S}\{\boldsymbol{W}^{(0)}\} > 0$ (orange distribution in Figure 6.2). We provide a concrete numerical example of what can go wrong when the support of the $\mathbb{P}_{\boldsymbol{W}|S}$ is finite in Appendix B.4. Interestingly, there are distributions over $\boldsymbol{W}_{1,S} \mid S$ that do not have a density w.r.t. the Lebesgue measure, but still satisfy Assumption 6.6. This is the case, e.g., when $\boldsymbol{W}_{1,S} \mid S$ puts uniform mass over a $(|S| - 1)$-dimensional sphere embedded in $\mathbb{R}^{|S|}$ and centered at zero. See Appendix B.6 for a justification.

The following assumption requires that the support $\mathcal{S}$ of $p(S)$ is "rich enough".

Assumption 6.7 (Sufficient variability of the task supports)**.** *For all $j \in [m]$,*

$$\bigcup_{S \in \mathcal{S} \mid j \notin S} S = [m] \setminus \{j\}\,.$$

Intuitively, Assumption 6.7 requires that, for every feature j, one can find a set of tasks such that their supports cover all features except j itself. Figure 6.3 shows an example of $\mathcal{S}$ satisfying Assumption 6.7. Appendix B.5 provides a probabilistic argument showing that Assumption 6.7 holds "in most cases" when the number of supports is very large. That being said, we conjecture that removing this assumption would yield a form of *partial disentanglement* resembling the one

developed by Lachapelle and Lacoste-Julien [2022] in which some groups of latent factors would remain entangled.

6.3.4. Tractable bilevel optimization problems for sparse multitask learning

The proposed approach to jointly estimate the representation and the task-specific predictors relies on a bilevel optimization problem (Problem (6.4)) that is intractable because of the non-convex constraints. To obtain a tractable bi-level optimization problem, the $L_{2,0}$ constraints are replaced by their convex relaxations in the penalized form, which are also known to promote group sparsity [Argyriou et al., 2008]:

$$\begin{aligned} \min_{\hat{\boldsymbol{\theta}}} \quad & -\frac{1}{Tn}\sum_{t=1}^{T}\sum_{(\boldsymbol{x},y)\in\mathcal{D}_t}\log p(y;\hat{\boldsymbol{W}}^{(t)}\boldsymbol{f}_{\hat{\boldsymbol{\theta}}}(\boldsymbol{x})) \\ \text{s.t. } & \hat{\boldsymbol{W}}^{(t)} \in \arg\min_{\tilde{\boldsymbol{W}}} \frac{1}{n}\sum_{(\boldsymbol{x},y)\in\mathcal{D}_t} -\log p(y;\tilde{\boldsymbol{W}}\boldsymbol{f}_{\hat{\boldsymbol{\theta}}}(\boldsymbol{x})) + \lambda_t||\tilde{\boldsymbol{W}}||_{2,1} \ , \end{aligned} \tag{6.6}$$

where the constraint holds for all $t \in [T]$. Following Bengio [2000], Pedregosa [2016], one can compute the (hyper)gradient of the outer function using implicit differentiation, even if the inner optimization problem is non-smooth [Bertrand et al., 2020, Bolte et al., 2021, Malézieux et al., 2022, Bolte et al., 2022]. Once the hypergradient is computed, one can optimize Problem (6.6) with usual first-order methods [Wright and Nocedal, 1999].

Note that the quantity $\hat{\boldsymbol{W}}^{(t)}\boldsymbol{f}_{\hat{\boldsymbol{\theta}}}(\boldsymbol{x})$ is invariant to simultaneous rescaling of $\hat{\boldsymbol{W}}^{(t)}$ by a scalar and of $\boldsymbol{f}_{\hat{\boldsymbol{\theta}}}(\boldsymbol{x})$ by its inverse. Thus, without constraints on $\boldsymbol{f}_{\hat{\boldsymbol{\theta}}}(\boldsymbol{x})$, $\|\hat{\boldsymbol{W}}^{(t)}\|_{2,1}$ can be made arbitrarily small. This issue is similar to the one faced in sparse dictionary learning [Kreutz-Delgado et al., 2003, Mairal et al., 2008, 2009, 2011], where unit-norm constraints are usually imposed on dictionary columns. In our case, since $\boldsymbol{f}_{\hat{\boldsymbol{\theta}}}$ is parametrized by a neural network, we suggest applying batch or layer normalization [Ioffe and Szegedy, 2015, Ba et al., 2016] to control the norm of $\boldsymbol{f}_{\hat{\boldsymbol{\theta}}}(\boldsymbol{x})$. Since the number of relevant features might be task-dependent, Problem (6.6) has one regularization hyperparameter λ_t per task. However, in practice, we select $\lambda_t := \lambda$ for all $t \in [T]$ to limit the number of hyperparameters. We also use an adaptive scheme to have λ in a reasonable range throughout training, which we explain in Appendix D.2.3.

Appendix B.3 introduces a similar relaxation of Theorem 6.4 (mentioned in Section 6.3.2) in which the sparsity penalty appears in the outer problem instead of the inner problem. Appendix D.2.5 presents empirical results showing this alternative approach yields very similar results.

Link with meta-learning. The bi-level formulation Problem (6.6) is closely related to *metric-based meta-learning* methods [Snell et al., 2017, Bertinetto et al., 2019], where a shared representation $\boldsymbol{f}_{\hat{\boldsymbol{\theta}}}$ is learned across all tasks via simple task-specific predictors, such as linear classifiers. In the general meta-learning setting [Finn et al., 2017], one is given a large number of training

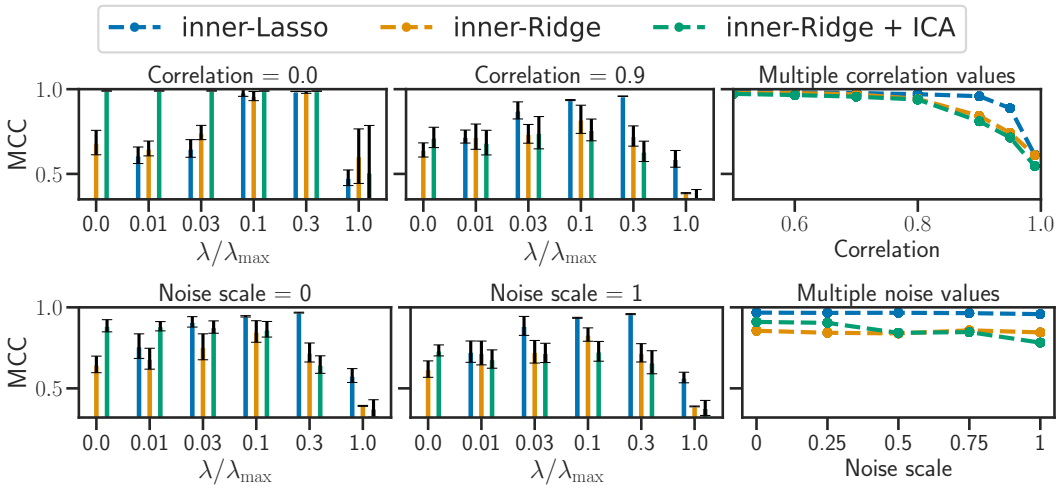


Figure 6.4. Disentanglement performance (MCC) for all three methods considered as a function of the regularization parameter (left and middle). Varying level of correlation between latents (top) and noise on the latents (bottom). The right columns show performances of the best hyperparameter for different values of correlation and noise. We explain what is $\lambda_{\max}$ in Appendix D.2.3.

datasets $(\mathcal{D}_t^{\text{train}})_{1\leq t\leq T}$, which usually only contain a small number of samples n. As opposed to the multi-task setting (*i.e.,* unlike in Section 6.3.1), one is also given separate *test datasets* $(\mathcal{D}_t^{\text{test}})_{1\leq t\leq T}$ of n' samples for each task t, to evaluate how well the learned model generalizes to new test samples. In meta-learning, the goal is to *learn a learning procedure* that will generalize well on new unseen tasks.

Formally, metric-based meta-learning can be formulated as

$$\min_{\hat{\boldsymbol{\theta}}} \frac{1}{Tn'} \sum_{t=1}^{T} \sum_{(\boldsymbol{x},y)\in\mathcal{D}_t^{\text{test}}} \mathcal{L}_{\text{out}}\big(\hat{\boldsymbol{W}}_{\hat{\boldsymbol{\theta}}}^{(t)}; f_{\hat{\boldsymbol{\theta}}}(\boldsymbol{x}), y\big) \tag{6.7}$$

$$\text{s.t. } \hat{\boldsymbol{W}}_{\hat{\boldsymbol{\theta}}}^{(t)} \in \arg\min_{\tilde{\boldsymbol{W}}} \frac{1}{n} \sum_{(\boldsymbol{x},y)\in\mathcal{D}_t^{\text{train}}} \mathcal{L}_{\text{in}}\big(\tilde{\boldsymbol{W}}; f_{\hat{\boldsymbol{\theta}}}(\boldsymbol{x}), y\big) \ .$$

The main difference between Problem (6.6) and (6.7) is that, in the latter, the inner and outer loss functions $\mathcal{L}_{\text{in}}$ and $\mathcal{L}_{\text{out}}$ are not evaluated on the same dataset. Section 6.5.2 shows experiments with a meta-learning variant of Problem (6.6) based on group Lasso multiclass SVM predictors.

6.4. Related Work

Disentanglement. Since the work of Bengio et al. [2013], many methods have been proposed to learn disentangled representations based on various heuristics [Higgins et al., 2017, Chen et al., 2018, Kim and Mnih, 2018, Kumar et al., 2018, Bouchacourt et al., 2018]. Following the work of Locatello et al. [2019], which highlighted the lack of identifiability in modern deep generative models, many works have proposed more or less weak forms of supervision motivated by identifiability analyses [Locatello et al., 2020a, Klindt et al., 2021, Von Kügelgen et al., 2021, Ahuja

et al., 2022a,c, Zheng et al., 2022]. A similar line of work have adopted the causal representation learning perspective [Lachapelle et al., 2022, Lachapelle and Lacoste-Julien, 2022, Lippe et al., 2022, 2023b, Ahuja et al., 2022b, Yao et al., 2022c, Brehmer et al., 2022].

The problem of identifiability was well known among the *independent component analysis* (ICA) community [Hyvärinen et al., 2001, Hyvärinen and Pajunen, 1999] which came up with solutions for general nonlinear mixing functions by leveraging auxiliary information [Hyvarinen and Morioka, 2016, 2017, Hyvärinen et al., 2019, Khemakhem et al., 2020a,b]. Another approach is to consider restricted hypothesis classes of mixing functions [Taleb and Jutten, 1999, Gresele et al., 2021, Zheng et al., 2022, Moran et al., 2022]. Locatello et al. [2020b] proposed a semi-supervised learning approach to disentangle in cases where a few samples are labelled with the values of the factors of variations themselves. This is different from our approach as the labels that we consider can be sampled from some $p(y; \boldsymbol{W} \boldsymbol{f}_{\hat{\theta}}(\boldsymbol{x}))$, which is more general. Ahuja et al. [2022c] consider a setting similar to ours, but they rely on the independence and non-gaussianity of the latent factors for disentanglement using linear ICA. See the end of Section 6.3.2 for further discussions on how our theory distinguishes itself from most methods cited above.

Multi-task, transfer & invariant learning. While the statistical advantages of multi-task representation learning are well understood [Lounici et al., 2011a,b, Maurer et al., 2016], the theoretical benefits of disentanglement for transfer learning are not clearly established (apart from Zhang et al. 2022). Some works have investigated this question empirically and obtained both positive [van Steenkiste et al., 2019, Miladinović et al., 2019, Dittadi et al., 2021] and negative results [Locatello et al., 2019, Montero et al., 2021]. Invariant risk minimization [Arjovsky et al., 2020, Ahuja et al., 2020, Krueger et al., 2021a, Lu et al., 2021] aims at learning a representation that elicits a single predictor that is optimal for all tasks. This differs from our approach which learns one predictor per task.

Dictionary learning and sparse coding. We contrast our approach, which jointly learns a *dense representation* and sparse task-specific predictors (Problem (6.6)), with the line of work which consists in learning *sparse representations* [Chen et al., 1998, Gribonval and Lesage, 2006]. For instance, sparse dictionary learning [Mairal et al., 2009, 2011, Maurer et al., 2013] is an unsupervised technique that aims at learning a dictionary of *atoms* used to reconstruct inputs via sparse linear combinations of its elements. The representation of a single input consists of the coefficients of the linear combination of atoms that minimizes a sparsity-regularized reconstruction loss. In the case of supervised dictionary learning [Mairal et al., 2008], an additional (potentially expressive) classifier is learned on top of that representation. This large literature has led to a wide variety of estimators: for instance, Mairal et al. [2008, Eq. 4], which minimizes the sum of the classification error and the approximation error of the code, or Mairal et al. [2011], which introduces bi-level formulations. While sharing similar optimization challenges, our method is conceptually different and computes the representation of a single input $\boldsymbol{x}$ by evaluating the learned function $\boldsymbol{f}_{\hat{\theta}}$.

6.5. Experiments

We present experiments on disentanglement and few-shot learning. Our implementation relies on `jax` and `jaxopt` [Bradbury et al., 2018, Blondel et al., 2022] and is available here: `https://github.com/tristandeleu/synergies-disentanglement-sparsity`.

6.5.1. Disentanglement in 3D Shapes

We now illustrate Theorem 6.1 by applying Problem (6.6) to tasks generated using the 3D Shapes dataset [Burgess and Kim, 2018].

Data generation. For all tasks t, the labelled dataset $\mathcal{D}_t = \{(\boldsymbol{x}^{(t,i)}), y^{(t,i)})\}_{i=1}^n$ is generated by first sampling the ground-truth latent variables $\boldsymbol{z}^{(t,i)}$ i.i.d. according to some distribution $p(\boldsymbol{z})$, while the corresponding input is obtained doing $\boldsymbol{x}^{(t,i)} := \boldsymbol{f}_{\boldsymbol{\theta}}^{-1}(\boldsymbol{z}^{(t,i)})$ ($\boldsymbol{f}_{\boldsymbol{\theta}}$ is invertible in 3D Shapes). Then, a sparse weight vector $\boldsymbol{w}^{(t)}$ is sampled randomly to compute the labels of each example as $y^{(t,i)} := \boldsymbol{w}^{(t)} \cdot \boldsymbol{z}^{(t,i)} + \epsilon^{(t,i)}$, where $\epsilon^{(t,i)}$ is independent Gaussian noise. Figure 6.4 explores various choices of $p(\boldsymbol{z})$ by varying the level of correlation between the latent variables and by varying the level of noise on the ground-truth latents. See Appendix D.2 for more details about the data generating process and Figure 6.7 to visualize various $p(\boldsymbol{z})$.

Algorithms. In this setting where $p(y; \boldsymbol{\eta})$ is a Gaussian with fixed variance, the inner problem of Problem (6.6) amounts to Lasso regression, we thus refer to this approach as inner-Lasso. We also evaluate a simple variation of Problem (6.6) in which the L_1 norm is replaced by an L_2 norm and refer to it as inner-Ridge. In addition, we evaluate the representation obtained by performing linear ICA [Comon, 1992] on the representation learned by inner-Ridge: the case $\lambda = 0$ corresponds to the approach of Ahuja et al. [2022c].

Discussion. Figure 6.4 reports disentanglement performances of the three methods, as measured by the *mean correlation coefficient*, or MCC [Hyvarinen and Morioka, 2016, Khemakhem et al., 2020a] (Appendix D.2). In all settings, inner-Lasso obtains high MCC for some values of λ, being on par or surpassing the baselines. As the theory suggests, it is robust to high levels of correlations between the latents, as opposed to inner-Ridge with ICA which is very much affected by strong correlations (since ICA assumes independence). We can also see how additional noise on the latent variables hurts inner-Ridge with ICA while leaving inner-Lasso unaffected. Figure 6.6 in Appendix D.2 shows that all methods find a representation which is linearly equivalent to the ground-truth representation, except for very large values of λ. Appendix D.2.4 studies empirically to what extent inner-Lasso is robust to violations of Assumption 6.7, Appendix D.2.6 presents a visual evaluation of disentanglement and Appendix D.2.7 reports the DCI metric [Eastwood and Williams, 2018] on the same experiments. We did not explore hyperparameter selection in this work, which is a difficult problem for disentanglement because a goodness-of-fit score evaluated on a held-out dataset will not be informative because of the lack of identifiability. Nevertheless, one

can use heuristics such as the *unsupervised disentanglement ranking* score proposed by Duan et al. [2020].

6.5.2. Sparse task-specific predictors in few-shot learning

Despite the lack of ground-truth latent factors in standard few-shot learning benchmarks, we also evaluate sparse meta-learning objectives on the *mini*ImageNet dataset [Vinyals et al., 2016]. The purpose of this experiment is to show that the sparse formulation of standard metric-based meta-learning techniques reaches similar performance while using a fraction of the features (Figure 6.5, right).

Inspired by Lee et al. [2019], where the task-specific classifiers are multiclass support-vector machines (SVMs, Crammer and Singer 2001), we propose to use group Lasso penalized multiclass SVMs, to introduce sparsity in the classifiers. Using the notation of (6.7), we choose

$$\mathcal{L}_{\text{in}}(\boldsymbol{W}; f_{\hat{\boldsymbol{\theta}}}(\boldsymbol{x}_i), \boldsymbol{y}_i) = \max_{l \in [k]} \left((\boldsymbol{W}_{\boldsymbol{y}_i:} - \boldsymbol{W}_{l:}) \cdot f_{\hat{\boldsymbol{\theta}}}(\boldsymbol{x}_i) - \boldsymbol{Y}_{il} \right) \tag{6.8}$$

$$\mathcal{L}_{\text{out}}(\boldsymbol{W}; f_{\hat{\boldsymbol{\theta}}}(\boldsymbol{x}_i), \boldsymbol{y}_i) = \text{CE}(\boldsymbol{W} f_{\hat{\boldsymbol{\theta}}}(\boldsymbol{x}_i), \boldsymbol{Y}_{i:}) \ , \tag{6.9}$$

with $\boldsymbol{Y} \in \mathbb{R}^{n \times k}$ the one-hot encoding of $\boldsymbol{y} \in \mathbb{R}^n$ and CE the cross-entropy. The difference with Lee et al. [2019] is the sparsity-promoting term $\|\boldsymbol{W}\|_{2,1}$, which makes the bi-level optimization problem harder to solve. That is why we propose solving the dual [Boyd et al., 2004, Chap. 5] of this inner optimization problem, which writes

$$\begin{aligned} &\min_{\boldsymbol{\Lambda} \in \mathbb{R}^{n \times k}} \frac{1}{\lambda_2} \sum_{j=1}^{m} \| \text{BST}\left((\boldsymbol{Y} - \boldsymbol{\Lambda})^\top \boldsymbol{F}_{:j}, \lambda_1 \right) \|^2 + \langle \boldsymbol{Y}, \boldsymbol{\Lambda} \rangle \\ &\text{s.t. } \forall i, l, \in [n] \times [k], \ \sum_{l'=1}^{k} \boldsymbol{\Lambda}_{il'} = 1 \ \text{ and } \ \boldsymbol{\Lambda}_{il} \geq 0 \ , \end{aligned} \tag{6.10}$$

with $\text{BST} : (\boldsymbol{a}, \tau) \mapsto (1 - \tau / \|\boldsymbol{a}\|)_+ \boldsymbol{a}$ is the block soft-thresholding operator, $\boldsymbol{F} \in \mathbb{R}^{n \times m}$ the concatenation of $\{\boldsymbol{f}_{\hat{\boldsymbol{\theta}}}(x)\}_{(x,y) \in \mathcal{D}^{\text{train}}}$. In addition, the primal-dual link writes, $\forall j \in [m]$, $\boldsymbol{W}_{:j} = \text{BST}\left((\boldsymbol{Y} - \boldsymbol{\Lambda})^\top \boldsymbol{F}_{:j}, \lambda_1 \right) / \lambda_2$. The derivation of the dual can be found in Appendix C.1, Solving this kind of problem in the dual is standard in the SVM literature: it has been proven to be computationally advantageous [Hsieh et al., 2008] when the number of features m is significantly larger than the number of samples n (here $m = 1.6 \times 10^4$ and $n \leq 25$). Details on how to solve and differentiate through Problem (6.10) are in Appendix D.3.

Discussion. In Figure 6.5 (right), we observe that the accuracy of the sparse meta-learning method on novel (meta-validation) tasks is similar to the dense counterpart ($\lambda = 0$), while using only a few of the features available (around 30% of sparsity, with no impact on the performance). Naturally, the performance starts to drop as the sparsity level increases though, albeit being still

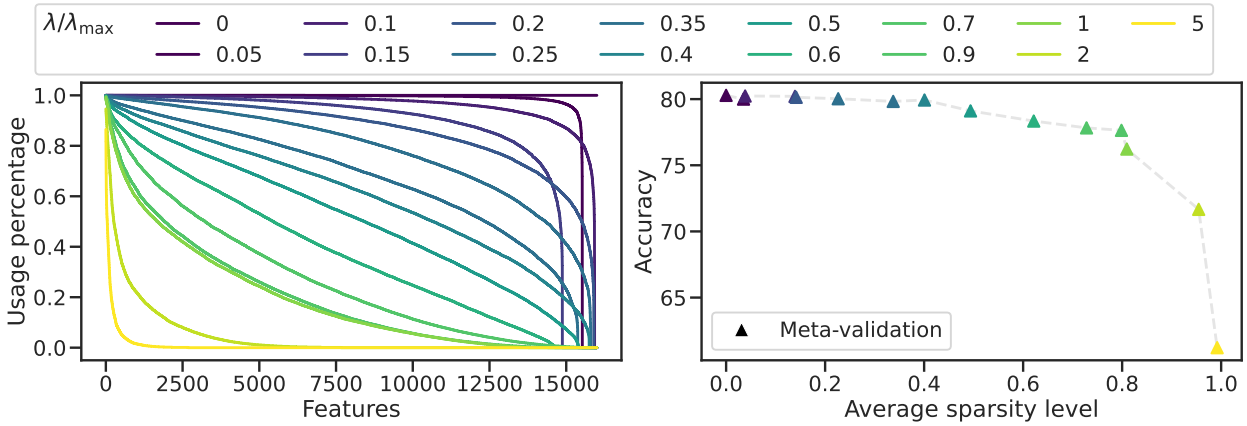


Figure 6.5. *Left.* Effect of sparsity on the percentage of tasks using specific features, with our meta-learning objective, on *mini*ImageNet. *Right.* The meta-validation accuracy of the meta-learning algorithm against the average level of sparsity in the task-specific predictor, for different values of λ.

competitive. We also report in Figure 6.5 (left) how frequently each feature in the learned representation is used by the task-specific predictors on meta-validation tasks (sorted by usage, for each λ). The gradual decrease in usage suggests that the features are reused in different contexts, across different tasks.

6.6. Conclusion

In this work, we investigated the synergies between sparsity, disentanglement and generalization. We showed that when the downstream task can be solved using only a fraction of the factors of variations, disentangled representations combined with sparse task-specific predictors can improve generalization (Section 6.2). Our novel identifiability result (Theorem 6.1) sheds light on how, in a multi-task setting, sparsity regularization on the task-specific predictors can induce disentanglement. This led to a practical bi-level optimization problem that was shown to yield disentangled representations on regression tasks based on the 3D Shapes dataset. Finally, we explored the connection between this bi-level formulation and meta-learning, and we showed how sparse task-specific predictors may achieve similar performance on unseen tasks with only a fraction of the features. Future work could explore identifiability in a more general setting where the task-specific predictors are potentially nonlinear, which should be applicable to more problems.

Appendices of Chapter 6

A. Proofs of Section 6.2

Proposition 6.1. *Let $\hat{\boldsymbol{W}}_n^{(\hat{\boldsymbol{\theta}})}$ and $\hat{\boldsymbol{W}}_n^{(\boldsymbol{\theta})}$ be the solutions to Problem (6.1) with the representations $\boldsymbol{f}_{\hat{\boldsymbol{\theta}}}$ and $\boldsymbol{f}_{\boldsymbol{\theta}}$, respectively (which we assume are unique). If $\boldsymbol{f}_{\hat{\boldsymbol{\theta}}}$ and $\boldsymbol{f}_{\boldsymbol{\theta}}$ are linearly equivalent (Assumption 6.1), then we have, $\forall \boldsymbol{x} \in \mathcal{X}$, $\hat{\boldsymbol{W}}_n^{(\hat{\boldsymbol{\theta}})} \boldsymbol{f}_{\hat{\boldsymbol{\theta}}}(\boldsymbol{x}) = \hat{\boldsymbol{W}}_n^{(\boldsymbol{\theta})} \boldsymbol{f}_{\boldsymbol{\theta}}(\boldsymbol{x})$.*

Proof By definition of $\hat{\boldsymbol{W}}^{(\hat{\boldsymbol{\theta}})}$, we have that, for all $\hat{\boldsymbol{W}} \in \mathbb{R}^{k \times m}$,

$$\sum_{(\boldsymbol{x},y)\in\mathcal{D}} \log p(y; \hat{\boldsymbol{W}}^{(\hat{\boldsymbol{\theta}})} \boldsymbol{f}_{\hat{\boldsymbol{\theta}}}(\boldsymbol{x})) \geq \sum_{(\boldsymbol{x},y)\in\mathcal{D}} \log p(y; \hat{\boldsymbol{W}} \boldsymbol{f}_{\hat{\boldsymbol{\theta}}}(\boldsymbol{x})) \tag{6.11}$$

$$\sum_{(\boldsymbol{x},y)\in\mathcal{D}} \log p(y; \hat{\boldsymbol{W}}^{(\hat{\boldsymbol{\theta}})} \boldsymbol{L} \boldsymbol{f}_{\boldsymbol{\theta}}(\boldsymbol{x})) \geq \sum_{(\boldsymbol{x},y)\in\mathcal{D}} \log p(y; \hat{\boldsymbol{W}} \boldsymbol{L} \boldsymbol{f}_{\boldsymbol{\theta}}(\boldsymbol{x})) \,. \tag{6.12}$$

Because $\mathbb{R}^{k\times m}\boldsymbol{L} = \mathbb{R}^{k\times m}$, we have that, for all $\hat{\boldsymbol{W}} \in \mathbb{R}^{k\times m}$,

$$\sum_{(\boldsymbol{x},y)\in\mathcal{D}} \log p(y; \hat{\boldsymbol{W}}^{(\hat{\boldsymbol{\theta}})} \boldsymbol{L} \boldsymbol{f}_{\boldsymbol{\theta}}(\boldsymbol{x})) \geq \sum_{(\boldsymbol{x},y)\in\mathcal{D}} \log p(y; \hat{\boldsymbol{W}} \boldsymbol{f}_{\boldsymbol{\theta}}(\boldsymbol{x})) \,, \tag{6.13}$$

which is to say that $\hat{\boldsymbol{W}}^{(\boldsymbol{\theta})} = \hat{\boldsymbol{W}}^{(\hat{\boldsymbol{\theta}})}\boldsymbol{L}$, or put differently, $\hat{\boldsymbol{W}}^{(\hat{\boldsymbol{\theta}})} = \hat{\boldsymbol{W}}^{(\boldsymbol{\theta})}\boldsymbol{L}^{-1}$. It implies

$$\hat{\boldsymbol{W}}^{(\hat{\boldsymbol{\theta}})} \boldsymbol{f}_{\hat{\boldsymbol{\theta}}}(\boldsymbol{x}) = \hat{\boldsymbol{W}}^{(\boldsymbol{\theta})} \boldsymbol{L}^{-1} \boldsymbol{L} \boldsymbol{f}_{\boldsymbol{\theta}}(\boldsymbol{x}) = \hat{\boldsymbol{W}}^{(\boldsymbol{\theta})} \boldsymbol{f}_{\boldsymbol{\theta}}(\boldsymbol{x}) \,, \tag{6.14}$$

which is what we wanted to show. ■

Proposition 6.2. *Let $\hat{\boldsymbol{W}}_\infty^{(\hat{\boldsymbol{\theta}})}$ be the (assumed unique) solution of the population-based MLE, $\arg\max_{\tilde{\boldsymbol{W}}} \mathbb{E}_{p(\boldsymbol{x},y)} \log p(y; \tilde{\boldsymbol{W}} \boldsymbol{f}_{\hat{\boldsymbol{\theta}}}(\boldsymbol{x}))$. If Assumption 6.1 (linear equivalence) & Assumption 6.2 (data generating process) hold, $\hat{\boldsymbol{W}}_\infty^{(\hat{\boldsymbol{\theta}})} = \boldsymbol{W}\boldsymbol{L}^{-1}$.*

Proof By definition of $\hat{\boldsymbol{W}}_\infty^{(\hat{\boldsymbol{\theta}})}$, we have that, for all $\tilde{\boldsymbol{W}} \in \mathbb{R}^{k\times m}$,

$$\mathbb{E}_{p(\boldsymbol{x},y)} \log p(y; \hat{\boldsymbol{W}}_\infty^{(\hat{\boldsymbol{\theta}})} \boldsymbol{f}_{\hat{\boldsymbol{\theta}}}(\boldsymbol{x})) \geq \mathbb{E}_{p(\boldsymbol{x},y)} \log p(y; \tilde{\boldsymbol{W}} \boldsymbol{f}_{\hat{\boldsymbol{\theta}}}(\boldsymbol{x})) \tag{6.15}$$

$$\mathbb{E}_{p(\boldsymbol{x},y)} \log p(y; \hat{\boldsymbol{W}}_\infty^{(\hat{\boldsymbol{\theta}})} \boldsymbol{L} \boldsymbol{f}_{\boldsymbol{\theta}}(\boldsymbol{x})) \geq \mathbb{E}_{p(\boldsymbol{x},y)} \log p(y; \tilde{\boldsymbol{W}} \boldsymbol{L} \boldsymbol{f}_{\boldsymbol{\theta}}(\boldsymbol{x})) \,. \tag{6.16}$$

		Norms & pseudonorms
$\|\cdot\|$		Euclidean/Frobenius norm on vectors/matrices
$\|\boldsymbol{A}\|_{2,1}$	$:=$	$\sum_{j=1}^{m} \|\boldsymbol{A}_{:j}\|$
$\|\boldsymbol{A}\|_{2,0}$	$:=$	$\sum_{j=1}^{m} \mathbb{1}_{\|\boldsymbol{A}_{:j}\|\neq 0}$, where $\mathbb{1}$ is the indicator function.
		Data
$\boldsymbol{x} \in \mathbb{R}^d$		Observations
$\mathcal{X} \subseteq \mathbb{R}^d$		Support of observations
$y \in \mathbb{R}$		Target
$\mathcal{Y} \subseteq \mathbb{R}$		Support of targets
		Learned/ground-truth model
$\boldsymbol{W} \in \mathbb{R}^{k\times m}$		Ground-truth coefficients
$\hat{\boldsymbol{W}} \in \mathbb{R}^{k\times m}$		Learned coefficients
$\boldsymbol{\theta}$		Ground-truth parameters of the representation
$\hat{\boldsymbol{\theta}}$		Learned parameters of the representation
$f_{\boldsymbol{\theta}} : \mathbb{R}^d \to \mathbb{R}^m$		Ground-truth representation
$f_{\hat{\boldsymbol{\theta}}} : \mathbb{R}^d \to \mathbb{R}^m$		Learned representation
$\boldsymbol{\eta} \in \mathbb{R}^k$		Parameter of the distribution $p(y; \boldsymbol{\eta})$
$\mathbb{P}_{\boldsymbol{W}}$		Distribution over ground-truth coefficient matrices $\boldsymbol{W}$
S	$:=$	$\{j \in [m] \mid \boldsymbol{W}_{:j} \neq \boldsymbol{0}\}$ (support of $\boldsymbol{W}$)
$\mathbb{P}_{\boldsymbol{W}\|S}$		Conditional distribution of $\boldsymbol{W}$ given S.
$p(S)$		Ground-truth distribution over possible supports S
$\mathcal{S}$		Support of the distribution $p(S)$
		Optimization
W		Primal variable
Λ		Dual variable
$h^* : \boldsymbol{a}$	$\mapsto$	$\sup_{\boldsymbol{b}\in\mathbb{R}^d} \langle \boldsymbol{a}, \boldsymbol{b}\rangle - h(\boldsymbol{b})$, Fenchel conjugate of the function $h : \mathbb{R}^d \to \mathbb{R}$
$f \square g : \boldsymbol{a}$	$\mapsto$	$\min_b f(\boldsymbol{a} - \boldsymbol{b}) + g(\boldsymbol{b})$, inf-convolution of the functions f and g
$\mathrm{BST} : (\boldsymbol{a}, \tau)$	$\mapsto$	$(1 - \tau/\|\boldsymbol{a}\|)_+ \boldsymbol{a}$, block soft-thresholding operator

Table 6.1. Table of notations.

In particular, the inequality holds for $\tilde{\boldsymbol{W}} := \boldsymbol{W}\boldsymbol{L}^{-1}$, which yields

$$\mathbb{E}_{p(\boldsymbol{x},y)} \log p(y; \hat{\boldsymbol{W}}_\infty^{(\hat{\boldsymbol{\theta}})} \boldsymbol{L} \boldsymbol{f}_{\boldsymbol{\theta}}(\boldsymbol{x})) \geq \mathbb{E}_{p(\boldsymbol{x},y)} \log p(y; \boldsymbol{W} \boldsymbol{f}_{\boldsymbol{\theta}}(\boldsymbol{x})) \tag{6.17}$$

$$0 \geq \mathbb{E}_{p(\boldsymbol{x},y)} \left[\log p(y; \boldsymbol{W} \boldsymbol{f}_{\boldsymbol{\theta}}(\boldsymbol{x})) - \log p(y; \hat{\boldsymbol{W}}_\infty^{(\hat{\boldsymbol{\theta}})} \boldsymbol{L} \boldsymbol{f}_{\boldsymbol{\theta}}(\boldsymbol{x})) \right] \tag{6.18}$$

$$0 \geq \mathbb{E}_{p(\boldsymbol{x})} \mathrm{KL}(p(y; \boldsymbol{W} \boldsymbol{f}_{\boldsymbol{\theta}}(\boldsymbol{x})) \,||\, p(y; \hat{\boldsymbol{W}}_\infty^{(\hat{\boldsymbol{\theta}})} \boldsymbol{L} \boldsymbol{f}_{\boldsymbol{\theta}}(\boldsymbol{x}))) \,. \tag{6.19}$$

Since the KL is always non-negative, we have that,

$$\mathbb{E}_{p(\boldsymbol{x})}\mathrm{KL}(p(y;\boldsymbol{W}\boldsymbol{f}_{\boldsymbol{\theta}}(\boldsymbol{x}))\;||\;p(y;\hat{\boldsymbol{W}}_{\infty}^{(\hat{\boldsymbol{\theta}})}\boldsymbol{L}\boldsymbol{f}_{\boldsymbol{\theta}}(\boldsymbol{x})))=0\,, \tag{6.20}$$

which in turn implies

$$\mathbb{E}_{p(\boldsymbol{x},y)}\log p(y;\hat{\boldsymbol{W}}_{\infty}^{(\hat{\boldsymbol{\theta}})}\boldsymbol{L}\boldsymbol{f}_{\boldsymbol{\theta}}(\boldsymbol{x}))=\mathbb{E}_{p(\boldsymbol{x},y)}\log p(y;\boldsymbol{W}\boldsymbol{f}_{\boldsymbol{\theta}}(\boldsymbol{x})) \tag{6.21}$$

$$\mathbb{E}_{p(\boldsymbol{x},y)}\log p(y;\hat{\boldsymbol{W}}_{\infty}^{(\hat{\boldsymbol{\theta}})}\boldsymbol{L}\boldsymbol{f}_{\boldsymbol{\theta}}(\boldsymbol{x}))=\mathbb{E}_{p(\boldsymbol{x},y)}\log p(y;\boldsymbol{W}\boldsymbol{L}^{-1}\boldsymbol{L}\boldsymbol{f}_{\boldsymbol{\theta}}(\boldsymbol{x})) \tag{6.22}$$

$$\mathbb{E}_{p(\boldsymbol{x},y)}\log p(y;\hat{\boldsymbol{W}}_{\infty}^{(\hat{\boldsymbol{\theta}})}\boldsymbol{f}_{\hat{\boldsymbol{\theta}}}(\boldsymbol{x}))=\mathbb{E}_{p(\boldsymbol{x},y)}\log p(y;\boldsymbol{W}\boldsymbol{L}^{-1}\boldsymbol{f}_{\hat{\boldsymbol{\theta}}}(\boldsymbol{x})) \tag{6.23}$$

$$\tag{6.24}$$

Since the solution to the population MLE from Problem (6.2) is assumed to be unique, this equality holds if and only if $\hat{\boldsymbol{W}}_{\infty}^{(\hat{\boldsymbol{\theta}})}=\boldsymbol{W}\boldsymbol{L}^{-1}$. ■

B. Proofs of Section 6.3

B.1. Technical Lemmas

The lemmas of this section can be skipped at first read.

The following lemma will be important for proving Theorem 6.3. The argument is taken from Lachapelle et al. [2022].

Lemma 6.1 (Sparsity pattern of an invertible matrix contains a permutation)**.** *Let $\boldsymbol{L}\in\mathbb{R}^{m\times m}$ be an invertible matrix. Then, there exists a permutation σ such that $\boldsymbol{L}_{i,\sigma(i)}\neq 0$ for all i.*

Proof Since the matrix $\boldsymbol{L}$ is invertible, its determinant is non-zero, *i.e.,*

$$\det(\boldsymbol{L}):=\sum_{\sigma\in\mathfrak{S}_m}\operatorname{sign}(\sigma)\prod_{i=1}^{m}\boldsymbol{L}_{i,\sigma(i)}\neq 0\,, \tag{6.25}$$

where $\mathfrak{S}_m$ is the set of m-permutations. This equation implies that at least one term of the sum is non-zero, meaning there exists $\sigma\in\mathfrak{S}_m$ such that for all $i\in[m]$, $\boldsymbol{L}_{i,\sigma(i)}\neq 0$. ■

The following technical lemma will help us dealing with almost-everywhere statements and can be safely skipped at a first read. Before presenting it, we recall the formal definition of a support of a distribution.

Definition 6.2. *The support of a Borel measure μ over a topological space (X,τ) is the set of point $x\in X$ such that, for all open set $U\in\tau$ containing x, $\mu(U)>0$.*

Throughout this work, we assume implicitly that all measures are Borel measures with respect to the standard topology of the space on which they are defined.

Lemma 6.2. *Assumption 6.5 is equivalent to the following statement: For all $E_0 \subseteq \mathbb{R}^{k\times m}$ such that $\mathbb{P}_{\boldsymbol{W}}(E_0) = 0$, there exists $\boldsymbol{W}^{(1)}, \dots, \boldsymbol{W}^{(m)} \in \mathcal{W} \setminus E_0$ and indices $i_1, ..., i_m \in [k]$ such that the row vectors $\boldsymbol{W}^{(1)}_{i_1,:}, \dots, \boldsymbol{W}^{(m)}_{i_m,:}$ are linearly independent.*

Proof First of all, the " $\Longleftarrow$ " direction is trivial since one can simply pick $E_0 = \emptyset$.

We now show the " $\Longrightarrow$ " direction. First of all, we notice that, since $\boldsymbol{W}^{(1)}_{i_1,:}, \dots, \boldsymbol{W}^{(m)}_{i_m,:}$ are linearly independent, they form a matrix with nonzero determinant, *i.e.,*

$$\det \begin{bmatrix} \boldsymbol{W}^{(1)}_{i_1,:} \\ \vdots \\ \boldsymbol{W}^{(m)}_{i_m,:} \end{bmatrix} \neq 0\,. \tag{6.26}$$

Define the map $\eta : (\mathbb{R}^{k\times m})^m \to \mathbb{R}^{m\times m}$ as

$$\eta(\bar{\boldsymbol{W}}^{(1)}, \dots, \bar{\boldsymbol{W}}^{(m)}) := \begin{bmatrix} \bar{\boldsymbol{W}}^{(1)}_{i_1,:} \\ \vdots \\ \bar{\boldsymbol{W}}^{(m)}_{i_m,:} \end{bmatrix},\ \forall (\bar{\boldsymbol{W}}^{(1)}, \dots, \bar{\boldsymbol{W}}^{(m)}) \in (\mathbb{R}^{k\times m})^m\,, \tag{6.27}$$

which is continuous. Note that $\det(\cdot)$ is also a continuous map, hence $\det \circ \eta$ is continuous as well. Thus, the set $V := (\det \circ \eta)^{-1}(\mathbb{R} \setminus \{0\})$ is open (since $\mathbb{R} \setminus \{0\}$ is open). Let $\mathbb{P}^m_{\boldsymbol{W}}$ be the product measure over tuples of matrices $(\bar{\boldsymbol{W}}^{(1)}, \dots, \bar{\boldsymbol{W}}^{(m)})$. Note that its support is $\mathcal{W}^m$. Because $(\boldsymbol{W}^{(1)}, ..., \boldsymbol{W}^{(m)})$ is in the open set V and in the support of $\mathbb{P}^m_{\boldsymbol{W}}$, we have that

$$\begin{aligned} 0 &< \mathbb{P}^m_{\boldsymbol{W}}(V) && (6.28)\\ &= \mathbb{P}^m_{\boldsymbol{W}}(V \cap \mathcal{W}^m) + \mathbb{P}^m_{\boldsymbol{W}}(V \cap (\mathcal{W}^m)^c) && (6.29)\\ &\leq \mathbb{P}^m_{\boldsymbol{W}}(V \cap \mathcal{W}^m) + \mathbb{P}^m_{\boldsymbol{W}}((\mathcal{W}^m)^c) && (6.30)\\ &= \mathbb{P}^m_{\boldsymbol{W}}(V \cap \mathcal{W}^m) && (6.31) \end{aligned}$$

Let $E_0 \subseteq \mathbb{R}^{k\times m}$ be such that $\mathbb{P}_{\boldsymbol{W}}(E_0) = 0$. Then, we also have that $\mathbb{P}^m_{\boldsymbol{W}}(E_0^m) = 0$ and thus

$$\mathbb{P}^m_{\boldsymbol{W}}((V \cap \mathcal{W}^m) \setminus E_0^m) > 0\,. \tag{6.32}$$

This implies that the set $((\det \circ \eta)^{-1}(\mathbb{R} \setminus \{0\}) \cap \mathcal{W}^m) \setminus E_0^m$ is not empty, *i.e.,* there exists $(\bar{\boldsymbol{W}}^{(1)}, \dots, \bar{\boldsymbol{W}}^{(m)}) \in \mathcal{W}^m \setminus E_0^m$ such that the rows $\bar{\boldsymbol{W}}^{(1)}_{i_1,:}, \dots, \bar{\boldsymbol{W}}^{(m)}_{i_m,:}$ are linearly independent. Since the measure zero set E_0 was arbitrary, this concludes the proof. ■

B.2. Proof of Theorem 6.1

This section presents the main results building up to Theorem 6.1.

For all $\boldsymbol{W} \in \mathcal{W}$, we are going to denote by $\hat{\boldsymbol{W}}^{(\boldsymbol{W})}$ some estimator of $\boldsymbol{W}$. The following result provides conditions under which if $\hat{\boldsymbol{W}}^{(\boldsymbol{W})}$ allows a perfect fit of the ground-truth distribution $p(y \mid \boldsymbol{x}, \boldsymbol{W})$, then the representation $\boldsymbol{f_\theta}$ and the parameter $\boldsymbol{W}$ are identified up to an invertible linear transformation. Many works have showed similar results in various context [Hyvarinen and Morioka, 2016, Khemakhem et al., 2020a, Roeder et al., 2021, Ahuja et al., 2022c]. We reuse some of their proof techniques.

Theorem 6.2 (Linear identifiability)**.** *Let $\hat{\boldsymbol{W}}^{(\cdot)} : \mathcal{W} \to \mathbb{R}^{k \times m}$. Suppose Assumptions 6.3, 6.4 and 6.5 hold and that, for $\mathbb{P}_{\boldsymbol{W}}$-almost every $\boldsymbol{W} \in \mathcal{W}$ and all $\boldsymbol{x} \in \mathcal{X}$, the following holds*

$$\mathrm{KL}(p(y; \hat{\boldsymbol{W}}^{(\boldsymbol{W})} \boldsymbol{f}_{\hat{\boldsymbol{\theta}}}(\boldsymbol{x})) || p(y; \boldsymbol{W} \boldsymbol{f}_{\boldsymbol{\theta}}(\boldsymbol{x})) = 0 \ . \tag{6.33}$$

Then, there exists an invertible matrix $\boldsymbol{L} \in \mathbb{R}^{m \times m}$ such that, for all $\boldsymbol{x} \in \mathcal{X}$, $\boldsymbol{f_\theta}(\boldsymbol{x}) = \boldsymbol{L} \boldsymbol{f}_{\hat{\boldsymbol{\theta}}}(\boldsymbol{x})$ and such that, for $\mathbb{P}_{\boldsymbol{W}}$-almost every $\boldsymbol{W} \in \mathcal{W}$, $\hat{\boldsymbol{W}}^{(\boldsymbol{W})} = \boldsymbol{W}\boldsymbol{L}$

Proof By Assumption 6.3, (6.33) implies that, for $\mathbb{P}_{\boldsymbol{W}}$-almost every $\boldsymbol{W}$ and all $\boldsymbol{x} \in \mathcal{X}$, $\boldsymbol{W} \boldsymbol{f_\theta}(\boldsymbol{x}) = \hat{\boldsymbol{W}}^{(\boldsymbol{W})} \boldsymbol{f}_{\hat{\boldsymbol{\theta}}}(\boldsymbol{x})$. Assumption 6.5 combined with Lemma 6.2 ensures that we can construct an invertible matrix $\boldsymbol{U} := \begin{bmatrix} \boldsymbol{W}_{i_1,:}^{(1)} \\ \vdots \\ \boldsymbol{W}_{i_{d_z},:}^{(d_z)} \end{bmatrix}$ such that $\boldsymbol{U} \boldsymbol{f_\theta}(\boldsymbol{x}) = \hat{\boldsymbol{U}} \boldsymbol{f}_{\hat{\boldsymbol{\theta}}}(\boldsymbol{x})$ for all $\boldsymbol{x} \in \mathcal{X}$ where $\hat{\boldsymbol{U}} := \begin{bmatrix} \hat{\boldsymbol{W}}_{i_1,:}^{(\boldsymbol{W}^{(1)})} \\ \vdots \\ \hat{\boldsymbol{W}}_{i_{d_z},:}^{(\boldsymbol{W}^{(d_z)})} \end{bmatrix}$. Left-multiplying by $\boldsymbol{U}^{-1}$ on both sides yields $\boldsymbol{f_\theta}(\boldsymbol{x}) = \boldsymbol{L} \boldsymbol{f}_{\hat{\boldsymbol{\theta}}}(\boldsymbol{x})$, where $\boldsymbol{L} := \boldsymbol{U}^{-1}\hat{\boldsymbol{U}}$. Using the invertible matrix $\boldsymbol{F}$ from Assumption 6.4, we can thus write $\boldsymbol{F} = \boldsymbol{L}\hat{\boldsymbol{F}}$ where we defined $\hat{\boldsymbol{F}} := [\boldsymbol{f}_{\hat{\boldsymbol{\theta}}}(\boldsymbol{x}^{(1)}), \cdots, \boldsymbol{f}_{\hat{\boldsymbol{\theta}}}(\boldsymbol{x}^{(d_z)})]$. Since $\boldsymbol{F}$ is invertible, so are $\boldsymbol{L}$ and $\hat{\boldsymbol{F}}$.

By substituting $\boldsymbol{F} = \boldsymbol{L}\hat{\boldsymbol{F}}$ in $\boldsymbol{W}\boldsymbol{F} = \hat{\boldsymbol{W}}^{(\boldsymbol{W})}\hat{\boldsymbol{F}}$, we obtain $\boldsymbol{W}\boldsymbol{L}\hat{\boldsymbol{F}} = \hat{\boldsymbol{W}}^{(\boldsymbol{W})}\hat{\boldsymbol{F}}$. By right-multiplying both sides by $\hat{\boldsymbol{F}}^{-1}$, we obtain $\boldsymbol{W}\boldsymbol{L} = \hat{\boldsymbol{W}}^{(\boldsymbol{W})}$. ■

The following theorem is where most of the theoretical contribution of this work lies. Note that Theorem 6.1, from the main text, is a straightforward application of this result.

Theorem 6.3. *(Disentanglement via task sparsity) Let $\hat{\boldsymbol{W}}^{(\cdot)} : \mathcal{W} \to \mathbb{R}^{k \times m}$. Suppose Assumptions 6.3, 6.4, 6.5, 6.6, 6.7 hold and that, for $\mathbb{P}_{\boldsymbol{W}}$-almost every $\boldsymbol{W} \in \mathcal{W}$ and all $\boldsymbol{x} \in \mathcal{X}$, the following holds*

$$\mathrm{KL}(p(y; \hat{\boldsymbol{W}}^{(\boldsymbol{W})} \boldsymbol{f}_{\hat{\boldsymbol{\theta}}}(\boldsymbol{x})) || p(y; \boldsymbol{W} \boldsymbol{f}_{\boldsymbol{\theta}}(\boldsymbol{x})) = 0 \ . \tag{6.34}$$

Moreover, assume that $\mathbb{E}\|\hat{\boldsymbol{W}}^{(\boldsymbol{W})}\|_{2,0} \leq \mathbb{E}\|\boldsymbol{W}\|_{2,0}$, where both expectations are taken w.r.t. $\mathbb{P}_{\boldsymbol{W}}$ and $\|\boldsymbol{W}\|_{2,0} := \sum_{j=1}^{m} \mathbb{1}(\boldsymbol{W}_{:j} \neq \boldsymbol{0})$ with $\mathbb{1}(\cdot)$ the indicator function. Then, $\boldsymbol{f}_{\hat{\boldsymbol{\theta}}}$ is disentangled w.r.t. $\boldsymbol{f_\theta}$ (Definition 6.1).

Proof First of all, by Assumptions 6.3, 6.4 and 6.5, we can apply Theorem 6.2 to conclude that $\boldsymbol{f_\theta}(\boldsymbol{x}) = \boldsymbol{L} \boldsymbol{f}_{\hat{\boldsymbol{\theta}}}(\boldsymbol{x})$ and $\boldsymbol{W}\boldsymbol{L} = \hat{\boldsymbol{W}}^{(\boldsymbol{W})}$ ($\mathbb{P}_{\boldsymbol{W}}$-almost everywhere) for some invertible matrix $\boldsymbol{L}$.

We can thus write $\mathbb{E}\|\boldsymbol{W}\boldsymbol{L}\|_{2,0} \leq \mathbb{E}\|\boldsymbol{W}\|_{2,0}$.

We can write

$$\mathbb{E}\|\boldsymbol{W}\|_{2,0} = \mathbb{E}_{p(S)}\mathbb{E}[\sum_{j=1}^{m} \mathbb{1}(\boldsymbol{W}_{:j} \neq \boldsymbol{0}) \mid S] \tag{6.35}$$

$$= \mathbb{E}_{p(S)} \sum_{j=1}^{m} \mathbb{E}[\mathbb{1}(\boldsymbol{W}_{:j} \neq \boldsymbol{0}) \mid S] \tag{6.36}$$

$$= \mathbb{E}_{p(S)} \sum_{j=1}^{m} \mathbb{P}_{\boldsymbol{W}|S}[\boldsymbol{W}_{:j} \neq \boldsymbol{0}] \tag{6.37}$$

$$= \mathbb{E}_{p(S)} \sum_{j=1}^{m} \mathbb{1}(j \in S)\,, \tag{6.38}$$

where the last step follows from the definition of S.

We now perform similar steps for $\mathbb{E}\|\boldsymbol{W}\boldsymbol{L}\|_{2,0}$:

$$\mathbb{E}\|\boldsymbol{W}L\|_{2,0} = \mathbb{E}_{p(S)}\mathbb{E}[\sum_{j=1}^{m} \mathbb{1}(\boldsymbol{W}\boldsymbol{L}_{:j} \neq \boldsymbol{0}) \mid S] \tag{6.39}$$

$$= \mathbb{E}_{p(S)} \sum_{j=1}^{m} \mathbb{E}[\mathbb{1}(\boldsymbol{W}\boldsymbol{L}_{:j} \neq \boldsymbol{0}) \mid S] \tag{6.40}$$

$$= \mathbb{E}_{p(S)} \sum_{j=1}^{m} \mathbb{P}_{\boldsymbol{W}|S}[\boldsymbol{W}\boldsymbol{L}_{:j} \neq \boldsymbol{0}] \tag{6.41}$$

$$= \mathbb{E}_{p(S)} \sum_{j=1}^{m} \mathbb{P}_{\boldsymbol{W}|S}[\boldsymbol{W}_{:S}\boldsymbol{L}_{S,j} \neq \boldsymbol{0}]\,. \tag{6.42}$$

Notice that

$$\mathbb{P}_{\boldsymbol{W}|S}[\boldsymbol{W}_{:S}\boldsymbol{L}_{S,j} \neq \boldsymbol{0}] = 1 - \mathbb{P}_{\boldsymbol{W}|S}[\boldsymbol{W}_{:S}\boldsymbol{L}_{S,j} = \boldsymbol{0}] \tag{6.43}$$

Let N_j be the support of $\boldsymbol{L}_{:j}$, *i.e.*, $N_j := \{i \in [m] \mid \boldsymbol{L}_{i,j} \neq 0\}$. When $S \cap N_j = \emptyset$, $\boldsymbol{L}_{S,j} = \boldsymbol{0}$ and thus $\mathbb{P}_{\boldsymbol{W}|S}[\boldsymbol{W}_{:S}\boldsymbol{L}_{S,j} = \boldsymbol{0}] = 1$. When $S \cap N_j \neq \emptyset$, $\boldsymbol{L}_{S,j} \neq \boldsymbol{0}$, by Assumption 6.6 we have that $\mathbb{P}_{\boldsymbol{W}|S}[\boldsymbol{W}_{:S}\boldsymbol{L}_{S,j} = \boldsymbol{0}] = 0$. Thus

$$\mathbb{P}_{\boldsymbol{W}|S}[\boldsymbol{W}_{:S}\boldsymbol{L}_{S,j} \neq \boldsymbol{0}] = 1 - \mathbb{1}(S \cap N_j = \emptyset) \tag{6.44}$$

$$= \mathbb{1}(S \cap N_j \neq \emptyset)\,, \tag{6.45}$$

which allows us to write

$$\mathbb{E}\|\boldsymbol{W}\boldsymbol{L}\|_{2,0} = \mathbb{E}_{p(S)} \sum_{j=1}^{m} \mathbb{1}(S \cap N_j \neq \emptyset)\,. \tag{6.46}$$

We thus have that

$$\mathbb{E}\|\boldsymbol{W}\boldsymbol{L}\|_{2,0} \leq \mathbb{E}\|\boldsymbol{W}\|_{2,0} \tag{6.47}$$

$$\mathbb{E}_{p(S)} \sum_{j=1}^{m} \mathbb{1}(S \cap N_j \neq \emptyset) \leq \mathbb{E}_{p(S)} \sum_{j=1}^{m} \mathbb{1}(j \in S)\,. \tag{6.48}$$

Since $\boldsymbol{L}$ is invertible, by Lemma 6.1, there exists a permutation $\sigma : [m] \to [m]$ such that, for all $j \in [m]$, $\boldsymbol{L}_{j,\sigma(j)} \neq 0$. In other words, for all $j \in [m]$, $j \in N_{\sigma(j)}$. Of course we can permute the terms of the l.h.s. of (6.48), which yields

$$\mathbb{E}_{p(S)} \sum_{j=1}^{m} \mathbb{1}(S \cap N_{\sigma(j)} \neq \emptyset) \leq \mathbb{E}_{p(S)} \sum_{j=1}^{m} \mathbb{1}(j \in S) \tag{6.49}$$

$$\mathbb{E}_{p(S)} \sum_{j=1}^{m} \big(\mathbb{1}(S \cap N_{\sigma(j)} \neq \emptyset) - \mathbb{1}(j \in S)\big) \leq 0\,. \tag{6.50}$$

We notice that each term $\mathbb{1}(S \cap N_{\sigma(j)} \neq \emptyset) - \mathbb{1}(j \in S) \geq 0$ since whenever $j \in S$, we also have that $j \in S \cap N_{\sigma(j)}$ (recall $j \in N_{\sigma(j)}$). Thus, the l.h.s. of (6.50) is a sum of non-negative terms which is itself non-positive. This means that every term in the sum is zero:

$$\forall S \in \mathcal{S},\ \forall j \in [m],\ \mathbb{1}(S \cap N_{\sigma(j)} \neq \emptyset) = \mathbb{1}(j \in S)\,. \tag{6.51}$$

Importantly,

$$\forall j \in [m],\ \forall S \in \mathcal{S},\ j \notin S \implies S \cap N_{\sigma(j)} = \emptyset\,, \tag{6.52}$$

and since $S \cap N_{\sigma(j)} = \emptyset \iff N_{\sigma(j)} \subseteq S^c$ we have that

$$\forall j \in [m],\ \forall S \in \mathcal{S},\ j \notin S \implies N_{\sigma(j)} \subseteq S^c \tag{6.53}$$

$$\forall j \in [m],\ N_{\sigma(j)} \subseteq \bigcap_{S \in \mathcal{S} | j \notin S} S^c\,. \tag{6.54}$$

By Assumption 6.7, we have that $\bigcup_{S \in \mathcal{S}|j \notin S} S = [m] \setminus \{j\}$. By taking the complement on both sides and using De Morgan's law, we get $\bigcap_{S \in \mathcal{S}|j \notin S} S^c = \{j\}$, which implies that $N_{\sigma(j)} = \{j\}$ by (6.54). Thus, $\boldsymbol{L} = \boldsymbol{D}\boldsymbol{P}$ where $\boldsymbol{D}$ is an invertible diagonal matrix and $\boldsymbol{P}$ is a permutation matrix. ■

Before presenting Theorem 6.1 from the main text, we first present a variation of it where we constrain $\mathbb{E}\|\hat{\boldsymbol{W}}^{(\boldsymbol{W})}\|_{2,0}$ to be smaller than $\mathbb{E}\|\boldsymbol{W}\|_{2,0}$. We note that this is weaker than imposing $\|\hat{\boldsymbol{W}}^{(\boldsymbol{W})}\|_{2,0} \leq \|\boldsymbol{W}\|_{2,0}$ for all $\boldsymbol{W} \in \mathcal{W}$, as is the case in Problem (6.4) of Theorem 6.1. Note that Appendix B.3 presents a natural relaxation of Problem (6.55) which we experiment with in Appendix D.2.5.

Theorem 6.4 (Sparse multitask learning for disentanglement)**.** *Let $\hat{\boldsymbol{\theta}}$ be a minimizer of*

$$\begin{aligned} &\min_{\hat{\boldsymbol{\theta}}} \mathbb{E}_{\mathbb{P}_{\boldsymbol{W}}} \mathbb{E}_{p(\boldsymbol{x},y|\boldsymbol{W})} - \log p(y; \hat{\boldsymbol{W}}^{(\boldsymbol{W})} \boldsymbol{f}_{\hat{\boldsymbol{\theta}}}(\boldsymbol{x})) \\ &\text{s.t.} \quad \forall\, \boldsymbol{W} \in \mathcal{W},\ \hat{\boldsymbol{W}}^{(\boldsymbol{W})} \in \arg\min_{\tilde{\boldsymbol{W}}} \mathbb{E}_{p(\boldsymbol{x},y|\boldsymbol{W})} - \log p(y; \tilde{\boldsymbol{W}} \boldsymbol{f}_{\hat{\boldsymbol{\theta}}}(\boldsymbol{x})) \\ &\qquad \mathbb{E}_{\mathbb{P}_{\boldsymbol{W}}} \|\hat{\boldsymbol{W}}^{(\boldsymbol{W})}\|_{2,0} \leq \mathbb{E}_{\mathbb{P}_{\boldsymbol{W}}} \|\boldsymbol{W}\|_{2,0} \ , \end{aligned} \tag{6.55}$$

where $\mathbb{P}_{\boldsymbol{W}}$ and $p(\boldsymbol{x}, y \mid \boldsymbol{W})$ are described in Section 6.3.1. Under Assumptions 6.3, 6.4, 6.5, 6.6, 6.7 and if $\boldsymbol{f}_{\tilde{\boldsymbol{\theta}}}$ is continuous for all $\tilde{\boldsymbol{\theta}}$, $\boldsymbol{f}_{\hat{\boldsymbol{\theta}}}$ is disentangled w.r.t. $\boldsymbol{f}_{\boldsymbol{\theta}}$ (Definition 6.1).

Proof First, notice that

$$0 \leq \mathbb{E}_{\mathbb{P}_{\boldsymbol{W}}} \mathbb{E}_{p(\boldsymbol{x}|\boldsymbol{W})} \mathrm{KL}(p(y; \boldsymbol{W} \boldsymbol{f}_{\boldsymbol{\theta}}(\boldsymbol{x})) \,||\, p(y; \hat{\boldsymbol{W}}^{(\boldsymbol{W})} \boldsymbol{f}_{\hat{\boldsymbol{\theta}}}(\boldsymbol{x}))) \tag{6.56}$$

$$\mathbb{E}_{\mathbb{P}_{\boldsymbol{W}}} \mathbb{E}_{p(\boldsymbol{x},y|\boldsymbol{W})} - \log p(y; \boldsymbol{W} \boldsymbol{f}_{\boldsymbol{\theta}}(\boldsymbol{x})) \leq \mathbb{E}_{\mathbb{P}_{\boldsymbol{W}}} \mathbb{E}_{p(\boldsymbol{x},y|\boldsymbol{W})} - \log p(y; \hat{\boldsymbol{W}}^{(\boldsymbol{W})} \boldsymbol{f}_{\hat{\boldsymbol{\theta}}}(\boldsymbol{x})) \,. \tag{6.57}$$

This means the objective is minimized (without constraint) if and only if

$$\mathbb{E}_{p(\boldsymbol{x}|\boldsymbol{W})} \mathrm{KL}(p(y; \boldsymbol{W} \boldsymbol{f}_{\boldsymbol{\theta}}(\boldsymbol{x})) \,||\, p(y; \hat{\boldsymbol{W}}^{(\boldsymbol{W})} \boldsymbol{f}_{\hat{\boldsymbol{\theta}}}(\boldsymbol{x}))) = 0 \tag{6.58}$$

$\mathbb{P}_{\boldsymbol{W}}$-almost everywhere. For a fixed $\boldsymbol{W}$, this equality holds if and only if the KL equals zero $p(\boldsymbol{x} \mid \boldsymbol{W})$-almost everywhere, which, by Assumption 6.3, is true if and only if $\boldsymbol{W} \boldsymbol{f}_{\boldsymbol{\theta}}(\boldsymbol{x}) = \hat{\boldsymbol{W}}^{(\boldsymbol{W})} \boldsymbol{f}_{\hat{\boldsymbol{\theta}}}(\boldsymbol{x})$ $p(\boldsymbol{x} \mid \boldsymbol{W})$-almost everywhere. Since both $\boldsymbol{W} \boldsymbol{f}_{\boldsymbol{\theta}}(\boldsymbol{x})$ and $\hat{\boldsymbol{W}}^{(\boldsymbol{W})} \boldsymbol{f}_{\hat{\boldsymbol{\theta}}}(\boldsymbol{x})$ are continuous functions of $\boldsymbol{x}$, the equality holds over $\mathcal{X}$ (the support of $p(\boldsymbol{x} \mid \boldsymbol{W})$).

This unconstrained global minimum can actually be achieved by respecting the constraints of Problem (6.55) simply by setting $\hat{\boldsymbol{\theta}} := \boldsymbol{\theta}$ and $\hat{\boldsymbol{W}}^{(\boldsymbol{W})} := \boldsymbol{W}$. Indeed, the first constraint is satisfied because, for all $\tilde{\boldsymbol{W}}$,

$$0 \leq \mathbb{E}_{p(\boldsymbol{x}|\boldsymbol{W})} \mathrm{KL}(p(y; \boldsymbol{W} \boldsymbol{f}_{\boldsymbol{\theta}}(\boldsymbol{x})) \,||\, p(y; \tilde{\boldsymbol{W}} \boldsymbol{f}_{\boldsymbol{\theta}}(\boldsymbol{x}))) \tag{6.59}$$

$$\mathbb{E}_{p(\boldsymbol{x},y|\boldsymbol{W})} - \log p(y; \boldsymbol{W} \boldsymbol{f}_{\boldsymbol{\theta}}(\boldsymbol{x})) \leq \mathbb{E}_{p(\boldsymbol{x},y|\boldsymbol{W})} - \log p(y; \tilde{\boldsymbol{W}} \boldsymbol{f}_{\boldsymbol{\theta}}(\boldsymbol{x})) \,, \tag{6.60}$$

and clearly the lower bound is attained when $\tilde{\boldsymbol{W}} := \boldsymbol{W}$. The second constraint is trivially satisfied, since $\mathbb{E}_{\mathbb{P}_{\boldsymbol{W}}} \|\hat{\boldsymbol{W}}^{(\boldsymbol{W})}\|_{2,0} = \mathbb{E}_{\mathbb{P}_{\boldsymbol{W}}} \|\boldsymbol{W}\|_{2,0}$.

The above implies that if $\hat{\boldsymbol{\theta}}$ is some minimizer of Problem (6.55), we must have that, (i) for $\mathbb{P}_{\boldsymbol{W}}$-almost every $\boldsymbol{W}$, $\boldsymbol{W} \boldsymbol{f}_{\boldsymbol{\theta}}(\boldsymbol{x}) = \hat{\boldsymbol{W}}^{(\boldsymbol{W})} \boldsymbol{f}_{\hat{\boldsymbol{\theta}}}(\boldsymbol{x})$ for all $\boldsymbol{x} \in \mathcal{X}$, (ii) $\mathbb{E}_{\mathbb{P}_{\boldsymbol{W}}} ||\hat{\boldsymbol{W}}^{(\boldsymbol{W})}||_0 \leq \mathbb{E}_{\mathbb{P}_{\boldsymbol{W}}} ||\boldsymbol{W}||_0$. Thus, Theorem 6.3 implies the desired conclusion. ■

Based on Theorem 6.4, we can slightly adjust the argument to prove Theorem 6.1 from the main text.

Theorem 6.1 (Sparse multi-task learning for disentanglement)**.** *Let $\hat{\boldsymbol{\theta}}$ be a minimizer of*

$$\begin{aligned} &\min_{\hat{\boldsymbol{\theta}}} \mathbb{E}_{\mathbb{P}_{\boldsymbol{W}}} \mathbb{E}_{p(\boldsymbol{x},y|\boldsymbol{W})} - \log p(y; \hat{\boldsymbol{W}}^{(\boldsymbol{W})} \boldsymbol{f}_{\hat{\boldsymbol{\theta}}}(\boldsymbol{x})) \\ &\text{s.t.} \quad \hat{\boldsymbol{W}}^{(\boldsymbol{W})} \in \underset{\substack{\tilde{\boldsymbol{W}} \text{ s.t.} \\ ||\tilde{\boldsymbol{W}}||_{2,0} \leq ||\boldsymbol{W}||_{2,0}}}{\arg\min} \mathbb{E}_{p(\boldsymbol{x},y|\boldsymbol{W})} - \log p(y; \tilde{\boldsymbol{W}} \boldsymbol{f}_{\hat{\boldsymbol{\theta}}}(\boldsymbol{x})) \ , \end{aligned} \tag{6.4}$$

where the constraint holds for all $\boldsymbol{W} \in \mathcal{W}$ and where $\mathbb{P}_{\boldsymbol{W}}$ and $p(\boldsymbol{x}, y \mid \boldsymbol{W})$ are described in Section 6.3.1. Under Assumptions 6.3, 6.4, 6.5, 6.6, 6.7 and if $\boldsymbol{f}_{\tilde{\boldsymbol{\theta}}}$ is continuous for all $\tilde{\boldsymbol{\theta}}$, $\boldsymbol{f}_{\hat{\boldsymbol{\theta}}}$ is disentangled w.r.t. $\boldsymbol{f}_{\boldsymbol{\theta}}$ (Definition 6.1).

Proof The first part of the argument in the proof of Theorem 6.4 applies here as well, meaning: unconstrained minimization of the objective holds if and only if, for $\mathbb{P}_{\boldsymbol{W}}$-almost every $\boldsymbol{W}$ and all $\boldsymbol{x} \in \mathcal{X}$, $\boldsymbol{W} \boldsymbol{f}_{\boldsymbol{\theta}}(\boldsymbol{x}) = \hat{\boldsymbol{W}}^{(\boldsymbol{W})} \boldsymbol{f}_{\hat{\boldsymbol{\theta}}}(\boldsymbol{x})$. And again, this unconstrained minimum can be achieved by respecting the constraint of Problem (6.4) simply by setting $\hat{\boldsymbol{\theta}} := \boldsymbol{\theta}$ and $\hat{\boldsymbol{W}}^{(\boldsymbol{W})} := \boldsymbol{W}$.

This means that if $\hat{\boldsymbol{\theta}}$ is some minimizer of Problem (6.4), we must have (i) for $\mathbb{P}_{\boldsymbol{W}}$-almost every $\boldsymbol{W}$, $\boldsymbol{W} \boldsymbol{f}_{\boldsymbol{\theta}}(\boldsymbol{x}) = \hat{\boldsymbol{W}}^{(\boldsymbol{W})} \boldsymbol{f}_{\hat{\boldsymbol{\theta}}}(\boldsymbol{x})$ for all $\boldsymbol{x} \in \mathcal{X}$ and (ii) for all $\boldsymbol{W} \in \mathcal{W}$, $\|\hat{\boldsymbol{W}}^{(\boldsymbol{W})}\|_{2,0} \leq \|\boldsymbol{W}\|_{2,0}$. Of course the latter point implies $\mathbb{E}_{\mathbb{P}_{\boldsymbol{W}}} ||\hat{\boldsymbol{W}}^{(\boldsymbol{W})}||_{2,0} \leq \mathbb{E}_{\mathbb{P}_{\boldsymbol{W}}} ||\boldsymbol{W}||_{2,0}$, which allows us to apply Theorem 6.3 to obtain the desired conclusion. ■

B.3. Regularization in the outer problem instead of in the inner problem

Theorem 6.4 presented an alternative bilevel optimization problem to the one of Theorem 6.1 in the main text. Essentially, the difference is that the constraints $\|\hat{\boldsymbol{W}}^{(\boldsymbol{W})}\|_{2,0} \leq \|\boldsymbol{W}\|_{2,0}$ for all $\boldsymbol{W} \in \mathcal{W}$ are replaced by the unique constraint $\mathbb{E}\|\hat{\boldsymbol{W}}^{(\boldsymbol{W})}\|_{2,0} \leq \mathbb{E}\|\boldsymbol{W}\|_{2,0}$, which is a weaker constraint.

In Section 6.3.4, we introduced a tractable relaxation of the problem of Theorem 6.1. In this section, we introduce a relaxation of the problem of Theorem 6.4.

A natural idea is to replace the constraint $\mathbb{E}\|\hat{\boldsymbol{W}}^{(\boldsymbol{W})}\|_{2,0} \leq \mathbb{E}\|\boldsymbol{W}\|_{2,0}$ of Theorem 6.4 by a penalty $\lambda \mathbb{E}\|\hat{\boldsymbol{W}}^{(\boldsymbol{W})}\|_{2,1}$ in the outer problem, like so:

$$\begin{aligned} &\min_{\hat{\boldsymbol{\theta}}} \mathbb{E}_{\mathbb{P}_{\boldsymbol{W}}} \mathbb{E}_{p(\boldsymbol{x},y|\boldsymbol{W})} - \log p(y; \hat{\boldsymbol{W}}^{(\boldsymbol{W})} \boldsymbol{f}_{\hat{\boldsymbol{\theta}}}(\boldsymbol{x})) + \lambda \mathbb{E}_{\mathbb{P}_{\boldsymbol{W}}} \|\hat{\boldsymbol{W}}^{(\boldsymbol{W})}\|_{2,1} \\ &\text{s.t.} \quad \forall\, \boldsymbol{W} \in \mathcal{W},\ \hat{\boldsymbol{W}}^{(\boldsymbol{W})} \in \underset{\tilde{\boldsymbol{W}}}{\arg\min}\, \mathbb{E}_{p(\boldsymbol{x},y|\boldsymbol{W})} - \log p(y; \tilde{\boldsymbol{W}} \boldsymbol{f}_{\hat{\boldsymbol{\theta}}}(\boldsymbol{x})) \,, \end{aligned} \tag{6.61}$$

in which we can replace the expectations by empirical averages to get

$$\min_{\hat{\boldsymbol{\theta}}} \frac{1}{T}\sum_{t=1}^{T}\left[-\frac{1}{n}\sum_{(\boldsymbol{x},y)\in\mathcal{D}_t}\log p(y;\hat{\boldsymbol{W}}^{(t)}\boldsymbol{f}_{\hat{\boldsymbol{\theta}}}(\boldsymbol{x}))+\lambda\|\hat{\boldsymbol{W}}^{(t)}\|_{2,1}\right] \tag{6.62}$$
$$\text{s.t. } \hat{\boldsymbol{W}}^{(t)}\in\arg\min_{\tilde{\boldsymbol{W}}}\frac{1}{n}\sum_{(\boldsymbol{x},y)\in\mathcal{D}_t}-\log p(y;\tilde{\boldsymbol{W}}\boldsymbol{f}_{\hat{\boldsymbol{\theta}}}(\boldsymbol{x}))\,.$$

This can be optimized in the same way as Problem (6.6) via implicit differentiation and standard gradient descent algorithms. The essential difference between Problem (6.62) and Problem (6.6) is that the former has regularization in the outer problem instead of in the inner problem. From a practical point of view, this problem is typically simpler than Problem (6.6) since the inner objective is generally smooth, and standard implicit differentiation techniques apply (the non-smooth term $\|\tilde{\boldsymbol{W}}\|_{2,1}$ in the inner objective of Problem (6.6) requiring some care with implicit differentiation; Bertrand et al., 2022). We provide some experimental results in Appendix D.2.5 demonstrating that this alternative works as well.

B.4. What can go wrong when Assumption 6.6 is violated?

Theorem 6.2 allowed us to conclude that $\hat{\boldsymbol{W}}^{(\boldsymbol{W})}=\boldsymbol{W}\boldsymbol{L}$ for $\mathbb{P}_{\boldsymbol{W}}$-almost every $\boldsymbol{W}$ and that $\boldsymbol{L}\boldsymbol{f}_{\hat{\boldsymbol{\theta}}}(\boldsymbol{x})=\boldsymbol{f}_{\boldsymbol{\theta}}(\boldsymbol{x})$ for all $\boldsymbol{x}\in\mathcal{X}$. The rest of the argument leading up to Theorem 6.1 essentially amounts to showing that having $\|\hat{\boldsymbol{W}}^{(\boldsymbol{W})}\|_{2,0}\leq\|\boldsymbol{W}\|_{2,0}$ for all $\boldsymbol{W}\in\mathcal{W}$ forces $\boldsymbol{L}$ to be a permutation-scaling matrix. The intuition is that $\|\boldsymbol{W}\boldsymbol{L}\|_{2,0}\leq\|\boldsymbol{W}\|_{2,0}$ everywhere should force $\boldsymbol{L}$ to be sparse, and maximal sparsity is precisely when $\boldsymbol{L}$ is a permutation-scaling matrix. But just how many $\boldsymbol{W}$ do we need and how diverse should they be to make this argument formal? Our answer is given by Assumption 6.6. But what can go wrong when this assumption is not satisfied? To answer this question, we construct a counterexample in which the distribution $\mathbb{P}_{\boldsymbol{W}}$ satisfies Assumption 6.7 but not Assumption 6.6 and a matrix $\boldsymbol{L}$ that satisfies the constraint $\|\boldsymbol{W}\boldsymbol{L}\|_{2,0}\leq\|\boldsymbol{W}\|_{2,0}$ everywhere but that is not a permutation-scaling matrix. Consider a distribution $\mathbb{P}_{\boldsymbol{W}}$ with support $\mathcal{W}:=\{[1,1,0],[1,0,1],[0,1,1]\}$ (which is finite) and let

$$\boldsymbol{L}:=\begin{bmatrix}3&-1&-1\\-1&1&3\\1&3&1\end{bmatrix}, \tag{6.63}$$

which, of course, is not a permutation-scaling matrix. One can then compute to show that the sparsity constraint holds for all $\boldsymbol{W} \in \mathcal{W}$:

$$\|[1\ 1\ 0]\boldsymbol{L}\|_{2,0} = \|[2\ 0\ 2]\|_{2,0} \leq 2 = \|[1\ 1\ 0]\|_{2,0} \tag{6.64}$$

$$\|[1\ 0\ 1]\boldsymbol{L}\|_{2,0} = \|[4\ 2\ 0]\|_{2,0} \leq 2 = \|[1\ 0\ 1]\|_{2,0} \tag{6.65}$$

$$\|[0\ 1\ 1]\boldsymbol{L}\|_{2,0} = \|[0\ 4\ 4]\|_{2,0} \leq 2 = \|[0\ 1\ 1]\|_{2,0}\,. \tag{6.66}$$

This means that, with such a $\mathbb{P}_{\boldsymbol{W}}$, solving the bilevel problem of Theorem 6.1 will not necessarily lead to a disentangled representation since one could fall on a "bad" $\boldsymbol{L}$ such as the one defined above.

B.5. Assumption 6.7 holds with high probability when $|\mathcal{S}|$ large

In this section, we provide a probabilistic argument showing that Assumption 6.7 holds with high probability when the number of supports is large. Let $\mathcal{S}^{(T)} := \{S^{(1)}, S^{(2)}, \ldots, S^{(T)}\}$ be the set of supports observed, where T is the number of supports. To make this argument, we will assume that the $S^{(t)}$ are sampled independently and identically. Moreover, $\mathbb{P}[i \in S^{(t)}] = p \in (0,1)$ and these events are assumed independent.

The next proposition shows that the probability that Assumption 6.7 fails under the above model is very small when T is large.

Proposition 6.3. *Given the probabilistic model described above, we have*

$$\mathbb{P}\left[\exists j \in [m] \text{ s.t. } \bigcup_{S \in \mathcal{S}^{(T)} | j \notin S} S \neq [m] \setminus \{j\}\right] \leq m(m-1)(1-p(1-p))^T \xrightarrow{T \to \infty} 0\,. \tag{6.67}$$

Proof By rewriting slightly the original probablity statement and applying the union bound, we get

$$\mathbb{P}\left[\exists j \in [m] \text{ s.t. } \bigcup_{S \in \mathcal{S}^{(T)} | j \notin S} S \neq [m] \setminus \{j\}\right] \tag{6.68}$$

$$= \mathbb{P}\left[\exists j \in [m], i \in [m] \setminus \{j\} \text{ s.t. } i \notin \bigcup_{S \in \mathcal{S}^{(T)} | j \notin S} S\right] \tag{6.69}$$

$$\leq \sum_{j=1}^{m} \sum_{i \in [m] \setminus \{j\}} \mathbb{P}\left[i \notin \bigcup_{S \in \mathcal{S}^{(T)} | j \notin S} S\right], \tag{6.70}$$

We can further write

$$\mathbb{P}\left[i \notin \bigcup_{S \in \mathcal{S}^{(T)} | j \notin S} S\right] = \mathbb{P}\left[\forall t \in [T], j \notin S^{(t)} \implies i \notin S^{(t)}\right] \tag{6.71}$$

$$= \mathbb{P}\left[\forall t \in [T], j \in S^{(t)} \vee i \notin S^{(t)}\right] \tag{6.72}$$

$$= \prod_{t=1}^{T} \mathbb{P}\left[j \in S^{(t)} \vee i \notin S^{(t)}\right] , \tag{6.73}$$

where the last step holds because the supports $S^{(t)}$ are mutually independent. We continue and get

$$\mathbb{P}\left[i \notin \bigcup_{S \in \mathcal{S}^{(T)} | j \notin S} S\right] = \prod_{t=1}^{T} \mathbb{P}\left[j \in S^{(t)} \vee i \notin S^{(t)}\right] \tag{6.74}$$

$$= \prod_{t=1}^{T} (1 - \mathbb{P}\left[j \notin S^{(t)} \wedge i \in S^{(t)}\right]) \tag{6.75}$$

$$= \prod_{t=1}^{T} (1 - \mathbb{P}\left[j \notin S^{(t)}\right] \mathbb{P}\left[i \in S^{(t)}\right]) \tag{6.76}$$

$$= \prod_{t=1}^{T} (1 - (1-p)p) , \tag{6.77}$$

where we used the fact that the events $j \notin S^{(t)}$ and $i \in S^{(t)}$ are independent (when $i \neq j$). Bringing everything together, one gets

$$\mathbb{P}\left[\exists j \in [m] \text{ s.t. } \bigcup_{S \in \mathcal{S}^{(T)} | j \notin S} S \neq [m] \setminus \{j\}\right] \leq \sum_{j=1}^{m} \sum_{i \in [m] \setminus \{j\}} \prod_{t=1}^{T} (1 - (1-p)p) \tag{6.78}$$

$$= m(m-1)(1-(1-p)p)^T \tag{6.79}$$

$$\tag{6.80}$$

which converges to 0 when $T \to \infty$ since $0 < 1 - (1-p)p < 1$. ■

B.6. A distribution without density satisfying Assumption 6.6

Interestingly, there are distributions over $\boldsymbol{W}_{1,S} \mid S$ that do not have a density w.r.t. the Lebesgue measure, but still satisfy Assumption 6.6. This is the case, e.g., when $\boldsymbol{W}_{1,S} \mid S$ puts uniform mass over a $(|S|-1)$-dimensional sphere embedded in $\mathbb{R}^{|S|}$ and centered at zero. In that case, for all $\boldsymbol{a} \in \mathbb{R}^{|S|} \setminus \{0\}$, the intersection of $\text{span}\{\boldsymbol{a}\}^\perp$, which is $(|S|-1)$-dimensional, with the $(|S|-1)$-dimensional sphere is $(|S|-2)$-dimensional and thus has probability zero of occurring. One can

certainly construct more exotic examples of measures satisfying Assumption 6.6 that concentrate mass on lower dimensional manifolds.

C. Optimization details

C.1. Group Lasso SVM Dual

Notation. The Fenchel conjugate of a function $h : \mathbb{R}^d \to \mathbb{R}$ is written h^* and is defined for any $y \in \mathbb{R}^d$, by $h^*(y) = \sup_{x \in \mathbb{R}^d} \langle x, y \rangle - h(x)$.

Definition 6.3. (Primal Group Lasso Soft-Margin Multiclass SVM.) *The primal problem of the group Lasso soft-margin multiclass SVM is defined as*

$$\min_{\boldsymbol{W} \in \mathbb{R}^{k \times m}} \mathcal{L}_{\text{in}}(\boldsymbol{W}; \boldsymbol{F}, \boldsymbol{Y}) := \sum_{i=1}^{n} \max_{l \in [k]} (1 + (\boldsymbol{W}_{\boldsymbol{y}_i:} - \boldsymbol{W}_{l:}) \boldsymbol{F}_{i:} - \boldsymbol{Y}_{il}) + \lambda_1 \|\boldsymbol{W}\|_{2,1} + \tfrac{\lambda_2}{2} \|\boldsymbol{W}\|^2 \tag{6.81}$$

Proposition 6.4. (Dual Group Lasso Soft-Margin Multiclass SVM.) *The dual of the inner problem with $\mathcal{L}_{\text{in}}$ as defined in (6.8) writes*

$$\begin{aligned} &\min_{\boldsymbol{\Lambda} \in \mathbb{R}^{n \times k}} \frac{1}{\lambda_2} \sum_{j=1}^{m} \|\text{BST}\left((\boldsymbol{Y} - \boldsymbol{\Lambda})^\top \boldsymbol{F}_{:j}, \lambda_1\right)\|^2 + \langle \boldsymbol{Y}, \boldsymbol{\Lambda} \rangle \\ &\text{s.t. } \forall i, l, \in [n] \times [k], \ \sum_{l'=1}^{k} \boldsymbol{\Lambda}_{il'} = 1 \ \textit{ and } \ \boldsymbol{\Lambda}_{il} \geq 0 \ , \end{aligned} \tag{6.10}$$

with $\text{BST} : (\boldsymbol{a}, \tau) \mapsto (1 - \tau / \|\boldsymbol{a}\|)_+ \boldsymbol{a}$ *is the block soft-thresholding operator,* $\boldsymbol{F} \in \mathbb{R}^{n \times m}$ *the concatenation of* $\{\boldsymbol{f}_{\hat{\boldsymbol{\theta}}}(x)\}_{(x,y) \in \mathcal{D}^{\text{train}}}$. *In addition, the primal-dual link writes,* $\forall j \in [m]$, $\boldsymbol{W}_{:j} = \text{BST}\left((\boldsymbol{Y} - \boldsymbol{\Lambda})^\top \boldsymbol{F}_{:j}, \lambda_1\right) / \lambda_2$.

The primal objective 6.81 can be hard to minimize with modern solvers. Moreover in few-shot learning applications, the number of features m is usually much larger than the number of samples n (in Lee et al. 2019, $m = 1.6 \cdot 10^4$ and $n \leq 25$), hence we solve the dual of Problem (6.81).

Proof [Proof of Proposition 6.4] Let $g : \boldsymbol{u} \mapsto \lambda_1 \|\boldsymbol{u}\| + \frac{\lambda_2}{2} \|\boldsymbol{u}\|^2$. Proof of Proposition 6.4 is composed of the following lemmas.

Lemma 6.3. i) *The dual of Problem (6.81) is*

$$\begin{aligned} &\min_{\boldsymbol{\Lambda} \in \mathbb{R}^{n \times k}} \sum_{j=1}^{m} g^*((\boldsymbol{Y} - \boldsymbol{\Lambda})^\top \boldsymbol{F}_{:j}) + \langle \boldsymbol{Y}, \boldsymbol{\Lambda} \rangle \\ &\text{s.t.} \quad \forall i \in [n], \sum_{l=1}^{k} \boldsymbol{\Lambda}_{il} = 1 \ , \quad \forall i \in [n], l \in [k], \ \boldsymbol{\Lambda}_{il} \geq 0 \ , \end{aligned} \tag{6.82}$$

where g^* is the Fenchel conjugate of the function g.

ii) *The Fenchel conjugate of the function g writes*

$$\forall \boldsymbol{v} \in \mathbb{R}^K,\ g^*(\boldsymbol{v}) = \frac{1}{\lambda_2}\|\mathrm{BST}(\boldsymbol{v}, \lambda_1)\|^2 \ . \tag{6.83}$$

Lemmas 6.4 *i)* & 6.4 *ii)* yields Proposition 6.4.

Proof [Proof of Lemma 6.4 *i)*.] The Lagrangian of Problem (6.81) writes:

$$\mathcal{L}(\boldsymbol{W}, \boldsymbol{\xi}, \boldsymbol{\Lambda}) = \sum_{j=1}^{m} g(\boldsymbol{W}_{:j}) + \sum_i \boldsymbol{\xi}_i + \sum_{i=1}^{n}\sum_{l=1}^{k}(1 - \boldsymbol{\xi}_i - \boldsymbol{W}_{\boldsymbol{y}_i:} \cdot \boldsymbol{F}_{i:} + \boldsymbol{W}_{l:} \cdot \boldsymbol{F}_{i:} - \boldsymbol{Y}_{il})\boldsymbol{\Lambda}_{il} \ . \tag{6.84}$$

$\partial_{\boldsymbol{\xi}}\mathcal{L}(\boldsymbol{W}, \boldsymbol{\xi}, \boldsymbol{\Lambda}) = 0$ yields $\forall i \in [n], \sum_{l=1}^{k} \boldsymbol{\Lambda}_{il} = 1$. Then the Lagrangian rewrites

$$\begin{aligned}
\min_{\boldsymbol{W}}\min_{\boldsymbol{\xi}} \mathcal{L}(\boldsymbol{W}, \boldsymbol{\xi}, \boldsymbol{\Lambda}) &= \min_{\boldsymbol{W}, \boldsymbol{\xi}} \sum_{j=1}^{m} g(\boldsymbol{W}_{:j}) + \sum_{i=1}^{n} \boldsymbol{\xi}_i + \sum_{i=1}^{n}\sum_{l=1}^{k}(-\boldsymbol{\xi}_i - \boldsymbol{W}_{\boldsymbol{y}_i:} \cdot \boldsymbol{F}_{i:} + \boldsymbol{W}_{l:} \cdot \boldsymbol{F}_{i:} - \boldsymbol{Y}_{il})\boldsymbol{\Lambda}_{il} \\
&= \sum_{j=1}^{m} \underbrace{\min_{\boldsymbol{W}_{:j}} g(\boldsymbol{W}_{:j}) - \underbrace{\sum_{i=1}^{n}\sum_{l=1}^{k}(\boldsymbol{F}_{i:}\boldsymbol{Y}_{il} - \boldsymbol{F}_{i:}\boldsymbol{\Lambda}_{il})\boldsymbol{W}_{l:}}_{=\langle(\boldsymbol{Y}-\boldsymbol{\Lambda})^\top \boldsymbol{F}_{:j}, \boldsymbol{W}_{:j}\rangle}}_{=-g^*((\boldsymbol{Y}-\boldsymbol{\Lambda})^\top \boldsymbol{F}_{:j})} - \sum_{i=1}^{n}\sum_{l=1}^{k} \boldsymbol{Y}_{il}\boldsymbol{\Lambda}_{il} \ .
\end{aligned}$$

Then the dual problem writes:

$$\min_{\boldsymbol{\Lambda} \in \mathbb{R}^{n \times k}} \sum_{j=1}^{m} g^*\left((\boldsymbol{Y} - \boldsymbol{\Lambda})^\top \boldsymbol{F}_{:j}\right) + \langle \boldsymbol{Y}, \boldsymbol{\Lambda} \rangle \tag{6.85}$$

$$\text{s. t.} \quad \forall i \in [n] \quad \sum_{l=1}^{k} \boldsymbol{\Lambda}_{il} = 1 \ , \forall i \in [n], l \in [k], \quad \boldsymbol{\Lambda}_{il} \geq 0 \ . \tag{6.86}$$

■

Proof [Proof of Lemma 6.4 *ii)*] Let $h : \boldsymbol{u} \mapsto \|\boldsymbol{u}\|_2 + \frac{\kappa}{2}\|\boldsymbol{u}\|^2$. The proof of Lemma 6.4 *i)* is done using the following steps.

Lemma 6.4. i) $h^*(\boldsymbol{v}) = \frac{1}{2\kappa}\|\boldsymbol{v}\|_2^2 - \left(\frac{\kappa}{2}\|\cdot\|_2^2 \square \|\cdot\|_2\right)(\boldsymbol{v}/\kappa)$.

ii) $\left(\frac{\kappa}{2}\|\cdot\|_2^2 \square \|\cdot\|_2\right)(\boldsymbol{v}) = \frac{\kappa}{2}\|\boldsymbol{v}\|_2^2 - \frac{1}{2\kappa}\|\mathrm{BST}(\kappa\boldsymbol{v}, 1)\|^2$.

Proof [Proof of Lemma 6.4 *i)*] With $\kappa = \lambda_2/\lambda_1$, the Fenchel transform of $h : \boldsymbol{w} \mapsto \|\boldsymbol{w}\|_2 + \kappa\|\boldsymbol{w}\|^2$.

$$
\begin{aligned}
h(\boldsymbol{u}) &= \|\boldsymbol{u}\|_2 + \tfrac{\kappa}{2}\|\boldsymbol{u}\|_2^2 \\
h^*(\boldsymbol{v}) &= \sup_{\boldsymbol{w}} \left(\boldsymbol{v}^\top \boldsymbol{w} - \|\boldsymbol{w}\|_2 - \tfrac{\kappa}{2}\|\boldsymbol{w}\|_2^2\right) \\
&= \tfrac{1}{2\kappa}\|\boldsymbol{v}\|_2^2 + \sup_{\boldsymbol{w}} \left(-\tfrac{\kappa}{2}\|\boldsymbol{w} - \boldsymbol{v}/\kappa\|_2^2 - \|\boldsymbol{w}\|_2\right) \\
&= \tfrac{1}{2\kappa}\|\boldsymbol{v}\|_2^2 - \inf_{\boldsymbol{w}} \left(\tfrac{\kappa}{2}\|\boldsymbol{w} - \boldsymbol{v}/\kappa\|_2^2 + \|\boldsymbol{w}\|_2\right) \\
&= \tfrac{1}{2\kappa}\|\boldsymbol{v}\|_2^2 - (\tfrac{\kappa}{2}\|\cdot\|_2^2 \square \|\cdot\|_2)(\boldsymbol{v}/\kappa) \ .
\end{aligned}
$$

■

Proof [Proof of Lemma 6.4 *ii)*]

$$
\begin{aligned}
(\tfrac{\kappa}{2}\|\cdot\|_2^2 \square \|\cdot\|_2)(\boldsymbol{v}) &= (\tfrac{\kappa}{2}\|\cdot\|_2^2 \square \|\cdot\|_2)^{**}(\boldsymbol{v}) \\
&= (\tfrac{1}{2\kappa}\|\cdot\|_2^2 + \iota_{\mathcal{B}_2})^*(\boldsymbol{v}) \\
&= \sup_{\|\boldsymbol{w}\|_2 \leq 1} \left(\boldsymbol{v}^\top \boldsymbol{w} - \tfrac{1}{2\kappa}\|\boldsymbol{w}\|_2^2\right) \\
&= \tfrac{\kappa}{2}\|\boldsymbol{v}\|^2 + \sup_{\|\boldsymbol{w}\|_2 \leq 1} -\tfrac{1}{2\kappa}\|\kappa\boldsymbol{v} - \boldsymbol{w}\|_2^2 \\
&= \tfrac{\kappa}{2}\|\boldsymbol{v}\|^2 - \tfrac{1}{2\kappa}\|\mathrm{BST}(\kappa\boldsymbol{v}, 1)\|_2^2 \ .
\end{aligned}
$$

■

$$
\begin{aligned}
g^*(\boldsymbol{u}) &= \lambda_1 h^*(\boldsymbol{u}/\lambda_1) \\
&= \frac{\lambda_1}{2\kappa}\|\mathrm{BST}(\boldsymbol{u}/\lambda_1, 1)\|^2 \\
&= \frac{\lambda_1^2}{2\lambda_2}\|\mathrm{BST}(\boldsymbol{u}/\lambda_1, 1)\|^2 \\
&= \frac{1}{\lambda_2}\|\mathrm{BST}(\boldsymbol{u}, \lambda_1)\|^2 \ .
\end{aligned}
$$

■

■

D. Experimental details

D.1. Disentangled representation coupled with sparsity regularization improves generalization

We consider the following data generating process: We sample the ground-truth features $\boldsymbol{f_\theta}(\boldsymbol{x})$ from a Gaussian distribution $\mathcal{N}(\mathbf{0}, \boldsymbol{\Sigma})$ where $\boldsymbol{\Sigma} \in \mathbb{R}^{m\times m}$ and $\boldsymbol{\Sigma}_{i,j} = 0.9^{|i-j|}$. Moreover, the labels are given by $y = \boldsymbol{w} \cdot \boldsymbol{f_\theta}(\boldsymbol{x}) + \epsilon$ where $\boldsymbol{w} \in \mathbb{R}^m$, $\epsilon \sim \mathcal{N}(0, 0.04)$ and $m = 100$. The ground-truth weight vector $\boldsymbol{w}$ is sampled once from $\mathcal{N}(0, I_{m\times m})$ and mask some of its components to zero: we vary the fraction of meaningful features (ℓ/m) from very sparse ($\ell/m = 5\%$) to less sparse ($\ell/m = 80\%$) settings. For each case, we study the sample complexity by varying the number of training samples from 25 to 150, but evaluating the generalization performance on a larger test dataset (1000 samples). To generate the entangled representations, we multiply the true latent variables $\boldsymbol{f_\theta}(\boldsymbol{x})$ by a randomly sampled orthogonal matrix $\boldsymbol{L}$, *i.e.,* $\boldsymbol{f_{\hat{\theta}}}(\boldsymbol{x}) := \boldsymbol{L}\boldsymbol{f_\theta}(\boldsymbol{x})$. For the disentangled representation, we simply consider the true latents, *i.e.,* $\boldsymbol{f_{\hat{\theta}}}(\boldsymbol{x}) := \boldsymbol{f_\theta}(\boldsymbol{x})$. Note that in principle we could have considered an invertible matrix $\boldsymbol{L}$ that is not orthogonal for the linearly entangled representation and a component-wise rescaling for the disentangled representation. The advantage of not doing so and opting for our approach is that the conditioning number of the covariance matrix of $\boldsymbol{f_{\hat{\theta}}}(\boldsymbol{x})$ is the same for both the entangled and the disentangled, hence offering a fairer comparison.

For both the case of entangled and disentangled representation, we solve the regression problem with Lasso and Ridge regression, where the associated hyperparameters (regularization strength) were inferred using 5-fold cross-validation on the input training dataset. Using both lasso and ridge regression would help us to show the effect of encouraging sparsity.

In Figure 6.1 for the sparsest case ($\ell/m = 5\%$), we observe that that Disentangled-Lasso approach has the best performance when we have fewer training samples, while the Entangled-Lasso approach performs the worst. As we increase the number of training samples, the performance of Entangled-Lasso approaches that of Disentangled-Lasso, however, learning under the Disentangled-Lasso approach is sample efficient. Disentangled-Lasso obtains R^2 greater than 0.5 with only 25 training samples, while other approaches obtain R^2 close to zero. Also, Disentagled-Lasso converges to the optimal R^2 using only 50 training samples, while Entangled-Lasso does the same with 150 samples.

Note that the improvement due to disentanglement does not happen for the case of ridge regression as expected and there is no difference between the methods Disentangled-Ridge and Entangled-Ridge because the L2 norm is invariant to orthogonal transformation. Also, having sparsity in the underlying task is important. Disentangled-Lasso shows the max improvement for

the case of $\ell/m = 5\%$, with the gains reducing as we decrease the sparsity in the underlying task ($l/m = 80\%$).

D.2. Disentanglement in 3D Shapes

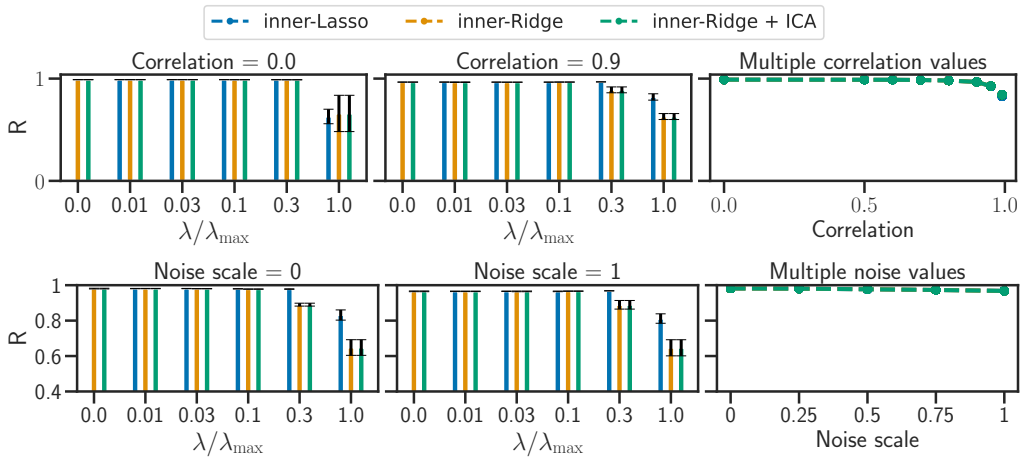


Figure 6.6. Prediction performance (R Score) for inner-Lasso, inner-Ridge and inner-Ridge combined with ICA as a function of the regularization parameter (left and middle). Varying level of correlation between latents (top) and noise on the latents (bottom). The right columns shows performance of the best hyperparameter for different values of correlation and noise levels.

D.2.1. Dataset generation. Details on 3D Shapes. The 3D Shapes dataset [Burgess and Kim, 2018] contains synthetic images of colored shapes resting in a simple 3D scene. These images vary across 6 factors: Floor hue (10 values linearly spaced in [0, 1]); Wall hue (10 values linearly spaced in [0, 1]); Object hue (10 values linearly spaced in [0, 1]); Scale (8 values linearly spaced in [0, 1]); Shape (4 values in [0, 1, 2, 3]); and Orientation (15 values linearly spaced in [-30, 30]). These are the factors we aim to disentangle. We standardize them to have mean 0 and variance 1. We denote by $\mathcal{Z} \subseteq \mathbb{R}^6$, the set of all possible latent factor combinations. In our framework, this corresponds to the support of the ground-truth features $\boldsymbol{f_\theta}(\boldsymbol{x})$. We note that the points in $\mathcal{Z}$ are arranged in a grid-like fashion in $\mathbb{R}^6$.

Task generation. For all tasks t, the labelled dataset $\mathcal{D}_t = \{(\boldsymbol{x}^{(t,i)}), y^{(t,i)})\}_{i=1}^n$ is generated by first sampling the ground-truth latent variables $\boldsymbol{z}^{(t,i)} := \boldsymbol{f_\theta}(\boldsymbol{x}^{(t,i)})$ i.i.d. according to some distribution $p(\boldsymbol{z})$ over $\mathcal{Z}$, while the corresponding input is obtained doing $\boldsymbol{x}^{(t,i)} := \boldsymbol{f_\theta}^{-1}(\boldsymbol{z}^{(t,i)})$ ($\boldsymbol{f_\theta}$ is invertible in 3D Shapes). Then, a sparse weight vector $\boldsymbol{w}^{(t)}$ is sampled randomly by doing $\boldsymbol{w}^{(t)} := \bar{\boldsymbol{w}}^{(t)} \odot \boldsymbol{s}^{(t)}$, were $\odot$ is the Hadamard (component-wise) product, $\bar{\boldsymbol{w}}^{(t)} \sim \mathcal{N}(\boldsymbol{0}, I)$ and $\boldsymbol{s} \in \{0,1\}^6$ is a binary vector with independent components sampled from a Bernoulli distribution with ($p = 0.5$). Then, the labels are computedfor each example as $y^{(t,i)} := \boldsymbol{w}^{(t)} \cdot \boldsymbol{x}^{(t,i)} + \epsilon^{(t,i)}$, where $\epsilon^{(t,i)}$ is independent Gaussian noise. In every task, the dataset has size $n = 50$. New tasks are generated continuously as we train. Figures 6.4 & 6.6 explores various choices of $p(\boldsymbol{z})$, *i.e.,* by

varying the level of correlation between the latent variables and by varying the level of noise on the ground-truth latents. Figure 6.7 shows a visualization of some of these distributions over latents.

Noise on latents. To make the dataset slightly more realistic, we get rid of the artificial grid-like structure of the latents by adding noise to it. This procedure transforms $\mathcal{Z}$ into a new support $\mathcal{Z}_\alpha$, where α is the noise level. Formally, $\mathcal{Z}_\alpha := \bigcup_{\boldsymbol{z}\in\mathcal{Z}}\{\boldsymbol{z}+\boldsymbol{u}_{\boldsymbol{z}}\}$ where the $\boldsymbol{u}_z$ are i.i.d samples from the uniform over the hypercube

$$\left[-\alpha\frac{\Delta \boldsymbol{z}_1}{2}, \alpha\frac{\Delta \boldsymbol{z}_1}{2}\right] \times \left[-\alpha\frac{\Delta \boldsymbol{z}_2}{2}, \alpha\frac{\Delta \boldsymbol{z}_2}{2}\right] \times \ldots \times \left[-\alpha\frac{\Delta \boldsymbol{z}_6}{2}, \alpha\frac{\Delta \boldsymbol{z}_6}{2}\right],$$

where $\Delta \boldsymbol{z}_i$ denotes the gap between contiguous values of the factor $\boldsymbol{z}_i$. When $\alpha = 0$, no noise is added and the support $\mathcal{Z}$ is unchanged, *i.e.,* $\mathcal{Z}_1 = \mathcal{Z}$. As long as $\alpha \in [0, 1]$, contiguous points in $\mathcal{Z}$ cannot be interchanged in $\mathcal{Z}_\alpha$. We also clarify that the ground-truth mapping $\boldsymbol{f}_{\boldsymbol{\theta}}$ is modified to $\boldsymbol{f}_{\boldsymbol{\theta},\alpha}$ consequently: for all $\boldsymbol{x} \in \mathcal{X}$, $\boldsymbol{f}_{\boldsymbol{\theta},\alpha}(\boldsymbol{x}) := \boldsymbol{f}_{\boldsymbol{\theta}}(\boldsymbol{x}) + \boldsymbol{u}_{\boldsymbol{z}}$. We emphasize that the $\boldsymbol{u}_{\boldsymbol{z}}$ are sampled only once such that $\boldsymbol{f}_{\boldsymbol{\theta},\alpha}(\boldsymbol{x})$ is actually a deterministic mapping.

Varying correlations. To verify that our approach is robust to correlations in the latents, we construct $p(\boldsymbol{z})$ as follows: We consider a Gaussian density centred at $\mathbf{0}$ with covariance $\boldsymbol{\Sigma}_{i,j} := \rho + \mathbb{1}(i = j)(1 - \rho)$. Then, we evaluate this density on the points of $\mathcal{Z}_\alpha$ and renormalize to have a well-defined probability distribution over $\mathcal{Z}_\alpha$. We denote by $p_{\alpha,\rho}(\boldsymbol{z})$ the distribution obtain by this construction.

In the top rows of Figures 6.4 & 6.6, the latents are sampled from $p_{\alpha=1,\rho}(\boldsymbol{z})$ and ρ varies between 0 and 0.99. In the bottom rows of Figures 6.4 & 6.6, the latents are sampled from $p_{\alpha,\rho=0.9}(\boldsymbol{z})$ and α varies from 0 to 1.

D.2.2. Metrics. We evaluate disentanglement via the *mean correlation coefficient* [Hyvarinen and Morioka, 2016, Khemakhem et al., 2020a] which is computed as follows: The Pearson correlation matrix C between the ground-truth features and learned ones is computed. Then, $\text{MCC} = \max_{\pi\in\text{permutations}} \frac{1}{m}\sum_{j=1}^{m} |C_{j,\pi(j)}|$. We also evaluate linear equivalence by performing linear regression to predict the ground-truth factors from the learned ones, and report the mean of the Pearson correlations between the ground-truth latents and the learned ones. This metric is known as the *coefficient of multiple correlations*, R, and turns out to be the square-root of the more widely known *coefficient of determination*, R^2. The advantage of using R over R^2 is that we always have $\text{MCC} \leq R$.

D.2.3. Architecture, inner solver & hyperparameters. We use the four-layer convolutional neural network typically used in the disentanglement literature [Locatello et al., 2019]. As mentioned in Section 6.3.4, the norm of the representation $\boldsymbol{f}_{\hat{\boldsymbol{\theta}}}(\boldsymbol{x})$ must be controlled to make sure the regularization remains effective. To do so, we apply batch normalization [Ioffe and Szegedy, 2015] at the very last layer of the neural network and do not learn its scale and shift parameters.

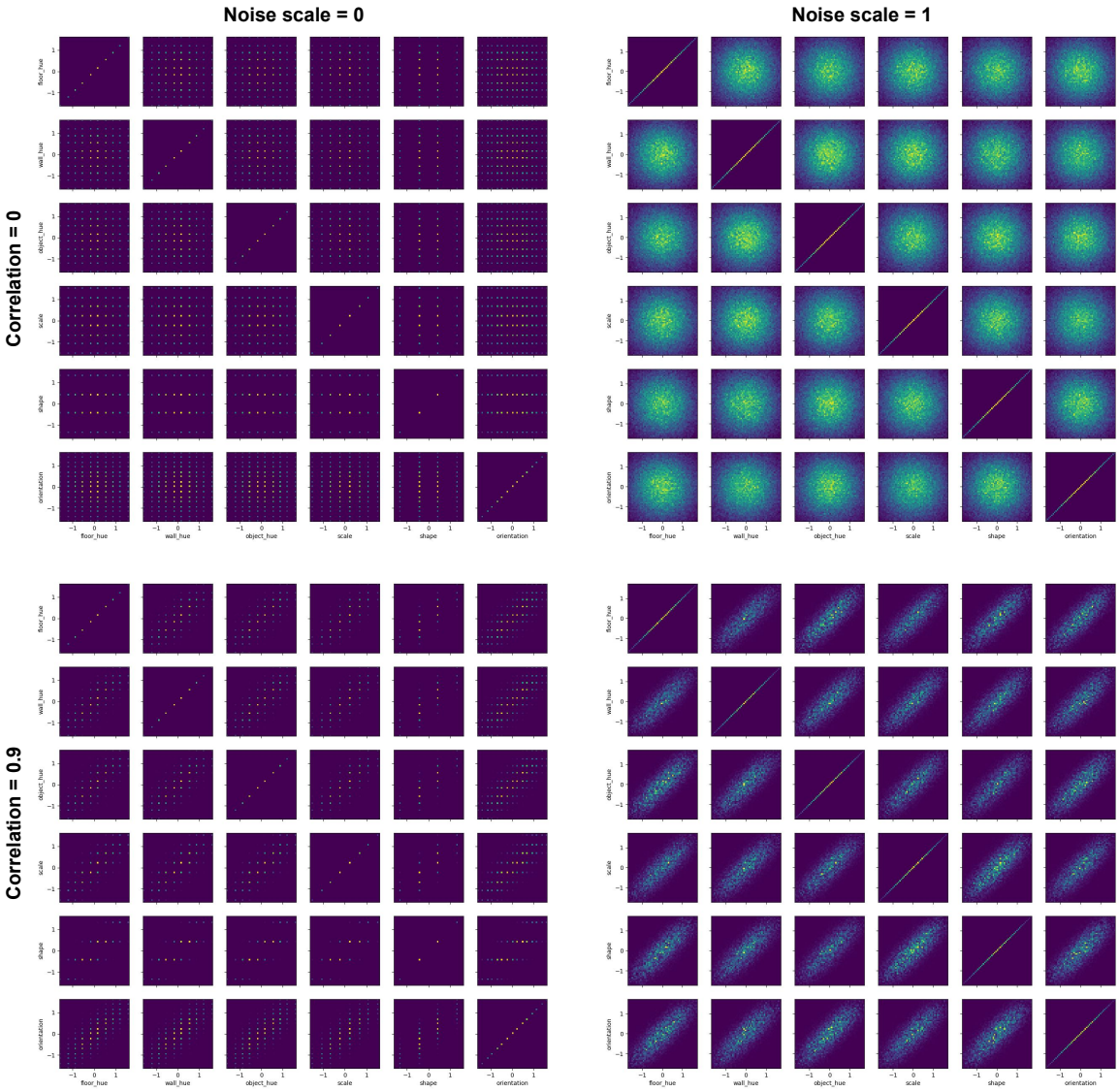


Figure 6.7. Visualization of the various distributions over latents. For 4 combinations of correlation levels and noise levels, we show the 2-dimensional histograms of samples from the corresponding distribution over latents described in Appendix D.2.1. Each histogram shows the joint distribution over two latent factors.

Empirically, we do see the expected behavior that, without any normalization, the norm of $\boldsymbol{f}_{\hat{\boldsymbol{\theta}}}(\boldsymbol{x})$ explodes as we train, leading to instabilities and low sparsity.

In these experiments, the distribution $p(y; \boldsymbol{\eta})$ used for learning is a Gaussian with fixed variance. In that case, the inner problem of Section 6.3.4 reduces to Lasso regression. Computing the hypergradient w.r.t. $\boldsymbol{\theta}$ requires solving this inner problem. To do so, we use Proximal Coordinate Descent [Tseng, 2001, Richtárik and Takáč, 2014].

Details on $\lambda/\lambda_{\max}$. In Figures 6.4, & 6.6, we explore various levels of regularization λ. In our implementation, we set $\lambda = \epsilon\lambda_{\max}$ where $\epsilon \geq 0$. In inner-Lasso, we set $\lambda_{\max} := \frac{1}{n}\|\boldsymbol{F}^\top \boldsymbol{y}\|_\infty$

($\boldsymbol{F} \in \mathbb{R}^{n \times m}$ is the design matrix of the features of the samples of a task), while in inner-Ridge we have $\lambda_{\max} := \frac{1}{n}\|\boldsymbol{F}\|^2$. Note that this means λ is dynamically changing as we train because $\boldsymbol{F}$ changes. However we never backpropagate through $\lambda_{\max}$ (we block the gradient from flowing). Thus, in all figures, we report $\epsilon = \lambda / \lambda_{\max}$.

D.2.4. Experiments violating assumptions. In this section, we explore variations of the experiments of Section 6.5, but this time the assumptions of Theorem 6.1 are violated.

Figure 6.8 shows different degrees of violation of Assumption 6.7. We consider the cases where $\mathcal{S} := \{\{1,2\},\{3,4\},\{5,6\}\}$ (block size = 2), $\mathcal{S} := \{\{1,2,3\},\{4,5,6\}\}$ (block size = 3) and $\mathcal{S} := \{\{1,2,3,4,5,6\}\}$ (block size = 6). Note that the latter case corresponds to having no sparsity at all in the ground-truth model, *i.e.,* all tasks require all features. The reader can verify that these three cases indeed violate Assumption 6.7. In all cases, the distribution $p(S)$ puts uniform mass over its support $\mathcal{S}$. Similarly to the experiments from the main text, $\boldsymbol{w} := \bar{\boldsymbol{w}} \odot \boldsymbol{s}$, where $\bar{\boldsymbol{w}} \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{I})$ and $\boldsymbol{s} \sim p(S)$ ($\boldsymbol{s}$ is the binary representation of the set S). Overall, we can see that inner-Lasso does not perform as well when Assumption 6.7 is violated. For example, when there is no sparsity at all (block size = 6), inner-Lasso performs poorly and is even surpassed by inner-Ridge. Nevertheless, for mild violations (block size = 2), disentanglement (as measured by MCC) remains reasonably high. We further notice that all methods obtain very good R score in all settings. This is expected in light of Theorem 6.2, which guarantees identifiability up to linear transformation without requiring Assumption 6.7.

Figure 6.9 presents experiments that are identitical to those of Figure 6.4 in the main text, except for how $\boldsymbol{w}$ is generated. Here, the components of $\boldsymbol{w}$ are sampled independently according to $\boldsymbol{w}_i \sim \text{Laplace}(\mu = 0, b = 1)$. We note that, under this process, the probability that $\boldsymbol{w}_i = 0$ is zero. This means all features are useful and Assumption 6.7 is violated. That being said, due to the fat tail behavior of the Laplacian distribution, many components of $\boldsymbol{w}$ will be close to zero (relatively to its variance). Thus, this can be thought of as a weaker form of sparsity where many features are relatively unimportant. Figure 6.9 shows that inner-Lasso can still disentangle very well. In fact, the performance is very similar to the experiments that presented actual sparsity (Figure 6.4).

D.2.5. Experiments with regularization in the outer problem. Theorem 6.4 presented an alternative optimization problem to that of Theorem 6.1 to learn a disentangled representation. Appendix B.3 presented a tractable relaxation of this alternative. The essential operational difference is that the sparsity regulatization appears in the outer problem instead of the inner problem. Figure 6.10 shows this alternative works as well empirically. Details in the caption.

D.2.6. Visual evaluation. Figures 6.11, 6.12, 6.13 & 6.14 show how various learned representations respond to changing a single factor of variation in the image [Higgins et al., 2017, Figure 7.A.B].

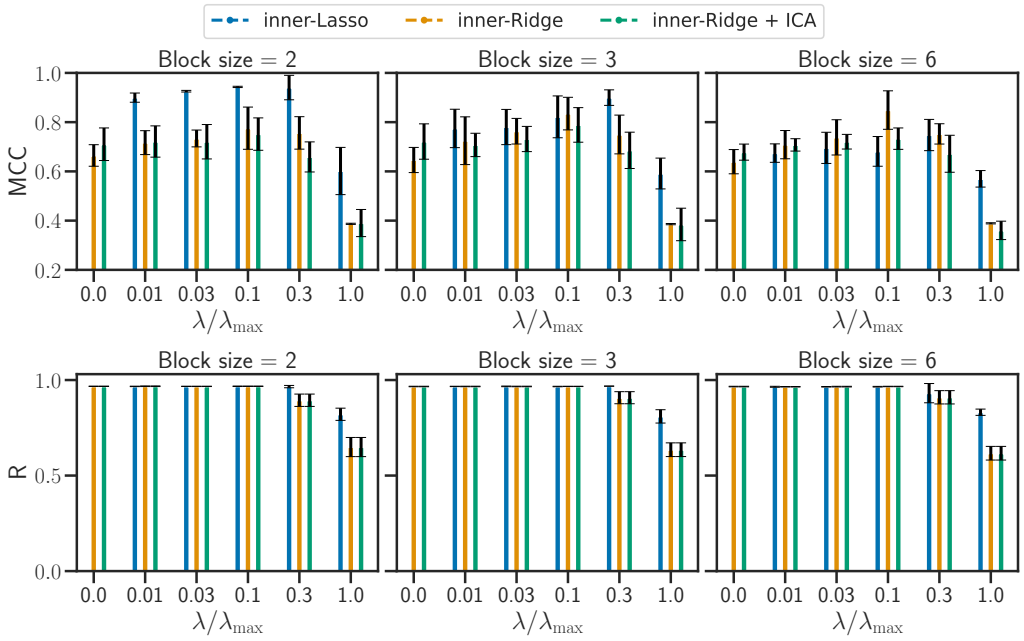


Figure 6.8. Disentanglement (MCC, top) and prediction (R Score, bottom) performances for inner-Lasso, inner-Ridge and inner-Ridge combined with ICA as a function of the regularization parameter. The metrics are plotted for multiple value of block size for the support. Block size $= 6$ corresponds to no sparsity in the ground truth coefficients.

We see what was expected: the higher the MCC, the more disentangled the learned features appear, thus validating MCC as a good metric for disentanglement. See captions for details.

D.2.7. Additional metrics for disentanglement. We implemented metrics from the DCI framework [Eastwood and Williams, 2018] to evaluate disentanglement. 1) DCI-Disentanglement: How many ground truth latent components are related to a particular component of the learned latent representation; 2) DCI-Completeness: How many learned latent components are related to a particular component of the ground truth latent representation. Note that for the definition of disentanglement used in the present work Definition 6.1, we want both DCI-disentanglement and DCI-completeness to be high.

The DCI framework requires a matrix of relative importance. In our implementation, this matrix is the coefficient matrix resulting from performing linear regression with inputs as the learned latent representation $\boldsymbol{f}_{\hat{\theta}}(\boldsymbol{x})$ and targets as the ground truth latent representation $\boldsymbol{f}_{\boldsymbol{\theta}}(\boldsymbol{x})$, and denote the solution as the matrix W. Further, denote by $I = |W|$ as the importance matrix, as $I_{i,j}$ denotes the relevance of inferred latent $\boldsymbol{f}_{\hat{\theta}}(\boldsymbol{x})_j$ for predicting the true latent $\boldsymbol{f}_{\boldsymbol{\theta}}(\boldsymbol{x})_i$.

Now, for computing DCI-disentanglement, we normalize each row of the importance matrix $I[i,:]$ by its sum so that it represents a probability distribution. Then disentanglement is given by $\frac{1}{m} \times \sum_i^m 1 - H(I[i,:])$, where H denotes the entropy of a distribution. Note that for the desired case of each ground truth latent component being explained by a single inferred latent component,

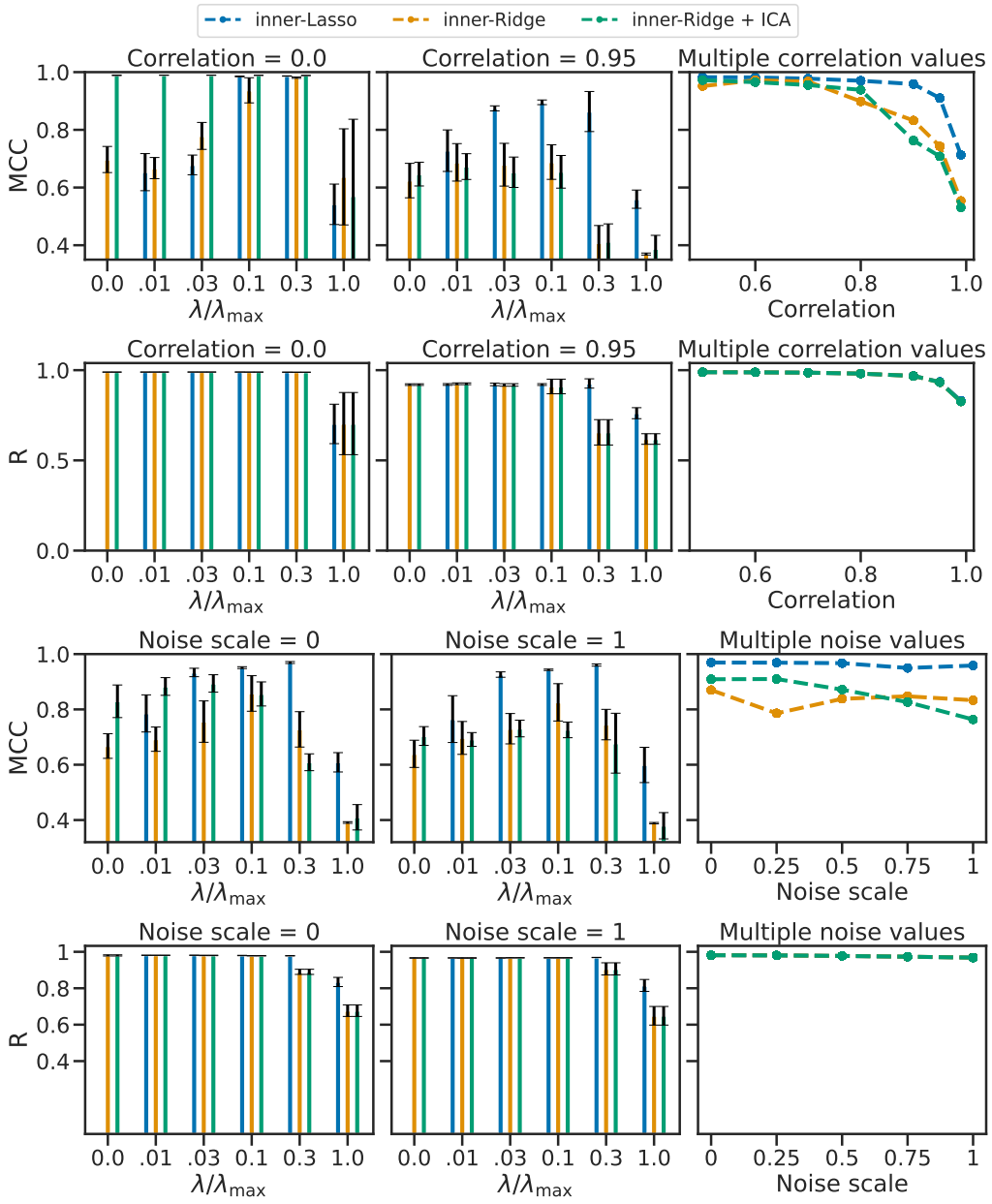


Figure 6.9. Same experiment as Figure 6.4, but the task coefficient vectors $\boldsymbol{w}$ are sampled from a Laplacian distribution (instead of what was described in Appendix D.2.1). Performance is barely affected, showing some amount of robustness to violations of Assumption 6.7.

we would have $H(I[i,:]) = 0$ as we have a one-hot vector for the probability distribution. Similarly, for the case of each ground truth latent component being explained uniformly by all the inferred latents, $H(I[i,:])$ would be maximized and hence the DCI score would be minimized. To compute the DCI-completeness, we first normalize each column of the importance matrix $I[:,j]$ by its sum so that it represents a probability distribution and then compute $\frac{1}{m} \times \sum_i^m 1 - H(I[:,j])$.

Figure 6.15 shows the results for the 3D Shapes experiments (Section 6.5) with the DCI metric to evaluate disentanglement. Notice that we find the same trend as we had with the MCC metric 6.4,

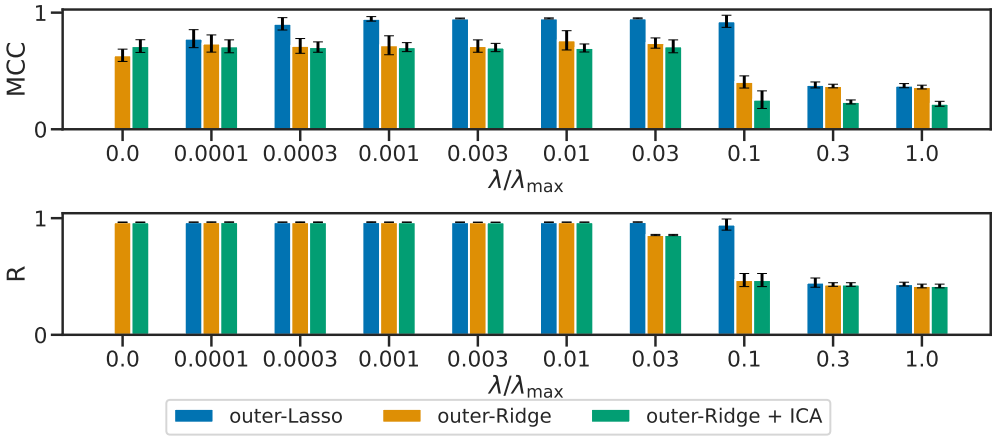


Figure 6.10. outer-Lasso solves Problem (6.62) (with regularization in the outer problem) while **outer-Ridge** solves the same problem but with an L_2-norm instead of $L_{2,1}$. The method **outer-Ridge + ICA** is outer-Ridge with an additional step of linear ICA on top of the learned representation. The results obtained are very similar to the main results of Figures 6.4, 6.6. In this dataset, the latents are sampled from $p_{\alpha=1,\rho=0.9}(\boldsymbol{z})$ (See Appendix D.2.1) and the weight coefficients are sampled from the binomial-Gaussian process described in Appendix D.2.1.

that inner-Lasso is more robust to correlation between the latent variables, and inner-Ridge + ICA performance drops down significantly with increasing correlation.

D.3. Meta-learning experiments

Experimental settings. We evaluate the performance of our meta-learning algorithm based on a group-sparse SVM learners on the *mini*ImageNet [Vinyals et al., 2016] dataset. Following the standard nomenclature in few-shot classification [Hospedales et al., 2021] with k-shot N-way, where N is the number of classes in each classification task, and k is the number of samples per class in the training dataset $\mathcal{D}_t^{\text{train}}$, we consider the experimental setting 5-shot 5-way. We use the same residual network architecture as in [Lee et al., 2019], with 12 layers and a representation of size $p = 1.6 \times 10^4$.

Technical details. The objective of Problem (6.10) is composed of a smooth term and block separable non-smooth term, hence it can be solved efficiently using proximal block coordinate descent [Tseng, 2001]. Although Theorem 6.1 is not directly applicable to the meta-learning formulation proposed in this section, we conjecture that similar techniques could be reused to prove an identifiability result in this setting. As in Section 6.3.4, the argmin differentiation of the solution of Problem (6.10) can be done using implicit differentiation [Bertrand et al., 2022].

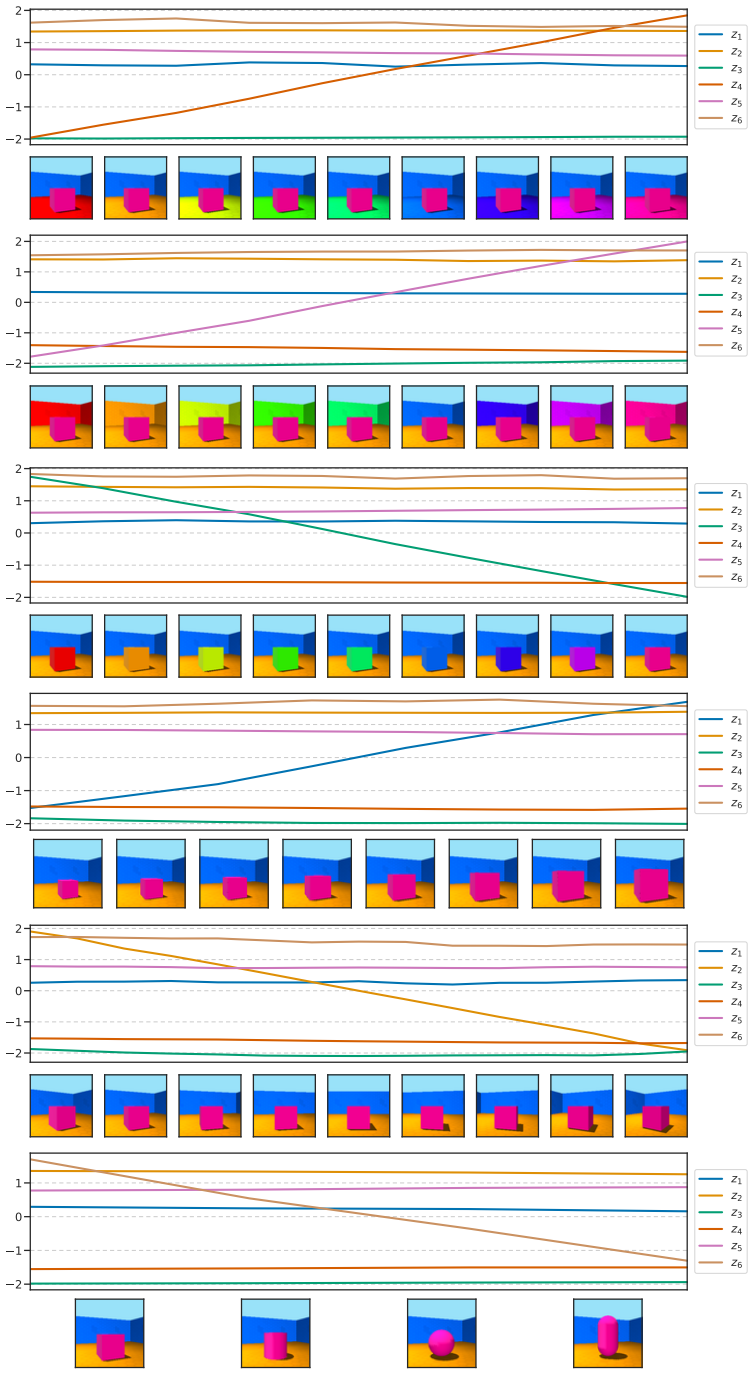


Figure 6.11. Varying one factor at a time in the image and showing how the learned representation varies in response. This representation was learned by **inner-Lasso** (best hyperparameter) on a dataset with **0 correlation between latents** and a noise scale of 1. The corresponding **MCC is 0.99**. We can see that varying a single factor in the image always result in changing a single factor in the learned representation.

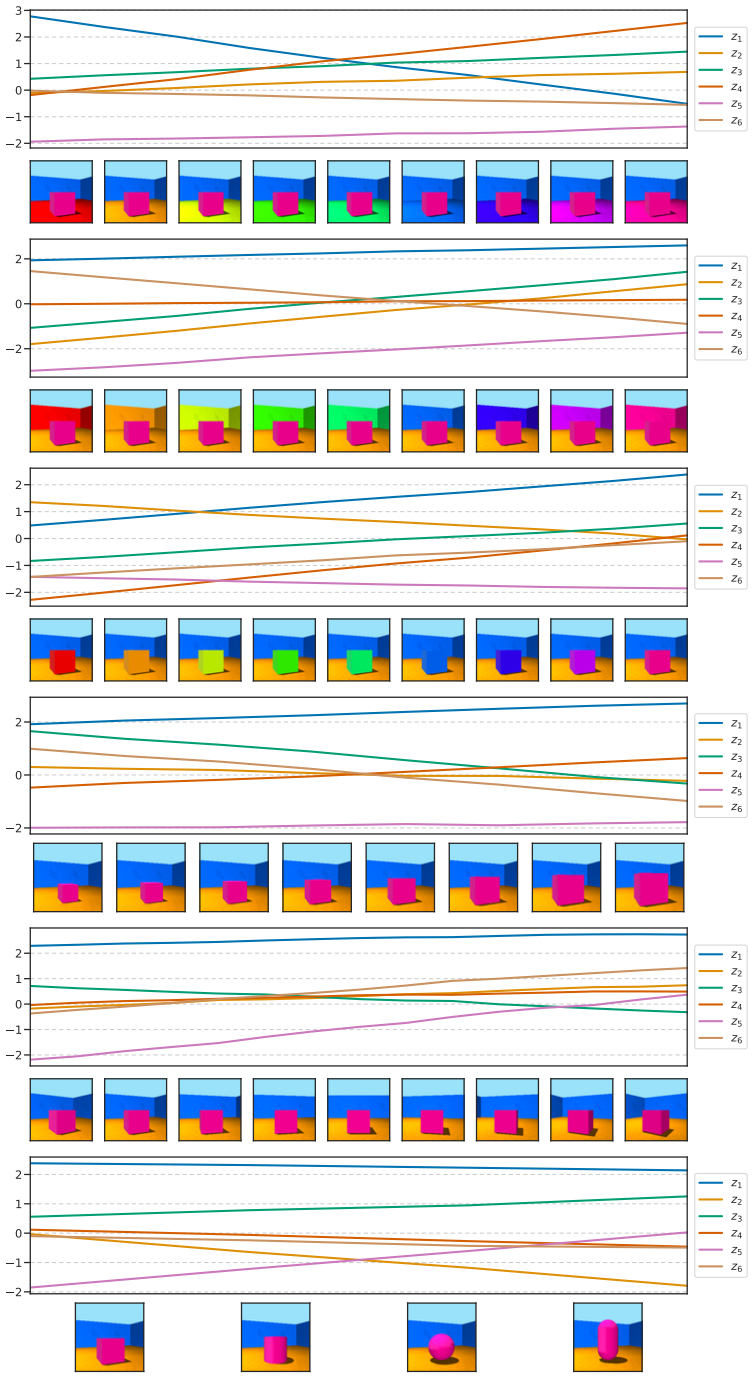


Figure 6.12. Varying one factor at a time in the image and showing how the learned representation varies in response. This representation was learned **without regularization** of any kind (*i.e.,* with inner-Ridge with regularization coefficient equal to zero) on a dataset with **0 correlation between** and a noise scale of 1. The corresponding **MCC is 0.63**. We can see that varying a single factor in the image result in changing multiple factors in the learned representation, *i.e.,* the representation is not disentangled.

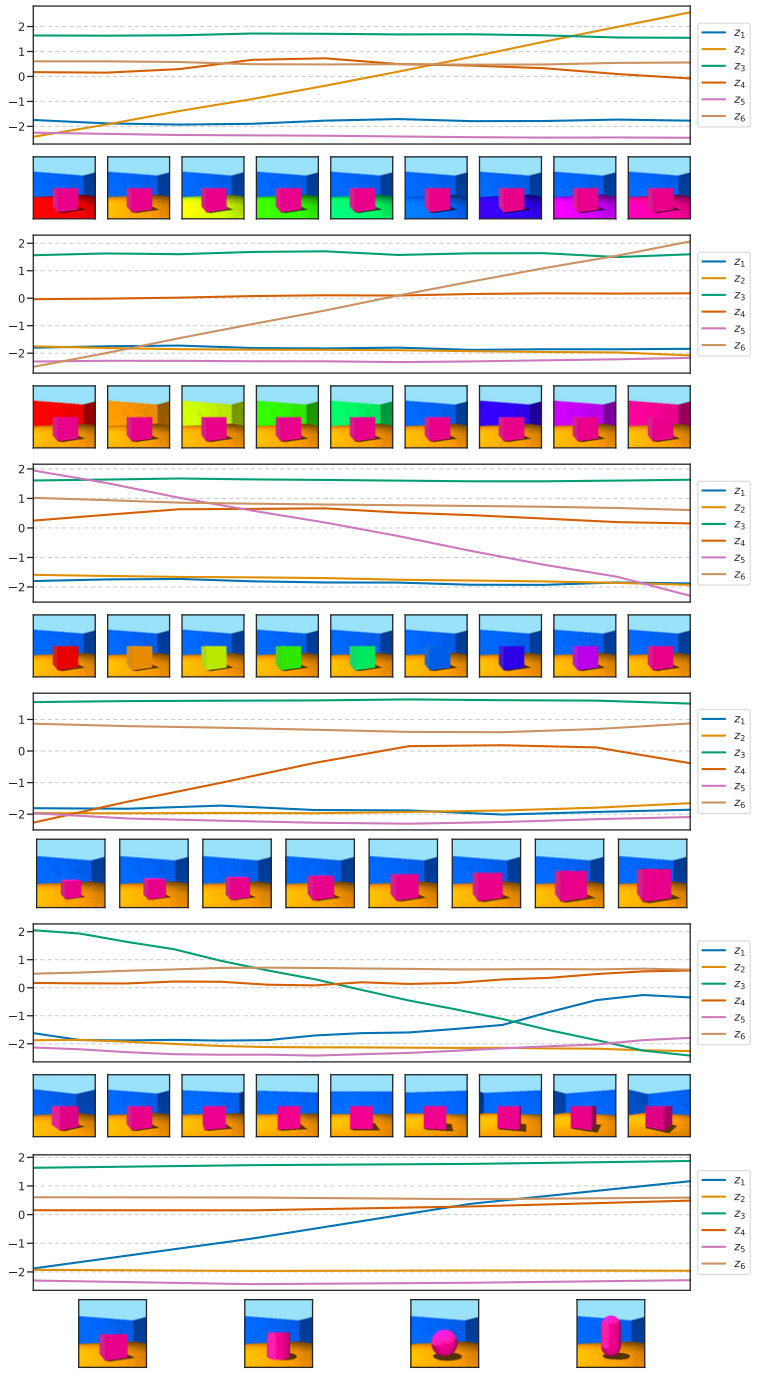


Figure 6.13. Varying one factor at a time in the image and showing how the learned representation varies in response. This representation was learned with **inner-Lasso** (best hyperparameter) on a dataset with **correlation 0.9 between latents** and a noise scale of 1. The corresponding **MCC is 0.96**. Qualitatively, the representation appears to be well disentangled, but not as well as in Figure 6.11 (reflected by a drop in MCC of 0.03).

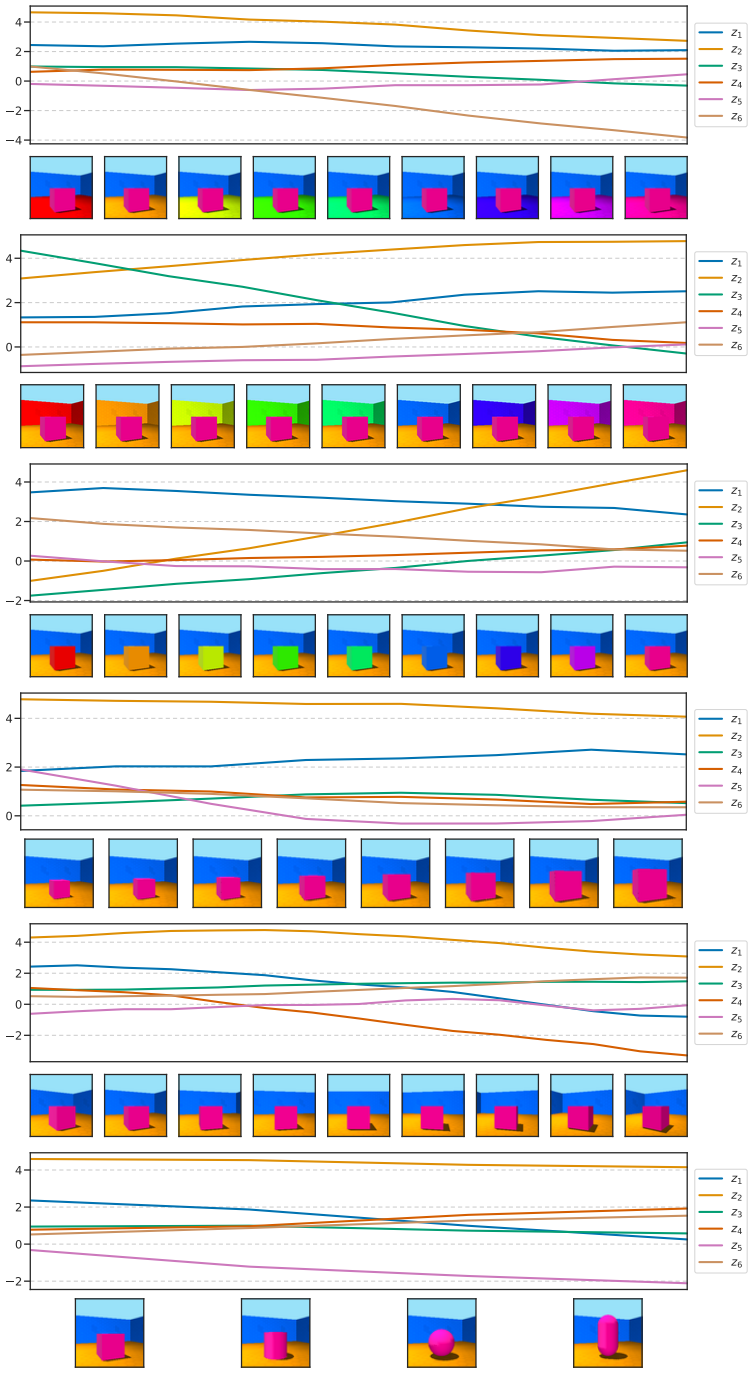


Figure 6.14. Varying one factor at a time in the image and showing how the learned representation varies in response. This representation was learned with **inner-Ridge** (best hyperparameter) on a dataset with **correlation 0.9 between latents** and a noise scale of 1. The corresponding **MCC is 0.79**. For most latent factors, we cannot identify a dominant feature, except maybe for background and object colors. The representation appears more disentangled than Figure 6.12, but less disentangled than Figure 6.13, as reflected by their corresponding MCC values.

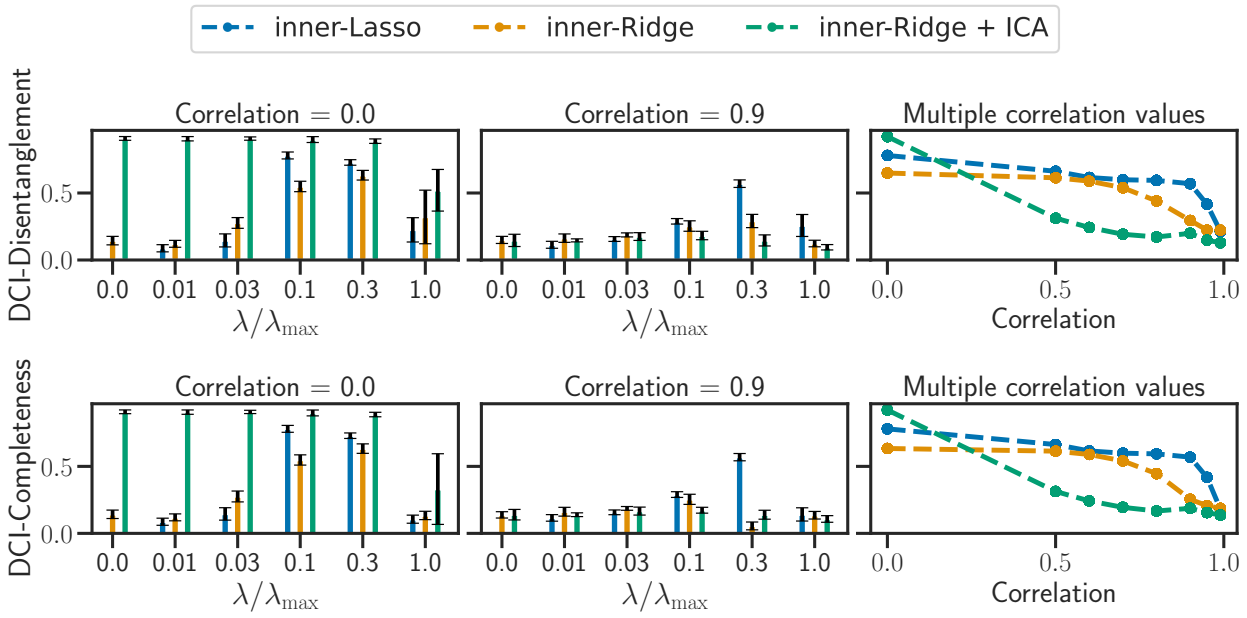


Figure 6.15. Disentanglement performance (DCI) for inner-Lasso, inner-Ridge and inner-Ridge combined with ICA as a function of the regularization parameter (left and middle). The right columns shows performance of the best hyperparameter for different values of correlation and noise. The top row shows the results for the disentanglement metric of DCI and the bottom row shows the results for the completeness metric of DCI.

Prologue to the Fifth Contribution

Article Details

Additive Decoders for Latent Variables Identification and Cartesian-Product Extrapolation by Sébastien Lachapelle*, Divyat Mahajan*, Ioannis Mitliagkas and Simon Lacoste-Julien. This work was published at the Thirty-seventh Conference on Neural Information Processing Systems (NeurIPS 2023) with an **oral**.
*Equal contributions.

Contributions of the Authors

Sébastien Lachapelle developed the ideas, the proofs, wrote the first draft and contributed to the experimental design. **Divyat Mahajan** led the experiments and contributed to the writing. **Ioannis Mitliagkas** contributed to the writing. **Simon Lacoste-Julien** provided supervison, contributed to the writing and contributed to some technical aspects of the theory.

Context and Limitations

Similarly to the previous contribution, this work was born from a desire to show rigorously how disentanglement can help for some downstream goal. Here, the goal is inspired from recent strides in generative models which now appear to be able to generate realistic images resulting from a combination of concepts that was not present in the training data [Ramesh et al., 2022, Rombach et al., 2022]. How and when is it possible to generate images that were not present in the support of the training distribution but that are *on the manifold of reasonable images*? To start answering this question, we propose additivity as an assumption on the mixing function and show that, under further regularity conditions, it is sufficient to obtain both disentanglement and compositional generalization (Cartesian-product extrapolation in the paper). Also see Section 8.2.2 from Chapter 8 for a formalization of this goal of extrapolation within statistical decision theory.

We also motivate our contribution as a first step towards explaining why object-centric representation learning (OCRL) such as Slot-Attention [Locatello et al., 2020c] can perform segmentation

without any segmentation labels, i.e. without supervision. Although additive decoders are a crude simplification of decoders actually used in OCRL, we believe this analysis illustrates how identifiability analyses can be used to shed light on intriguing phenomena observed in practice. We expand on this point further in Chapter 9.

The strong disentanglement and extrapolation guarantees of additive decoders come at the cost of low expressivity. For instance, they cannot represent simple datasets with occlusion (Appendix A.12) or with a variable number of objects, unlike OCRL decoders. The theory also assumes the number of latent factors to be known, which do not reflect practice. Hopefully, future analyses will shed light on these open questions.

Recent developments

The work of Wiedemer et al. [2023], which was also presented at NeurIPS 2023, also studies compositionality in a similar context. Essentially, they show that, if two functions of the form $\boldsymbol{c}(\boldsymbol{f}^{(B_1)}(\boldsymbol{z}_{B_1}), ..., \boldsymbol{f}^{(B_\ell)}(\boldsymbol{z}_{B_\ell}))$ and $\boldsymbol{c}(\hat{\boldsymbol{f}}^{(B_1)}(\boldsymbol{z}_{B_1}), ..., \hat{\boldsymbol{f}}^{(B_\ell)}(\boldsymbol{z}_{B_\ell}))$ are equal on a set $\mathcal{Z}^{\text{train}}$, they must be equal on its Cartesian-product extension (they use a terminology different from ours). Note that $\boldsymbol{c}$ is known here, i.e. "$\boldsymbol{c} = \hat{\boldsymbol{c}}$". The key difference with our work is that there is no discussion about disentanglement and identifiability, since $\boldsymbol{z}$ is assumed to be observed.

In a follow-up work which got an oral at ICLR 2024 [Wiedemer et al., 2024], the authors tackle the problem of disentanglement. They leverage the identifiability of *compositional decoders* shown by Brady et al. [2023] and the fact that these are additive to show that both disentanglement and compositional generalization are possible. While our identifiability result is stronger in the sense that additive decoders are strictly more expressive than compositional ones (Appendix A.3), Wiedemer et al. [2024] proposes an additional regularizer which enables extrapolation of the *encoder*, which we did not.

Chapter 7

Additive Decoders for Latent Variables Identification and Cartesian-Product Extrapolation

Abstract

We tackle the problems of latent variables identification and "out-of-support" image generation in representation learning. We show that both are possible for a class of decoders that we call *additive*, which are reminiscent of decoders used for object-centric representation learning (OCRL) and well suited for images that can be decomposed as a sum of object-specific images. We provide conditions under which exactly solving the reconstruction problem using an additive decoder is guaranteed to identify the blocks of latent variables up to permutation and block-wise invertible transformations. This guarantee relies only on very weak assumptions about the distribution of the latent factors, which might present statistical dependencies and have an almost arbitrarily shaped support. Our result provides a new setting where nonlinear independent component analysis (ICA) is possible and adds to our theoretical understanding of OCRL methods. We also show theoretically that additive decoders can generate novel images by recombining observed factors of variations in novel ways, an ability we refer to as *Cartesian-product extrapolation*. We show empirically that additivity is crucial for both identifiability and extrapolation on simulated data.

7.1. Introduction

The integration of connectionist and symbolic approaches to artificial intelligence has been proposed as a solution to the lack of robustness, transferability, systematic generalization and interpretability of current deep learning algorithms [Marcus, 2001, Bengio et al., 2013, d'Avila Garcez and Lamb, 2020, Greff et al., 2020, Goyal and Bengio, 2021] with justifications rooted in cognitive sciences [Fodor and Pylyshyn, 1988, Harnad, 1990, Lake et al., 2017] and causality [Pearl, 2019, Schölkopf et al., 2021]. However, the problem of extracting meaningful symbols grounded in

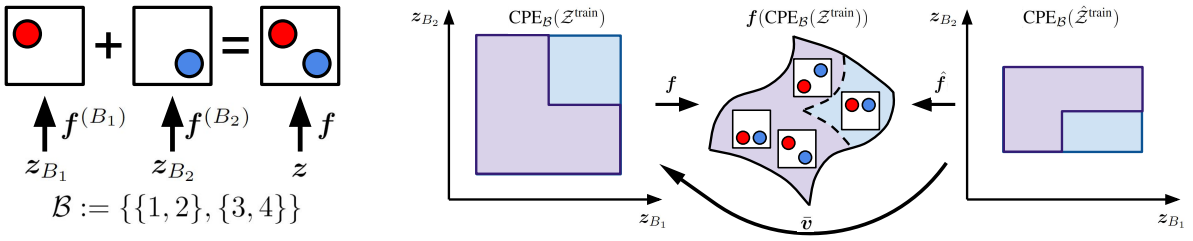


Figure 7.1. Left: Additive decoders model the additive structure of scenes composed of multiple objects. **Right:** Additive decoders allow to generate novel images never seen during training via Cartesian-product extrapolation (Corollary 7.1). Purple regions correspond to latents/observations seen during training. The blue regions correspond to the Cartesian-product extension. The middle set is the manifold of images of balls. In this example, the learner never saw both balls high, but these can be generated nevertheless thanks to the additive nature of the scene. Details in Section 7.3.2.

low-level observations, e.g. images, is still open. This problem is sometime referred to as *disentanglement* [Bengio et al., 2013, Locatello et al., 2019] or *causal representation learning* [Schölkopf et al., 2021]. The question of *identifiability* in representation learning, which originated in works on *nonlinear independent component analysis* (ICA) [Taleb and Jutten, 1999, Hyvarinen and Morioka, 2017, Hyvärinen et al., 2019, Khemakhem et al., 2020a], has been the focus of many recent efforts [Locatello et al., 2020a, Von Kügelgen et al., 2021, Gresele et al., 2021, Lippe et al., 2022, Ahuja et al., 2023, Buchholz et al., 2022, Lachapelle et al., 2023a]. The mathematical results of these works provide rigorous explanations for when and why symbolic representations can be extracted from low-level observations. In a similar spirit, *Object-centric representation learning* (OCRL) aims to learn a representation in which the information about different objects are encoded separately [Eslami et al., 2016, Greff et al., 2016, Burgess et al., 2019, Greff et al., 2019, Engelcke et al., 2020, Locatello et al., 2020c, Dittadi et al., 2022]. These approaches have shown impressive results empirically, but the exact reason why they can perform this form of segmentation without any supervision is poorly understood.

7.1.1. Contributions

Our first contribution is an analysis of the identifiability of a class of decoders we call *additive* (Definition 7.1). Essentially, a decoder $\boldsymbol{f}(\boldsymbol{z})$ acting on a latent vector $\boldsymbol{z} \in \mathbb{R}^{d_z}$ to produce an observation $\boldsymbol{x}$ is said to be additive if it can be written as $\boldsymbol{f}(\boldsymbol{z}) = \sum_{B\in\mathcal{B}} \boldsymbol{f}^{(B)}(\boldsymbol{z}_B)$ where $\mathcal{B}$ is a partition of $\{1,\dots,d_z\}$, $\boldsymbol{f}^{(B)}(\boldsymbol{z}_B)$ are "block-specific" decoders and the $\boldsymbol{z}_B$ are non-overlapping subvectors of $\boldsymbol{z}$. This class of decoder is particularly well suited for images $\boldsymbol{x}$ that can be expressed as a sum of images corresponding to different objects (left of Figure 7.1). Unsurprisingly, this class of decoder bears similarity with the decoding architectures used in OCRL (Section 7.2), which already showed important successes at disentangling objects without any supervision. Our identifiability results provide conditions under which exactly solving the reconstruction problem with

an additive decoder identifies the latent blocks $\boldsymbol{z}_B$ up to permutation and block-wise transformations (Theorems 7.1 & 7.2). We believe these results will be of interest to both the OCRL community, as they partly explain the empirical success of these approaches, and to the nonlinear ICA and disentanglement community, as it provides an important special case where identifiability holds. This result relies on the block-specific decoders being "sufficiently nonlinear" (Assumption 7.2) and requires only very weak assumptions on the distribution of the ground-truth latent factors of variations. In particular, these factors can be statistically dependent and their support can be (almost) arbitrary.

Our second contribution is to show theoretically that additive decoders can generate images never seen during training by recombining observed factors of variations in novel ways (Corollary 7.1). To describe this ability, we coin the term "Cartesian-product extrapolation" (right of Figure 7.1). We believe the type of identifiability analysis laid out in this work to understand "out-of-support" generation is novel and could be applied to other function classes or learning algorithms such as DALLE-2 [Ramesh et al., 2022] and Stable Diffusion [Rombach et al., 2022] to understand their apparent creativity and hopefully improve it.

Both latent variables identification and Cartesian-product extrapolation are validated experimentally on simulated data (Section 7.4). More specifically, we observe that additivity is crucial for both by comparing against a non-additive decoder which fails to disentangle and extrapolate.

Notation. Scalars are denoted in lower-case and vectors in lower-case bold, e.g. $x \in \mathbb{R}$ and $\boldsymbol{x} \in \mathbb{R}^n$. We maintain an analogous notation for scalar-valued and vector-valued functions, e.g. f and $\boldsymbol{f}$. The ith coordinate of the vector $\boldsymbol{x}$ is denoted by $\boldsymbol{x}_i$. The set containing the first n integers excluding 0 is denoted by $[n]$. Given a subset of indices $S \subseteq [n]$, $\boldsymbol{x}_S$ denotes the subvector consisting of entries $\boldsymbol{x}_i$ for $i \in S$. Given a function $\boldsymbol{f}(\boldsymbol{x}_S) \in \mathbb{R}^m$ with input $\boldsymbol{x}_S$, the derivative of $\boldsymbol{f}$ w.r.t. $\boldsymbol{x}_i$ is denoted by $D_i \boldsymbol{f}(\boldsymbol{x}_S) \in \mathbb{R}^m$ and the second derivative w.r.t. $\boldsymbol{x}_i$ and $\boldsymbol{x}_{i'}$ is $D^2_{i,i'} \boldsymbol{f}(\boldsymbol{x}_S) \in \mathbb{R}^m$. See Table 7.2 in appendix for more.

Code: Our code repository can be found at this link.

7.2. Background & Literature review

Identifiability of latent variable models. The problem of latent variables identification can be best explained with a simple example. Suppose observations $\boldsymbol{x} \in \mathbb{R}^{d_x}$ are generated i.i.d. by first sampling a latent vector $\boldsymbol{z} \in \mathbb{R}^{d_z}$ from a distribution $\mathbb{P}_{\boldsymbol{z}}$ and feeding it into a decoder function $\boldsymbol{f} : \mathbb{R}^{d_z} \to \mathbb{R}^{d_x}$, i.e. $\boldsymbol{x} = \boldsymbol{f}(\boldsymbol{z})$. By choosing an alternative model defined as $\hat{\boldsymbol{f}} := \boldsymbol{f} \circ \boldsymbol{v}$ and $\hat{\boldsymbol{z}} := \boldsymbol{v}^{-1}(\boldsymbol{z})$ where $\boldsymbol{v} : \mathbb{R}^{d_z} \to \mathbb{R}^{d_z}$ is some bijective transformation, it is easy to see that the distributions of $\hat{\boldsymbol{x}} = \hat{\boldsymbol{f}}(\hat{\boldsymbol{z}})$ and $\boldsymbol{x}$ are the same since $\hat{\boldsymbol{f}}(\hat{\boldsymbol{z}}) = \boldsymbol{f} \circ \boldsymbol{v}(\boldsymbol{v}^{-1}(\boldsymbol{z})) = \boldsymbol{f}(\boldsymbol{z})$. The problem of identifiability is that, given only the distribution over $\boldsymbol{x}$, it is impossible to distinguish between the two models $(\boldsymbol{f}, \boldsymbol{z})$ and $(\hat{\boldsymbol{f}}, \hat{\boldsymbol{z}})$. This is problematic when one wants to discover interpretable

factors of variations since $\boldsymbol{z}$ and $\hat{\boldsymbol{z}}$ could be drastically different. There are essentially two strategies to go around this problem: (i) restricting the hypothesis class of decoders $\hat{\boldsymbol{f}}$ [Taleb and Jutten, 1999, Gresele et al., 2021, Leeb et al., 2021, Moran et al., 2022, Buchholz et al., 2022, Zheng et al., 2022], and/or (ii) restricting/adding structure to the distribution of $\hat{\boldsymbol{z}}$ [Hyvärinen et al., 2019, Locatello et al., 2020b, Lachapelle et al., 2022, Lippe et al., 2022]. By doing so, the hope is that the only bijective mappings $\boldsymbol{v}$ keeping $\hat{\boldsymbol{f}}$ and $\hat{\boldsymbol{z}}$ into their respective hypothesis classes will be trivial indeterminacies such as permutations and element-wise rescalings. Our contribution, which is to restrict the decoder function $\hat{\boldsymbol{f}}$ to be additive (Definition 7.1), falls into the first category. Other restricted function classes for $\boldsymbol{f}$ proposed in the literature include post-nonlinear mixtures [Taleb and Jutten, 1999], local isometries [Donoho and Grimes, 2003b,a, Horan et al., 2021a], conformal and orthogonal maps [Gresele et al., 2021, Reizinger et al., 2022, Buchholz et al., 2022] as well as various restrictions on the sparsity of $\boldsymbol{f}$ [Moran et al., 2022, Zheng et al., 2022, Brady et al., 2023, Xi and Bloem-Reddy, 2023]. Methods that do not restrict the decoder must instead restrict/structure the distribution of the latent factors by assuming, e.g., sparse temporal dependencies [Hyvarinen and Morioka, 2017, Klindt et al., 2021, Lachapelle et al., 2022, Lachapelle and Lacoste-Julien, 2022], conditionally independent latent variables given an observed auxiliary variable [Hyvärinen et al., 2019, Khemakhem et al., 2020a], that interventions targeting the latent factors are observed [Lachapelle et al., 2022, Lippe et al., 2022, 2023b, Brehmer et al., 2022, Ahuja et al., 2022b, 2023, Squires et al., 2023, Buchholz et al., 2023, von Kügelgen et al., 2023, Zhang et al., 2023, Jiang and Aragam, 2023], or that the support of the latents is a Cartesian-product [Wang and Jordan, 2022, Roth et al., 2023]. In contrast, our result makes very mild assumptions about the distribution of the latent factors, which can present statistical dependencies, have an almost arbitrarily shaped support and does not require any interventions. Additionally, none of these works provide extrapolation guarantees as we do in Section 7.3.2.

Relation to nonlinear ICA. Hyvärinen and Pajunen [1999] showed that the standard nonlinear ICA problem where the decoder $\boldsymbol{f}$ is nonlinear and the latent factors $\boldsymbol{z}_i$ are *statistically independent* is unidentifiable. This motivated various extensions of nonlinear ICA where more structure on the factors is assumed [Hyvarinen and Morioka, 2016, 2017, Hyvärinen et al., 2019, Khemakhem et al., 2020a,b, Hälvä et al., 2021]. Our approach departs from the standard nonlinear ICA problem along three axes: (i) we restrict the mixing function to be additive, (ii) the factors do not have to be necessarily independent, and (iii) we can identify only the blocks $\boldsymbol{z}_B$ as opposed to each $\boldsymbol{z}_i$ individually up to element-wise transformations, unless $\mathcal{B} = \{\{1\}, ..., \{d_z\}\}$ (see Section 7.3.1).

Object-centric representation learning (OCRL). Lin et al. [2020] classified OCRL methods in two categories: *scene mixture models* [Greff et al., 2016, 2017, 2019, Locatello et al., 2020c] & *spatial-attention models* [Eslami et al., 2016, Crawford and Pineau, 2019, Burgess et al., 2019, Engelcke et al., 2020]. Additive decoders can be seen as an approximation to the decoding

architectures used in the former category, which typically consist of an object-specific decoder $\boldsymbol{f}^{(\text{obj})}$ acting on object-specific latent blocks $\boldsymbol{z}_B$ and "mixed" together via a masking mechanism $\boldsymbol{m}^{(B)}(\boldsymbol{z})$ which selects which pixel belongs to which object. More precisely,

$$\boldsymbol{f}(\boldsymbol{z}) = \sum_{B \in \mathcal{B}} \boldsymbol{m}^{(B)}(\boldsymbol{z}) \odot \boldsymbol{f}^{(\text{obj})}(\boldsymbol{z}_B) \text{ , where } \boldsymbol{m}_k^{(B)}(\boldsymbol{z}) = \frac{\exp(\boldsymbol{a}_k(\boldsymbol{z}_B))}{\sum_{B' \in \mathcal{B}} \exp(\boldsymbol{a}_k(\boldsymbol{z}_{B'}))}, \tag{7.1}$$

and where $\mathcal{B}$ is a partition of $[d_z]$ made of equal-size blocks B and $\boldsymbol{a} : \mathbb{R}^{|B|} \to \mathbb{R}^{d_x}$ outputs a score that is normalized via a softmax operation to obtain the masks $\boldsymbol{m}^{(B)}(\boldsymbol{z})$. Many of these works also present some mechanism to select dynamically how many objects are present in the scene and thus have a variable-size representation $\boldsymbol{z}$, an important technical aspect we omit in our analysis. Empirically, training these decoders based on some form of reconstruction objective, probabilistic or not, yields latent blocks $\boldsymbol{z}_B$ that represent the information of individual objects separately. We believe our work constitutes a step towards providing a mathematically grounded explanation for why these approaches can perform this form of disentanglement without supervision (Theorems 7.1 & 7.2). Many architectural innovations in scene mixture models concern the encoder, but our analysis focuses solely on the structure of the decoder $\boldsymbol{f}(\boldsymbol{z})$, which is a shared aspect across multiple methods. Generalization capabilities of object-centric representations were studied empirically by Dittadi et al. [2022] but did not cover Cartesian-product extrapolation (Corollary 7.1) on which we focus here.

Diagonal Hessian penalty Peebles et al. [2020]. Additive decoders are also closely related to the penalty introduced by Peebles et al. [2020] which consists in regularizing the Hessian of the decoder to be diagonal. In Appendix A.2, we show that "additivity" and "diagonal Hessian" are equivalent properties. They showed empirically that this penalty can induce disentanglement on datasets such as CLEVR [Johnson et al., 2016], which is a standard benchmark for OCRL, but did not provide any formal justification. Our work provides a rigorous explanation for these successes and highlights the link between the diagonal Hessian penalty and OCRL.

Compositional decoders [Brady et al., 2023]. Compositional decoders were recently introduced by Brady et al. [2023] as a model for OCRL methods with identifiability guarantees. A decoder $\boldsymbol{f}$ is said to be *compositional* when its Jacobian $D\boldsymbol{f}$ satisfies the following property everywhere: For all $i \in [d_z]$ and $B \in \mathcal{B}$, $D_B \boldsymbol{f}_i(\boldsymbol{z}) \neq \boldsymbol{0} \implies D_{B^c} \boldsymbol{f}_i(\boldsymbol{z}) = \boldsymbol{0}$, where $B^c := [d_z] \setminus B$. In other words, each $\boldsymbol{x}_i$ can *locally* depend solely on one block $\boldsymbol{z}_B$ (this block can change for different $\boldsymbol{z}$). In Appendix A.3, we show that compositional C^2 decoders are additive. Furthermore, Example 7.3 shows a decoder that is additive but not compositional, which means that additive C^2 decoders are strictly more expressive than compositional C^2 decoders. Another important distinction with our work is that we consider more general supports for $\boldsymbol{z}$ and provide a novel extrapolation analysis. That being said, our identifiability result does not supersede theirs since they assume only C^1 decoders while our theory assumes C^2.

Extrapolation. Du and Mordatch [2019] studied empirically how one can combine energy-based models for what they call *compositional generalization*, which is similar to our notion of Cartesian-product extrapolation, but suppose access to datasets in which only one latent factor varies and do not provide any theory. Webb et al. [2020] studied extrapolation empirically and proposed a novel benchmark which does not have an additive structure. Besserve et al. [2021] proposed a theoretical framework in which out-of-distribution samples are obtained by applying a transformation to a single hidden layer inside the decoder network. Krueger et al. [2021b] introduced a domain generalization method which is trained to be robust to tasks falling outside the convex hull of training distributions. Extrapolation in text-conditioned image generation was recently discussed by Wang et al. [2023].

7.3. Additive decoders for disentanglement & extrapolation

Our theoretical results assume the existence of some data-generating process describing how the observations $\boldsymbol{x}$ are generated and, importantly, what are the "natural" factors of variations.

Assumption 7.1 (Data-generating process)**.** *The set of possible observations is given by a lower dimensional manifold $\boldsymbol{f}(\mathcal{Z}^{\text{test}})$ embedded in $\mathbb{R}^{d_x}$ where $\mathcal{Z}^{\text{test}}$ is an open set of $\mathbb{R}^{d_z}$ and $\boldsymbol{f} : \mathcal{Z}^{\text{test}} \to \mathbb{R}^{d_x}$ is a C^2-diffeomorphism onto its image. We will refer to $\boldsymbol{f}$ as the* ground-truth decoder*. At training time, the observations are i.i.d. samples given by $\boldsymbol{x} = \boldsymbol{f}(\boldsymbol{z})$ where $\boldsymbol{z}$ is distributed according to the probability measure $\mathbb{P}_{\boldsymbol{z}}^{\text{train}}$ with support $\mathcal{Z}^{\text{train}} \subseteq \mathcal{Z}^{\text{test}}$. Throughout, we assume that $\mathcal{Z}^{\text{train}}$ is regularly closed (Definition 7.6).*

Intuitively, the ground-truth decoder $\boldsymbol{f}$ is effectively relating the "natural factors of variations" $\boldsymbol{z}$ to the observations $\boldsymbol{x}$ in a one-to-one fashion. The map $\boldsymbol{f}$ is a C^2-diffeomorphism onto its image, which means that it is C^2 (has continuous second derivative) and that its inverse (restricted to the image of $\boldsymbol{f}$) is also C^2. Analogous assumptions are very common in the literature on nonlinear ICA and disentanglement [Hyvärinen et al., 2019, Khemakhem et al., 2020a, Lachapelle et al., 2022, Ahuja et al., 2022a]. Mansouri et al. [2022] pointed out that the injectivity of $\boldsymbol{f}$ is violated when images show two objects that are indistinguishable, an important practical case that is not covered by our theory.

We emphasize the distinction between $\mathcal{Z}^{\text{train}}$, which corresponds to the observations seen during training, and $\mathcal{Z}^{\text{test}}$, which corresponds to the set of all possible images. The case where $\mathcal{Z}^{\text{train}} \neq \mathcal{Z}^{\text{test}}$ will be of particular interest when discussing extrapolation in Section 7.3.2. The "regularly closed" condition on $\mathcal{Z}^{\text{train}}$ is mild, as it is satisfied as soon as the distribution of $\boldsymbol{z}$ has a density w.r.t. the Lebesgue measure on $\mathbb{R}^{d_z}$. It is violated, for example, when $\boldsymbol{z}$ is a discrete random vector. Figure 7.2 illustrates this assumption with simple examples.

Objective. Our analysis is based on the simple objective of reconstructing the observations $\boldsymbol{x}$ by learning an encoder $\hat{\boldsymbol{g}} : \mathbb{R}^{d_x} \to \mathbb{R}^{d_z}$ and a decoder $\hat{\boldsymbol{f}} : \mathbb{R}^{d_z} \to \mathbb{R}^{d_x}$. Note that we assumed implicitly

that the dimensionality of the learned representation matches the dimensionality of the ground-truth. We define the set of latent codes the encoder can output when evaluated on the training distribution:

$$\hat{\mathcal{Z}}^{\text{train}} := \hat{\boldsymbol{g}}(\boldsymbol{f}(\mathcal{Z}^{\text{train}}))\,. \tag{7.2}$$

When the images of the ground-truth and learned decoders match, i.e. $\boldsymbol{f}(\mathcal{Z}^{\text{train}}) = \hat{\boldsymbol{f}}(\hat{\mathcal{Z}}^{\text{train}})$, which happens when the reconstruction task is solved exactly, one can define the map $\boldsymbol{v} : \hat{\mathcal{Z}}^{\text{train}} \to \mathcal{Z}^{\text{train}}$ as

$$\boldsymbol{v} := \boldsymbol{f}^{-1} \circ \hat{\boldsymbol{f}}\,. \tag{7.3}$$

This function is going to be crucial throughout the work, especially to define $\mathcal{B}$-disentanglement (Definition 7.3), as it relates the learned representation to the ground-truth representation.

Before introducing our formal definition of additive decoders, we introduce the following notation: Given a set $\mathcal{Z} \subseteq \mathbb{R}^{d_z}$ and a subset of indices $B \subseteq [d_z]$, let us define $\mathcal{Z}_B$ to be the projection of $\mathcal{Z}$ onto dimensions labelled by the index set B. More formally,

$$\mathcal{Z}_B := \{\boldsymbol{z}_B \mid \boldsymbol{z} \in \mathcal{Z}\} \subseteq \mathbb{R}^{|B|}\,. \tag{7.4}$$

Intuitively, we will say that a decoder is *additive* when its output is the summation of the outputs of "object-specific" decoders that depend only on each latent block $\boldsymbol{z}_B$. This captures the idea that an image can be seen as the juxatoposition of multiple images which individually correspond to objects in the scene or natural factors of variations (left of Figure 7.1).

Definition 7.1 (Additive functions)**.** *Let $\mathcal{B}$ be a partition of $[d_z]$*[1]*. A function $\boldsymbol{f} : \mathcal{Z} \to \mathbb{R}^{d_x}$ is said to be **additive** if there exist functions $\boldsymbol{f}^{(B)} : \mathcal{Z}_B \to \mathbb{R}^{d_x}$ for all $B \in \mathcal{B}$ such that*

$$\forall \boldsymbol{z} \in \mathcal{Z}, \boldsymbol{f}(\boldsymbol{z}) = \sum_{B \in \mathcal{B}} \boldsymbol{f}^{(B)}(\boldsymbol{z}_B)\,. \tag{7.5}$$

This additivity property will be central to our analysis as it will be the driving force of identifiability (Theorem 7.1 & 7.2) and Cartesian-product extrapolation (Corollary 7.1).

Remark 7.1. *Suppose we have $\boldsymbol{x} = \sigma(\sum_{B \in \mathcal{B}} \boldsymbol{f}^{(B)}(\boldsymbol{z}_B))$ where σ is a known bijective function. For example, if $\sigma(\boldsymbol{y}) := \exp(\boldsymbol{y})$ (component-wise), the decoder can be thought of as being multiplicative. Our results still apply since we can simply transform the data doing $\tilde{\boldsymbol{x}} := \sigma^{-1}(\boldsymbol{x})$ to recover the additive form $\tilde{\boldsymbol{x}} = \sum_{B \in \mathcal{B}} \boldsymbol{f}^{(B)}(\boldsymbol{z}_B)$.*

Differences with OCRL in practice. We point out that, although the additive decoders make intuitive sense for OCRL, they are not expressive enough to represent the "masked decoders" typically used in practice (Equation (7.1)). The lack of additivity stems from the normalization in the masks $\boldsymbol{m}^{(B)}(\boldsymbol{z})$. We hypothesize that studying the simpler additive decoders might still reveal interesting phenomena present in modern OCRL approaches due to their resemblance. Another difference is that, in practice, the same object-specific decoder $\boldsymbol{f}^{(\text{obj})}$ is applied to every latent block

[1]Without loss of generality, we assume that the partition $\mathcal{B}$ is contiguous, i.e. each $B \in \mathcal{B}$ can be written as $B = \{i+1, i+2, \ldots, i+|B|\}$.

$\boldsymbol{z}_B$. Our theory allows for these functions to be different, but also applies when functions are the same. Additionally, this parameter sharing across $\boldsymbol{f}^{(B)}$ enables modern methods to have a variable number of objects across samples, an important practical point our theory does not cover.

7.3.1. Identifiability analysis

We now study the identifiability of additive decoders and show how they can yield disentanglement. Our definition of disentanglement will rely on *partition-respecting permutations*:

Definition 7.2 (Partition-respecting permutations)**.** *Let $\mathcal{B}$ be a partition of $\{1, ..., d_z\}$. A permutation π over $\{1, ..., d_z\}$ respects $\mathcal{B}$ if, for all $B \in \mathcal{B}$, $\pi(B) \in \mathcal{B}$.*

Essentially, a permutation that respects $\mathcal{B}$ is one which can permute blocks of $\mathcal{B}$ and permute elements within a block, but cannot "mix" blocks together. We now introduce $\mathcal{B}$-disentanglement.

Definition 7.3 ($\mathcal{B}$-disentanglement)**.** *A learned decoder $\hat{\boldsymbol{f}} : \mathbb{R}^{d_z} \to \mathbb{R}^{d_x}$ is said to be **$\mathcal{B}$-disentangled** w.r.t. the ground-truth decoder $\boldsymbol{f}$ when $\boldsymbol{f}(\mathcal{Z}^{\text{train}}) = \hat{\boldsymbol{f}}(\hat{\mathcal{Z}}^{\text{train}})$ and the mapping $\boldsymbol{v} := \boldsymbol{f}^{-1} \circ \hat{\boldsymbol{f}}$ is a diffeomorphism from $\hat{\mathcal{Z}}^{\text{train}}$ to $\mathcal{Z}^{\text{train}}$ satisfying the following property: there exists a permutation π respecting $\mathcal{B}$ such that, for all $B \in \mathcal{B}$, there exists a function $\bar{\boldsymbol{v}}_{\pi(B)} : \hat{\mathcal{Z}}^{\text{train}}_B \to \mathcal{Z}^{\text{train}}_{\pi(B)}$ such that, for all $\boldsymbol{z} \in \hat{\mathcal{Z}}^{\text{train}}$, $\boldsymbol{v}_{\pi(B)}(\boldsymbol{z}) = \bar{\boldsymbol{v}}_{\pi(B)}(\boldsymbol{z}_B)$. In other words, $\boldsymbol{v}_{\pi(B)}(\boldsymbol{z})$ depends only on $\boldsymbol{z}_B$.*

Thus, $\mathcal{B}$-disentanglement means that the blocks of latent dimensions $\boldsymbol{z}_B$ are disentangled from one another, but that variables within a given block might remain entangled. Note that, unless the partition is $\mathcal{B} = \{\{1\}, \ldots, \{d_z\}\}$, this corresponds to a weaker form of disentanglement than what is typically seeked in nonlinear ICA, i.e. recovering each variable individually.

Example 7.1. *To illustrate $\mathcal{B}$-disentanglement, imagine a scene consisting of two balls moving around in 2D where the "ground-truth" representation is given by $\boldsymbol{z} = (x^1, y^1, x^2, y^2)$ where $\boldsymbol{z}_{B_1} = (x^1, y^1)$ and $\boldsymbol{z}_{B_2} = (x^2, y^2)$ are the coordinates of each ball (here, $\mathcal{B} := \{\{1, 2\}, \{3, 4\}\}$). In that case, a learned representation is $\mathcal{B}$-disentangled when the balls are disentangled from one another. However, the basis in which the position of each ball is represented might differ in both representations.*

Our first result (Theorem 7.1) shows a weaker form of disentanglement we call *local* $\mathcal{B}$-disentanglement. This means the Jacobian matrix of $\boldsymbol{v}$, $D\boldsymbol{v}$, has a "block-permutation" structure everywhere.

Definition 7.4 (Local $\mathcal{B}$-disentanglement)**.** *A learned decoder $\hat{\boldsymbol{f}} : \mathbb{R}^{d_z} \to \mathbb{R}^{d_x}$ is said to be **locally $\mathcal{B}$-disentangled** w.r.t. the ground-truth decoder $\boldsymbol{f}$ when $\boldsymbol{f}(\mathcal{Z}^{\text{train}}) = \hat{\boldsymbol{f}}(\hat{\mathcal{Z}}^{\text{train}})$ and the mapping $\boldsymbol{v} := \boldsymbol{f}^{-1} \circ \hat{\boldsymbol{f}}$ is a diffeomorphism from $\hat{\mathcal{Z}}^{\text{train}}$ to $\mathcal{Z}^{\text{train}}$ with a mapping $\boldsymbol{v} : \hat{\mathcal{Z}}^{\text{train}} \to \mathcal{Z}^{\text{train}}$ satisfying the following property: for all $\boldsymbol{z} \in \hat{\mathcal{Z}}^{\text{train}}$, there exists a permutation π respecting $\mathcal{B}$ such that, for all $B \in \mathcal{B}$, the columns of $D\boldsymbol{v}_{\pi(B)}(\boldsymbol{z}) \in \mathbb{R}^{|B| \times d_z}$ outside block B are zero.*

In Appendix A.4, we provide three examples where local disentanglement holds but not global disentanglement. The first one illustrates how having a disconnected support can allow for a

permutation π (from Definition 7.4) that changes between disconnected regions of the support. The last two examples show how, even if the permutation stays the same throughout the support, we can still violate global disentanglement, even with a connected support.

We now state the main identifiability result of this work which provides conditions to guarantee *local* disentanglement. We will then see how to go from local to *global* disentanglement in the subsequent Theorem 7.2. For pedagogical reasons, we delay the formalization of the sufficient nonlinearity Assumption 7.2 on which the result crucially relies.

Theorem 7.1 (Local disentanglement via additive decoders)**.** *Suppose that the data-generating process satisfies Assumption 7.1, that the learned decoder $\hat{\boldsymbol{f}} : \mathbb{R}^{d_z} \to \mathbb{R}^{d_x}$ is a C^2-diffeomorphism, that the encoder $\hat{\boldsymbol{g}} : \mathbb{R}^{d_x} \to \mathbb{R}^{d_z}$ is continuous, that both $\boldsymbol{f}$ and $\hat{\boldsymbol{f}}$ are additive (Definition 7.1) and that $\boldsymbol{f}$ is sufficiently nonlinear as formalized by Assumption 7.2. Then, if $\hat{\boldsymbol{f}}$ and $\hat{\boldsymbol{g}}$ solve the reconstruction problem on the training distribution, i.e. $\mathbb{E}^{\text{train}}||\boldsymbol{x} - \hat{\boldsymbol{f}}(\hat{\boldsymbol{g}}(\boldsymbol{x}))||^2 = 0$, we have that $\hat{\boldsymbol{f}}$ is locally $\mathcal{B}$-disentangled w.r.t. $\boldsymbol{f}$ (Definition 7.4).*

The proof of Theorem 7.1, which can be found in Appendix A.5, is inspired from Hyvärinen et al. [2019]. The essential differences are that (i) they leverage the additivity of the conditional log-density of $\boldsymbol{z}$ given an auxiliary variable $\boldsymbol{u}$ (i.e. conditional independence) instead of the additivity of the decoder function $\boldsymbol{f}$, (ii) we extend their proof techniques to allow for "block" disentanglement, i.e. when $\mathcal{B}$ is not the trivial partition $\{\{1\}, \dots, \{d_z\}\}$, (iii) the asssumption "sufficient variability" of the prior $p(\boldsymbol{z} \mid \boldsymbol{u})$ of Hyvärinen et al. [2019] is replaced by an analogous assumption of "sufficient nonlinearity" of the decoder $\boldsymbol{f}$ (Assumption 7.2), and (iv) we consider much more general supports $\mathcal{Z}^{\text{train}}$ which makes the jump from local to global disentanglement less direct in our case.

The identifiability-expressivity trade-off. The level of granularity of the partition $\mathcal{B}$ controls the trade-off between identifiability and expressivity: the finer the partition, the tighter the identifiability guarantee but the less expressive is the function class. The optimal level of granularity is going to depend on the application at hand. Whether $\mathcal{B}$ could be learned from data is left for future work.

Sufficient nonlinearity. The following assumption is key in proving Theorem 7.2, as it requires that the ground-truth decoder is "sufficiently nonlinear". This is reminiscent of the "sufficient variability" assumptions found in the nonlinear ICA litterature, which usually concerns the distribution of the latent variable $\boldsymbol{z}$ as opposed to the decoder $\boldsymbol{f}$ [Hyvarinen and Morioka, 2016, 2017, Hyvärinen et al., 2019, Khemakhem et al., 2020a,b, Lachapelle et al., 2022, Zheng et al., 2022]. We clarify this link in Appendix A.6 and provide intuitions why sufficient nonlinearity can be satisfied when $d_x \gg d_z$.

Assumption 7.2 (Sufficient nonlinearity of $\boldsymbol{f}$)**.** *Let* $q := d_z + \sum_{B\in\mathcal{B}} \frac{|B|(|B|+1)}{2}$. *For all* $\boldsymbol{z} \in \mathcal{Z}^{\text{train}}$, $\boldsymbol{f}$ *is such that the following matrix has linearly independent columns (i.e. full column-rank):*

$$\boldsymbol{W}(\boldsymbol{z}) := \left[\left[D_i \boldsymbol{f}^{(B)}(\boldsymbol{z}_B) \right]_{i\in B} \left[D^2_{i,i'} \boldsymbol{f}^{(B)}(\boldsymbol{z}_B) \right]_{(i,i')\in B^2_{\leq}} \right]_{B\in\mathcal{B}} \in \mathbb{R}^{d_x \times q}, \tag{7.6}$$

where $B^2_{\leq} := B^2 \cap \{(i,i') \mid i' \leq i\}$. *Note this implies* $d_x \geq q$.

The following example shows that Theorem 7.1 does not apply if the ground-truth decoder $\boldsymbol{f}$ is linear. If that was the case, it would contradict the well known fact that linear ICA with independent Gaussian factors is unidentifiable.

Example 7.2 (Importance of Assumption 7.2)**.** *Suppose* $\boldsymbol{x} = \boldsymbol{f}(\boldsymbol{z}) = \boldsymbol{A}\boldsymbol{z}$ *where* $\boldsymbol{A} \in \mathbb{R}^{d_x \times d_z}$ *is full rank. Take* $\hat{\boldsymbol{f}}(\boldsymbol{z}) := \boldsymbol{A}\boldsymbol{V}\boldsymbol{z}$ *and* $\hat{\boldsymbol{g}}(\boldsymbol{x}) := \boldsymbol{V}^{-1}\boldsymbol{A}^{\dagger}\boldsymbol{x}$ *where* $\boldsymbol{V} \in \mathbb{R}^{d_z \times d_z}$ *is invertible and* $\boldsymbol{A}^{\dagger}$ *is the left pseudo inverse of* $\boldsymbol{A}$. *By construction, we have that* $\mathbb{E}[\boldsymbol{x} - \hat{\boldsymbol{f}}(\hat{\boldsymbol{g}}(\boldsymbol{x}))] = 0$ *and* $\boldsymbol{f}$ *and* $\hat{\boldsymbol{f}}$ *are* $\mathcal{B}$*-additive because* $\boldsymbol{f}(\boldsymbol{z}) = \sum_{B\in\mathcal{B}} \boldsymbol{A}_{\cdot,B}\boldsymbol{z}_B$ *and* $\hat{\boldsymbol{f}}(\boldsymbol{z}) = \sum_{B\in\mathcal{B}} (\boldsymbol{A}\boldsymbol{V})_{\cdot,B}\boldsymbol{z}_B$. *However, we still have that* $\boldsymbol{v}(\boldsymbol{z}) := \boldsymbol{f}^{-1} \circ \hat{\boldsymbol{f}}(\boldsymbol{z}) = \boldsymbol{V}\boldsymbol{z}$ *where* $\boldsymbol{V}$ *does not necessarily have a block-permutation structure, i.e. no disentanglement. The reason we cannot apply Theorem 7.1 here is because Assumption 7.2 is not satisfied. Indeed, the second derivatives of* $\boldsymbol{f}^{(B)}(\boldsymbol{z}_B) := \boldsymbol{A}_{\cdot,B}\boldsymbol{z}_B$ *are all zero and hence* $\boldsymbol{W}(\boldsymbol{z})$ *cannot have full column-rank.*

Example 7.3 (A sufficiently nonlinear $\boldsymbol{f}$)**.** *In Appendix A.7 we show numerically that the function*

$$\boldsymbol{f}(\boldsymbol{z}) := [\boldsymbol{z}_1, \boldsymbol{z}_1^2, \boldsymbol{z}_1^3, \boldsymbol{z}_1^4]^{\top} + [(\boldsymbol{z}_2 + 1), (\boldsymbol{z}_2 + 1)^2, (\boldsymbol{z}_2 + 1)^3, (\boldsymbol{z}_2 + 1)^4]^{\top} \tag{7.7}$$

is a diffeomorphism from the square $[-1, 0] \times [0, 1]$ *to its image that satisfies Assumption 7.2.*

Example 7.4 (Smooth balls dataset is sufficiently nonlinear)**.** *In Appendix A.7 we present a simple synthetic dataset consisting of images of two colored balls moving up and down. We also verify numerically that its underlying ground-truth decoder* $\boldsymbol{f}$ *is sufficiently nonlinear.*

7.3.1.1. From local to global disentanglement. The following result provides additional assumptions to guarantee *global* disentanglement (Definition 7.3) as opposed to only local disentanglement (Definition 7.4). See Appendix A.8 for its proof.

Theorem 7.2 (From local to global disentanglement)**.** *Suppose that all the assumptions of Theorem 7.1 hold. Additionally, assume* $\mathcal{Z}^{\text{train}}$ *is path-connected (Definition 7.8) and that the block-specific decoders* $\boldsymbol{f}^{(B)}$ *and* $\hat{\boldsymbol{f}}^{(B)}$ *are injective for all blocks* $B \in \mathcal{B}$. *Then, if* $\hat{\boldsymbol{f}}$ *and* $\hat{\boldsymbol{g}}$ *solve the reconstruction problem on the training distribution, i.e.* $\mathbb{E}^{\text{train}}||\boldsymbol{x} - \hat{\boldsymbol{f}}(\hat{\boldsymbol{g}}(\boldsymbol{x}))||^2 = 0$, *we have that* $\hat{\boldsymbol{f}}$ *is (globally)* $\mathcal{B}$*-disentangled w.r.t.* $\boldsymbol{f}$ *(Definition 7.3) and, for all* $B \in \mathcal{B}$,

$$\hat{\boldsymbol{f}}^{(B)}(\boldsymbol{z}_B) = \boldsymbol{f}^{(\pi(B))}(\bar{\boldsymbol{v}}_{\pi(B)}(\boldsymbol{z}_B)) + \boldsymbol{c}^{(B)}, \text{for all } \boldsymbol{z}_B \in \hat{\mathcal{Z}}^{\text{train}}_B, \tag{7.8}$$

where the functions $\bar{\boldsymbol{v}}_{\pi(B)}$ *are from Defintion 7.3 and the vectors* $\boldsymbol{c}^{(B)} \in \mathbb{R}^{d_x}$ *are constants such that* $\sum_{B\in\mathcal{B}} \boldsymbol{c}^{(B)} = 0$. *We also have that the functions* $\bar{\boldsymbol{v}}_{\pi(B)} : \hat{\mathcal{Z}}^{\text{train}}_B \to \mathcal{Z}^{\text{train}}_{\pi(B)}$ *are* C^2*-diffeomorphisms*

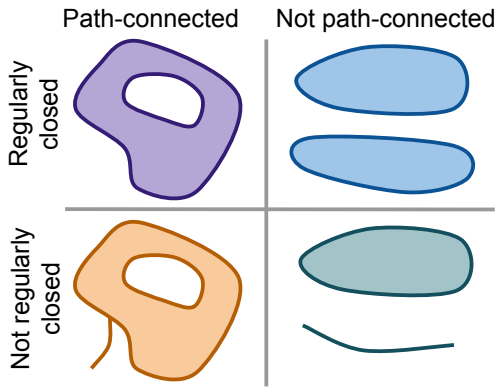


Figure 7.2. Illustrating regularly closed sets (Definition 7.6) and path-connected sets (Definition 7.8). Theorem 7.2 requires $\mathcal{Z}^{\text{train}}$ to satisfy both properties.

and have the following form:

$$\bar{\boldsymbol{v}}_{\pi(B)}(\boldsymbol{z}_B) = (\boldsymbol{f}^{\pi(B)})^{-1}(\hat{\boldsymbol{f}}^{(B)}(\boldsymbol{z}_B) - \boldsymbol{c}^{(B)}), \text{ for all } \boldsymbol{z}_B \in \hat{\mathcal{Z}}_B^{\text{train}} . \tag{7.9}$$

Equation (7.8) in the above result shows that each block-specific learned decoder $\hat{\boldsymbol{f}}^{(B)}$ is "imitating" a block-specific ground-truth decoder $\boldsymbol{f}^{\pi(B)}$. Indeed, the "object-specific" image outputted by the decoder $\hat{\boldsymbol{f}}^{(B)}$ evaluated at some $\boldsymbol{z}_B \in \hat{\mathcal{Z}}_B^{\text{train}}$ is the same as the image outputted by $\boldsymbol{f}^{(B)}$ evaluated at $\boldsymbol{v}(\boldsymbol{z}_B) \in \mathcal{Z}_B^{\text{train}}$, *up to an additive constant vector* $\boldsymbol{c}^{(B)}$. These constants cancel each other out when taking the sum of the block-specific decoders.

Equation (7.9) provides an explicit form for the function $\bar{\boldsymbol{v}}_{\pi(B)}$, which is essentially the learned block-specific decoder composed with the inverse of the ground-truth block-specific decoder.

Additional assumptions to go from local to global. Assuming that the support of $\mathbb{P}_{\boldsymbol{z}}^{\text{train}}$, $\mathcal{Z}^{\text{train}}$, is **path-connected** (see Definition 7.8 in appendix) is useful since it prevents the permutation π of Definition 7.4 from changing between two disconnected regions of $\hat{\mathcal{Z}}^{\text{train}}$. See Figure 7.2 for an illustration. In Appendix A.9, we discuss the additional assumption that each $\boldsymbol{f}^{(B)}$ must be injective and show that, in general, it is not equivalent to the assumption that $\sum_{B\in\mathcal{B}} \boldsymbol{f}^{(B)}$ is injective.

7.3.2. Cartesian-product extrapolation

In this section, we show how a learned additive decoder can be used to generate images $\boldsymbol{x}$ that are "out of support" in the sense that $\boldsymbol{x} \notin \boldsymbol{f}(\mathcal{Z}^{\text{train}})$, but that are still on the manifold of "reasonable" images, i.e. $\boldsymbol{x} \in \boldsymbol{f}(\mathcal{Z}^{\text{test}})$. To characterize the set of images the learned decoder can generate, we will rely on the notion of "cartesian-product extension", which we define next.

Definition 7.5 (Cartesian-product extension)**.** *Given a set* $\mathcal{Z} \subseteq \mathbb{R}^{d_z}$ *and partition* $\mathcal{B}$ *of* $[d_z]$*, we define the Cartesian-product extension of* $\mathcal{Z}$ *as*

$$\text{CPE}_{\mathcal{B}}(\mathcal{Z}) := \prod_{B\in\mathcal{B}} \mathcal{Z}_B \text{ , where } \mathcal{Z}_B := \{\boldsymbol{z}_B \mid \boldsymbol{z} \in \mathcal{Z}\}.$$

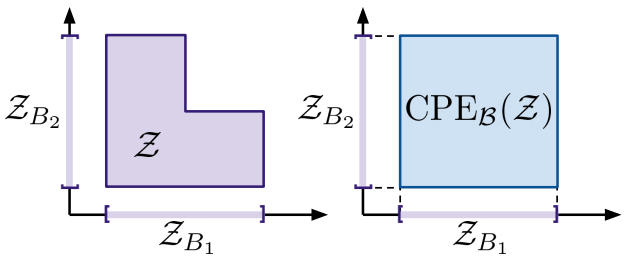


Figure 7.3. Illustration of Definition 7.5.

It is indeed an extension of $\mathcal{Z}$ since $\mathcal{Z} \subseteq \prod_{B\in\mathcal{B}} \mathcal{Z}_B$.

Let us define $\bar{\boldsymbol{v}} : \mathrm{CPE}_{\mathcal{B}}(\hat{\mathcal{Z}}^{\text{train}}) \to \mathrm{CPE}_{\mathcal{B}}(\mathcal{Z}^{\text{train}})$ to be the natural extension of the function $\boldsymbol{v} : \hat{\mathcal{Z}}^{\text{train}} \to \mathcal{Z}^{\text{train}}$. More explicitly, $\bar{\boldsymbol{v}}$ is the "concatenation" of the functions $\bar{\boldsymbol{v}}_B$ given in Definition 7.3:

$$\bar{\boldsymbol{v}}(\boldsymbol{z})^\top := [\bar{\boldsymbol{v}}_{B_1}(\boldsymbol{z}_{\pi^{-1}(B_1)})^\top \cdots \bar{\boldsymbol{v}}_{B_\ell}(\boldsymbol{z}_{\pi^{-1}(B_\ell)})^\top], \tag{7.10}$$

where ℓ is the number of blocks in $\mathcal{B}$. This map is a diffeomorphism because each $\bar{\boldsymbol{v}}_{\pi(B)}$ is a diffeomorphism from $\hat{\mathcal{Z}}_B^{\text{train}}$ to $\mathcal{Z}_{\pi(B)}^{\text{train}}$ by Theorem 7.2.

We already know that $\hat{\boldsymbol{f}}(\boldsymbol{z}) = \boldsymbol{f} \circ \bar{\boldsymbol{v}}(\boldsymbol{z})$ for all $\boldsymbol{z} \in \hat{\mathcal{Z}}^{\text{train}}$. The following result shows that this equality holds in fact on the larger set $\mathrm{CPE}_{\mathcal{B}}(\hat{\mathcal{Z}}^{\text{train}})$, the Cartesian-product extension of $\hat{\mathcal{Z}}^{\text{train}}$. See right of Figure 7.1 for an illustration of the following corollary.

Corollary 7.1 (Cartesian-product extrapolation)**.** *Suppose the assumptions of Theorem 7.2 holds. Then,*

$$\textit{for all } \boldsymbol{z} \in \mathrm{CPE}_{\mathcal{B}}(\hat{\mathcal{Z}}^{\text{train}}),\ \sum_{B\in\mathcal{B}} \hat{\boldsymbol{f}}^{(B)}(\boldsymbol{z}_B) = \sum_{B\in\mathcal{B}} \boldsymbol{f}^{(\pi(B))}(\bar{\boldsymbol{v}}_{\pi(B)}(\boldsymbol{z}_B))\,. \tag{7.11}$$

Furthermore, if $\mathrm{CPE}_{\mathcal{B}}(\mathcal{Z}^{\text{train}}) \subseteq \mathcal{Z}^{\text{test}}$, then $\hat{\boldsymbol{f}}(\mathrm{CPE}_{\mathcal{B}}(\hat{\mathcal{Z}}^{\text{train}})) \subseteq \boldsymbol{f}(\mathcal{Z}^{\text{test}})$.

Equation (7.11) tells us that the learned decoder $\hat{\boldsymbol{f}}$ "imitates" the ground-truth $\boldsymbol{f}$ not just over $\hat{\mathcal{Z}}^{\text{train}}$, but also over its Cartesian-product extension. This is important since it guarantees that we can generate observations never seen during training as follows: Choose a latent vector $\boldsymbol{z}^{\text{new}}$ that is in the Cartesian-product extension of $\hat{\mathcal{Z}}^{\text{train}}$, but not in $\hat{\mathcal{Z}}^{\text{train}}$ itself, i.e. $\boldsymbol{z}^{\text{new}} \in \mathrm{CPE}_{\mathcal{B}}(\hat{\mathcal{Z}}^{\text{train}}) \setminus \hat{\mathcal{Z}}^{\text{train}}$. Then, evaluate the learned decoder on $\boldsymbol{z}^{\text{new}}$ to get $\boldsymbol{x}^{\text{new}} := \hat{\boldsymbol{f}}(\boldsymbol{z}^{\text{new}})$. By Corollary 7.1, we know that $\boldsymbol{x}^{\text{new}} = \boldsymbol{f} \circ \bar{\boldsymbol{v}}(\boldsymbol{z}^{\text{new}})$, i.e. it is the observation one would have obtain by evaluating the ground-truth decoder $\boldsymbol{f}$ on the point $\bar{\boldsymbol{v}}(\boldsymbol{z}^{\text{new}}) \in \mathrm{CPE}_{\mathcal{B}}(\mathcal{Z}^{\text{train}})$. In addition, this $\boldsymbol{x}^{\text{new}}$ has never been seen during training since $\bar{\boldsymbol{v}}(\boldsymbol{z}^{\text{new}}) \notin \bar{\boldsymbol{v}}(\hat{\mathcal{Z}}^{\text{train}}) = \mathcal{Z}^{\text{train}}$. The experiment of Figure 7.4 illustrates this procedure.

About the extra assumption "$\mathrm{CPE}_{\mathcal{B}}(\mathcal{Z}^{\text{train}}) \subseteq \mathcal{Z}^{\text{test}}$". Recall that, in Assumption 7.1, we interpreted $\boldsymbol{f}(\mathcal{Z}^{\text{test}})$ to be the set of "reasonable" observations $\boldsymbol{x}$, of which we only observe a subset $\boldsymbol{f}(\mathcal{Z}^{\text{train}})$. Under this interpretation, $\mathcal{Z}^{\text{test}}$ is the set of reasonable values for the vector $\boldsymbol{z}$ and the additional assumption that $\mathrm{CPE}_{\mathcal{B}}(\mathcal{Z}^{\text{train}}) \subseteq \mathcal{Z}^{\text{test}}$ in Corollary 7.1 requires that the Cartesian-product extension of $\mathcal{Z}^{\text{train}}$ consists only of reasonable values of $\boldsymbol{z}$. From this assumption, we can easily

	ScalarLatents				BlockLatents (independent z)		BlockLatents (dependent z)	
Decoders	RMSE	$\text{LMS}_{\text{Spear}}$	RMSE^{OOS}	$\text{LMS}^{\text{OOS}}_{\text{Spear}}$	RMSE	LMS_{Tree}	RMSE	LMS_{Tree}
Non-add.	.06 $\pm$.002	70.6$\pm$5.21	.18$\pm$.012	73.7$\pm$4.64	.02$\pm$.001	53.9$\pm$7.58	.02$\pm$.001	78.1$\pm$2.92
Additive	.06$\pm$.002	**91.5$\pm$3.57**	.11$\pm$.018	**89.5$\pm$5.02**	.03$\pm$.012	**92.2$\pm$4.91**	.01$\pm$.002	**99.9$\pm$0.02**

Table 7.1. Reporting reconstruction mean squared error (RMSE $\downarrow$) and the Latent Matching Score (LMS $\uparrow$) for the three datasets considered: **ScalarLatents** and **BlockLatents** with independent and dependent latents. Runs were repeated with 10 random initializations. RMSE^{OOS} and $\text{LMS}^{\text{OOS}}_{\text{Spear}}$ are the same metric but evaluated out of support (see Appendix B.3 for details). While the standard error is high, the differences are still clear as can be seen in their box plot version in Appendix B.4.

conclude that $\hat{\boldsymbol{f}}(\text{CPE}_{\mathcal{B}}(\hat{\mathcal{Z}}^{\text{train}})) \subseteq \boldsymbol{f}(\mathcal{Z}^{\text{test}})$, which can be interpreted as: "The novel observations $\boldsymbol{x}^{\text{new}}$ obtained via Cartesian-product extrapolation are *reasonable*". Appendix A.11 describes an example where the assumption is violated, i.e. $\text{CPE}_{\mathcal{B}}(\mathcal{Z}^{\text{train}}) \not\subseteq \mathcal{Z}^{\text{test}}$. The practical implication of this is that the new observations $\boldsymbol{x}^{\text{new}}$ obtained via Cartesian-product extrapolation might not always be reasonable.

Disentanglement is not enough for extrapolation. To the best of our knowledge, Corollary 7.1 is the first result that formalizes how disentanglement can induce extrapolation. We believe it illustrates the fact that disentanglement alone is not sufficient to enable extrapolation and that one needs to restrict the hypothesis class of decoders in some way. Indeed, given a learned decoder $\hat{\boldsymbol{f}}$ that is disentangled w.r.t. $\boldsymbol{f}$ on the training support $\mathcal{Z}^{\text{train}}$, one cannot guarantee both decoders will "agree" outside the training domain without further restricting $\hat{\boldsymbol{f}}$ and $\boldsymbol{f}$. This work has focused on "additivity", but we believe other types of restriction could correspond to other types of extrapolation.

7.4. Experiments

We now present empirical validations of the theoretical results presented earlier. To achieve this, we compare the ability of additive and non-additive decoders to both identify ground-truth latent factors (Theorems 7.1 & 7.2) and extrapolate (Corollary 7.1) when trained to solve the reconstruction task on simple images ($64 \times 64 \times 3$) consisting of two balls moving in space [Ahuja et al., 2022b]. See Appendix B.1 for training details. We consider two datasets: one where the two ball positions can only vary along the y-axis (**ScalarLatents**) and one where the positions can vary along both the x and y axes (**BlockLatents**).

ScalarLatents: The ground-truth latent vector $\boldsymbol{z} \in \mathbb{R}^2$ is such that $\boldsymbol{z}_1$ and $\boldsymbol{z}_2$ corresponds to the height (y-coordinate) of the first and second ball, respectively. Thus the partition is simply $\mathcal{B} = \{\{1\}, \{2\}\}$ (each object has only one latent factor). This simple setting is interesting to study since the low dimensionality of the latent space ($d_z = 2$) allows for exhaustive visualizations like Figure 7.4. To study Cartesian-product extrapolation (Corollary 7.1), we sample $\boldsymbol{z}$ from a

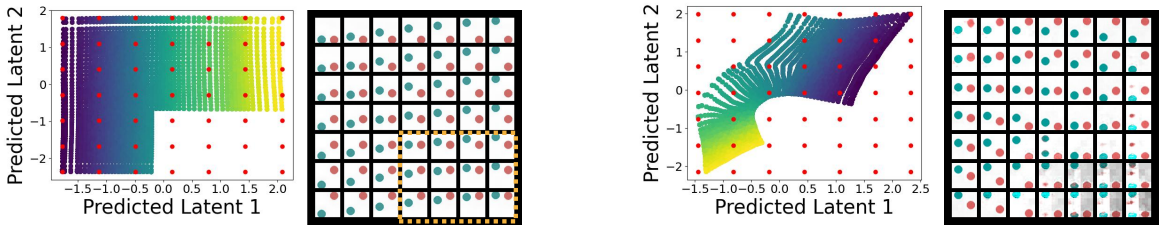
distribution with a L-shaped support given by $\mathcal{Z}^{\text{train}} := [0,1] \times [0,1] \setminus [0.5,1] \times [0.5,1]$, so that the training set does not contain images where both balls appear in the upper half of the image (see Appendix B.2).

BlockLatents: The ground-truth latent vector $\boldsymbol{z} \in \mathbb{R}^4$ is such that $\boldsymbol{z}_{\{1,2\}}$ and $\boldsymbol{z}_{\{3,4\}}$ correspond to the x, y position of the first and second ball, respectively (the partition is simply $\mathcal{B} = \{\{1,2\},\{3,4\}\}$, i.e. each object has two latent factors). Thus, this more challenging setting illustrates "block-disentanglement". The latent $\boldsymbol{z}$ is sampled uniformly from the hypercube $[0,1]^4$ but the images presenting occlusion (when a ball is behind another) are rejected from the dataset. We discuss how additive decoders cannot model images presenting occlusion in Appendix A.12. We also present an additional version of this dataset where we sample from the hypercube $[0,1]^4$ with dependencies. See Appendix B.2 for more details about data generation.

Evaluation metrics: To evaluate disentanglement, we compute a matrix of scores $(s_{B,B'}) \in \mathbb{R}^{\ell \times \ell}$ where ℓ is the number of blocks in $\mathcal{B}$ and $s_{B,B'}$ is a score measuring how well we can predict the ground-truth block $\boldsymbol{z}_B$ from the learned latent block $\hat{\boldsymbol{z}}_{B'} = \hat{\boldsymbol{g}}_{B'}(\boldsymbol{x})$ outputted by the encoder. The final Latent Matching Score (LMS) is computed as $\text{LMS} = \arg\max_{\pi \in \mathfrak{S}_{\mathcal{B}}} \frac{1}{\ell} \sum_{B \in \mathcal{B}} s_{B,\pi(B)}$, where $\mathfrak{S}_{\mathcal{B}}$ is the set of permutations respecting $\mathcal{B}$ (Definition 7.2). When $\mathcal{B} := \{\{1\},\dots,\{d_z\}\}$ and the score used is the absolute value of the correlation, LMS is simply the *mean correlation coefficient* (MCC), which is widely used in the nonlinear ICA literature [Hyvarinen and Morioka, 2016, 2017, Hyvärinen et al., 2019, Khemakhem et al., 2020a, Lachapelle et al., 2022]. Because our theory guarantees recovery of the latents only up to invertible and potentially nonlinear transformations, we use the Spearman correlation, which can capture nonlinear relationships unlike the Pearson correlation. We denote this score by $\text{LMS}_{\text{Spear}}$ and will use it in the dataset **ScalarLatents**. For the **BlockLatents** dataset, we cannot use Spearman correlation (because $\boldsymbol{z}_B$ are two dimensional). Instead, we take the score $s_{B,B'}$ to be the R^2 score of a regression tree. We denote this score by LMS_{tree}. There are subtleties to take care of when one wants to evaluate LMS_{tree} on a non-additive model due to the fact that the learned representation does not have a natural partition $\mathcal{B}$. We must thus search over partitions. We discuss this and provide further details on the metrics in Appendix B.3.

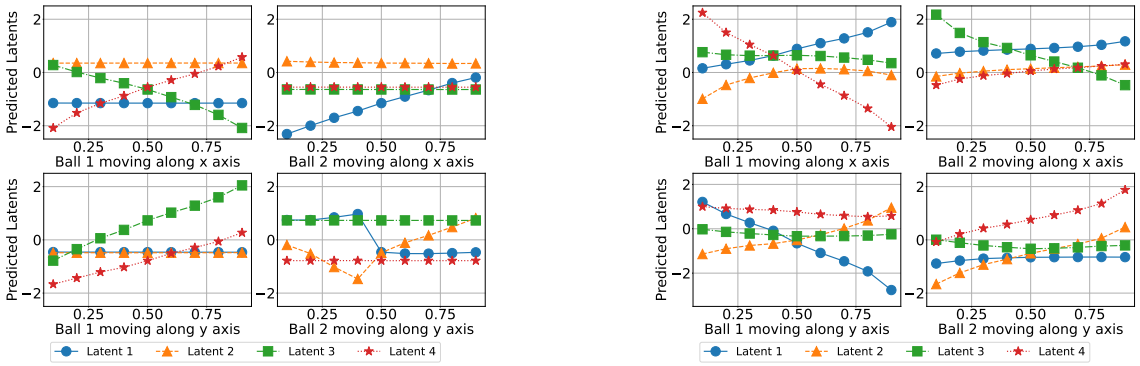
7.4.1. Results

Additivity is important for disentanglement. Table 7.1 shows that the additive decoder obtains a much higher $\text{LMS}_{\text{Spear}}$ & LMS_{Tree} than its non-additive counterpart on all three datasets considered, even if both decoders have very small reconstruction errors. This is corroborated by the visualizations of Figures 7.4 & 7.5. Appendix B.5 additionally shows object-specific reconstructions for the **BlockLatents** dataset. We emphasize that disentanglement is possible even when the latent factors are dependent (or causally related), as shown on the **ScalarLatents** dataset (L-shaped



(a) Additive decoder **(b)** Non-additive decoder

Figure 7.4. Figure (a) shows latent representation outputted by the encoder $\hat{\boldsymbol{g}}(\boldsymbol{x})$ over the *training* dataset, and the corresponding reconstructed images of the additive decoder with median $\text{LMS}_{\text{Spear}}$ among runs performed on the **ScalarLatents** dataset. Figure (b) shows the same thing for the non-additive decoder. The color gradient corresponds to the value of one of the ground-truth factor, the red dots correspond to factors used to generate the images and the yellow dashed square highlights extrapolated images.



(a) Additive Decoder **(b)** Non-Additive Decoder

Figure 7.5. Latent responses for the case of independent latents in the **BlockLatent** dataset. In each plot, we report the latent factors predicted from multiple images where one ball moves along only one axis at a time. For the additive case, at most two latents change, as it should, while more than two latents change for the non-additive case. See Appendix B.5 for details.

support implies dependencies) and on the **BlockLatents** dataset with dependencies (Table 7.1). Note that prior works have relied on interventions [Ahuja et al., 2023, 2022b, Brehmer et al., 2022] or Cartesian-product supports Wang and Jordan [2022], Roth et al. [2023] to deal with dependencies.

Additivity is important for Cartesian-product extrapolation. Figure 7.4 illustrates that the additive decoder can generate images that are outside the training domain (both balls in upper half of the image) while its non-additive counterpart cannot. Furthermore, Table 7.1 also corroborates this showing that the "out-of-support" (OOS) reconstruction MSE and $\text{LMS}_{\text{Spear}}$ (evaluated only on the samples never seen during training) are significantly better for the additive than for the non-additive decoder.

Importance of connected support. Theorem 7.2 required that the support of the latent factors, $\mathcal{Z}^{\text{train}}$, was path-connected. Appendix B.6 shows experiments where this assumption is violated,

which yields lower $\text{LMS}_{\text{Spear}}$ for the additive decoder, thus highlighting the importance of this assumption.

7.5. Conclusion

We provided an in-depth identifiability analysis of *additive decoders*, which bears resemblance to standard decoders used in OCRL, and introduced a novel theoretical framework showing how this architecture can generate reasonable images never seen during training via "Cartesian-product extrapolation". We validated empirically both of these results and confirmed that additivity was indeed crucial. By studying rigorously how disentanglement can induce extrapolation, our work highlighted the necessity of restricting the decoder to extrapolate and set the stage for future works to explore disentanglement and extrapolation in other function classes such as masked decoders typically used in OCRL. We postulate that the type of identifiability analysis introduced in this work has the potential of expanding our understanding of creativity in generative models, ultimately resulting in representations that generalize better.

Appendices of Chapter 7

A. Identifiability and Extrapolation Analysis

A.1. Useful definitions and lemmas

We start by recalling some notions of general topology that are going to be used later on. For a proper introduction to these concepts, see for example Munkres [2000].

Definition 7.6 (Regularly closed sets)**.** *A set $\mathcal{Z} \subseteq \mathbb{R}^{d_z}$ is regularly closed if $\mathcal{Z} = \overline{\mathcal{Z}^\circ}$, i.e. if it is equal to the closure of its interior (in the standard topology of $\mathbb{R}^n$).*

Definition 7.7 (Connected sets)**.** *A set $\mathcal{Z} \subseteq \mathbb{R}^{d_z}$ is connected if it cannot be written as a union of non-empty and disjoint open sets (in the subspace topology).*

Definition 7.8 (Path-connected sets)**.** *A set $\mathcal{Z} \subseteq \mathbb{R}^{d_z}$ is path-connected if for all pair of points $\boldsymbol{z}^0, \boldsymbol{z}^1 \in \mathcal{Z}$, there exists a continuous map $\boldsymbol{\phi} : [0,1] \to \mathcal{Z}$ such that $\boldsymbol{\phi}(0) = \boldsymbol{z}^0$ and $\boldsymbol{\phi}(1) = \boldsymbol{z}^1$. Such a map is called a path between $\boldsymbol{z}^0$ and $\boldsymbol{z}^1$.*

Definition 7.9 (Homeomorphism)**.** *Let A and B be subsets of $\mathbb{R}^n$ equipped with the subspace topology. A function $\boldsymbol{f} : A \to B$ is an homeomorphism if it is bijective, continuous and its inverse is continuous.*

The following technical lemma will be useful in the proof of Theorem 7.1. For it, we will need additional notation: Let $S \subseteq A \subseteq \mathbb{R}^n$. We already saw that $\overline{S}$ refers to the closure S in the $\mathbb{R}^n$ topology. We will denote by $\text{cl}_A(S)$ the closure of S in the subspace topology of A induced by $\mathbb{R}^n$, which is not necessarily the same as $\overline{S}$. In fact, both can be related via $\text{cl}_A = \overline{S} \cap A$ (see Munkres [2000, Theorem 17.4, p.95]).

Lemma 7.1. *Let $A, B \subseteq \mathbb{R}^n$ and suppose there exists an homeomorphism $\boldsymbol{f} : A \to B$. If A is regularly closed in $\mathbb{R}^n$, we have that $B \subseteq \overline{B^\circ}$.*

Proof Note that $\boldsymbol{f}\big|_{A^\circ}$ is a continuous injective function from the open set A° to $\boldsymbol{f}(A^\circ)$. By the "invariance of domain" theorem [Munkres, 2000, p.381], we have that $\boldsymbol{f}(A^\circ)$ must be open in $\mathbb{R}^n$. Of course, we have that $\boldsymbol{f}(A^\circ) \subseteq B$, and thus $\boldsymbol{f}(A^\circ) \subseteq B^\circ$ (the interior of B is the largest open set contained in B). Analogously, $\boldsymbol{f}^{-1}\big|_{B^\circ}$ is a continuous injective function from the open set B° to

Calligraphic & indexing conventions

$[n]$	$:=$	$\{1, 2, \ldots, n\}$
x		Scalar (random or not, depending on context)
$\boldsymbol{x}$		Vector (random or not, depending on context)
$\boldsymbol{X}$		Matrix
$\mathcal{X}$		Set/Support
f		Scalar-valued function
$\boldsymbol{f}$		Vector-valued function
$f\big\vert_A$		Restriction of f to the set A
Df, $D\boldsymbol{f}$		Jacobian of f and $\boldsymbol{f}$
$D^2 f$		Hessian of f
$B \subseteq [n]$		Subset of indices
$\lvert B \rvert$		Cardinality of the set B
$\boldsymbol{x}_B$		Vector formed with the ith coordinates of $\boldsymbol{x}$, for all $i \in B$
$\boldsymbol{X}_{B,B'}$		Matrix formed with the entries $(i, j) \in B \times B'$ of $\boldsymbol{X}$.
Given $\mathcal{X} \subseteq \mathbb{R}^n$, $\mathcal{X}_B$	$:=$	$\{\boldsymbol{x}_B \mid \boldsymbol{x} \in \mathcal{X}\}$ (projection of $\mathcal{X}$)

Recurrent notation

$\boldsymbol{x} \in \mathbb{R}^{d_x}$		Observation
$\boldsymbol{z} \in \mathbb{R}^{d_z}$		Vector of latent factors of variations
$\mathcal{Z} \subseteq \mathbb{R}^{d_z}$		Support of $\boldsymbol{z}$
$\boldsymbol{f}$		Ground-truth decoder function
$\hat{\boldsymbol{f}}$		Learned decoder function
$\mathcal{B}$		A partition of $[d_z]$ (assumed contiguous w.l.o.g.)
$B \in \mathcal{B}$		A block of the partition $\mathcal{B}$
$B(i) \in \mathcal{B}$		The unique block of $\mathcal{B}$ that contains i
$\pi : [d_z] \to [d_z]$		A permutation
$S_{\mathcal{B}}$	$:=$	$\bigcup_{B \in \mathcal{B}} B^2$
$S_{\mathcal{B}}^c$	$:=$	$[d_z]^2 \setminus S_{\mathcal{B}}$
$\mathbb{R}^{d_z \times d_z}_{S_{\mathcal{B}}}$	$:=$	$\{\boldsymbol{M} \in \mathbb{R}^{d_z \times d_z} \mid (i,j) \notin S_{\mathcal{B}} \implies \boldsymbol{M}_{i,j} = 0\}$

General topology

$\overline{\mathcal{X}}$	Closure of the subset $\mathcal{X} \subseteq \mathbb{R}^n$ in the standard topology of $\mathbb{R}^n$
$\mathcal{X}^\circ$	Interior of the subset $\mathcal{X} \subseteq \mathbb{R}^n$ in the standard topology of $\mathbb{R}^n$

Table 7.2. Table of notation.

$\boldsymbol{f}^{-1}(B^\circ)$. Again, by "invariance of domain", $\boldsymbol{f}^{-1}(B^\circ)$ must be open in $\mathbb{R}^n$ and thus $\boldsymbol{f}^{-1}(B^\circ) \subseteq A^\circ$. We can conclude that $\boldsymbol{f}(A^\circ) = B^\circ$.

We can conclude as follow:

$$B = \boldsymbol{f}(A) = \boldsymbol{f}(\overline{A^\circ}) = \boldsymbol{f}(\overline{A^\circ} \cap A) = \boldsymbol{f}(\mathrm{cl}_A(A^\circ)) \subseteq \mathrm{cl}_B(\boldsymbol{f}(A^\circ)) = \mathrm{cl}_B(B^\circ) = \overline{B^\circ} \cap B \subseteq \overline{B^\circ}\,,$$

where the first inclusion holds by continuity of $\boldsymbol{f}$ [Munkres, 2000, Thm.18.1 p.104]. ■

This lemma is taken from Lachapelle et al. [2022].

Lemma 7.2 (Sparsity pattern of an invertible matrix contains a permutation)**.** *Let $\boldsymbol{L} \in \mathbb{R}^{m\times m}$ be an invertible matrix. Then, there exists a permutation σ such that $\boldsymbol{L}_{i,\sigma(i)} \neq 0$ for all i.*

Proof Since the matrix $\boldsymbol{L}$ is invertible, its determinant is non-zero, i.e.

$$\det(\boldsymbol{L}) := \sum_{\pi\in\mathfrak{S}_m} \mathrm{sign}(\pi) \prod_{i=1}^{m} \boldsymbol{L}_{i,\pi(i)} \neq 0\,, \tag{7.12}$$

where $\mathfrak{S}_m$ is the set of m-permutations. This equation implies that at least one term of the sum is non-zero, meaning there exists $\pi \in \mathfrak{S}_m$ such that for all $i \in [m]$, $\boldsymbol{L}_{i,\pi(i)} \neq 0$. ■

Definition 7.10 (Aligned subspaces of $\mathbb{R}^{m\times n}$)**.** *Given a subset $S \subseteq \{1, ..., m\} \times \{1, ..., n\}$, we define*

$$\mathbb{R}^{m\times n}_S := \{\boldsymbol{M} \in \mathbb{R}^{m\times n} \mid (i, j) \notin S \implies \boldsymbol{M}_{i,j} = 0\}\,. \tag{7.13}$$

Definition 7.11 (Useful sets)**.** *Given a partition $\mathcal{B}$ of $[d]$, we define*

$$S_{\mathcal{B}} := \bigcup_{B\in\mathcal{B}} B^2 \qquad S^c_{\mathcal{B}} := \{1, \dots, d_z\}^2 \setminus S_{\mathcal{B}} \tag{7.14}$$

Definition 7.12 (C^k-diffeomorphism)**.** *Let $A \subseteq \mathbb{R}^n$ and $B \subseteq \mathbb{R}^m$. A map $\boldsymbol{f} : A \to B$ is said to be a C^k-diffeomorphism if it is bijective, C^2 and has a C^2 inverse.*

Remark 7.2. *Differentiability is typically defined for functions that have an open domain in $\mathbb{R}^n$. However, in the definition above, the set A might not be open in $\mathbb{R}^n$ and B might not be open in $\mathbb{R}^m$. In the case of an arbitrary domain A, it is customary to say that a function $\boldsymbol{f} : A \subseteq \mathbb{R}^n \to \mathbb{R}^m$ is C^k if there exists a C^k function $\boldsymbol{g}$ defined on an open set $U \subseteq \mathbb{R}^n$ that contains A such that $\boldsymbol{g}\big|_A = \boldsymbol{f}$ (i.e. $\boldsymbol{g}$ extends $\boldsymbol{f}$). With this definition, we have that a composition of C^k functions is C^k, as usual. See for example p.199 of Munkres [1991].*

The following lemma allows us to unambiguously define the k first derivatives of a C^k function $\boldsymbol{f} : A \to \mathbb{R}^m$ on the set $\overline{A^\circ}$.

Lemma 7.3. *Let $A \subseteq \mathbb{R}^n$ and $\boldsymbol{f} : A \to \mathbb{R}^m$ be a C^k function. Then, its k first derivatives is uniquely defined on $\overline{A^\circ}$ in the sense that they do not depend on the specific choice of C^k extension.*

Proof Let $\boldsymbol{g} : U \to \mathbb{R}^n$ and $\boldsymbol{h} : V \to \mathbb{R}^n$ be two C^k extensions of $\boldsymbol{f}$ to $U \subseteq \mathbb{R}^n$ and $V \subseteq \mathbb{R}^n$ both open in $\mathbb{R}^n$. By definition,

$$\boldsymbol{g}(\boldsymbol{x}) = \boldsymbol{f}(\boldsymbol{x}) = \boldsymbol{h}(\boldsymbol{x}),\ \forall \boldsymbol{x} \in A\,. \tag{7.15}$$

The usual derivative is uniquely defined on the interior of the domain, so that

$$D\boldsymbol{g}(\boldsymbol{x}) = D\boldsymbol{f}(\boldsymbol{x}) = D\boldsymbol{h}(\boldsymbol{x}),\ \forall \boldsymbol{x} \in A^\circ\,. \tag{7.16}$$

Consider a point $\boldsymbol{x}_0 \in \overline{A^\circ}$. By definition of closure, there exists a sequence $\{\boldsymbol{x}_k\}_{k=1}^\infty \subseteq A^\circ$ s.t. $\lim_{k\to\infty} \boldsymbol{x}_k = \boldsymbol{x}_0$. We thus have that

$$\lim_{k\to\infty} D\boldsymbol{g}(\boldsymbol{x}_k) = \lim_{k\to\infty} D\boldsymbol{h}(\boldsymbol{x}_k) \tag{7.17}$$

$$D\boldsymbol{g}(\boldsymbol{x}_0) = D\boldsymbol{h}(\boldsymbol{x}_0)\,, \tag{7.18}$$

where we used the fact that the derivatives of $\boldsymbol{g}$ and $\boldsymbol{h}$ are continuous to go to the second line. Thus, all the C^k extensions of $\boldsymbol{f}$ must have equal derivatives on $\overline{A^\circ}$. This means we can unambiguously define the derivative of $\boldsymbol{f}$ everywhere on $\overline{A^\circ}$ to be equal to the derivative of one of its C^k extensions.

Since $\boldsymbol{f}$ is C^k, its derivative $D\boldsymbol{f}$ is C^{k-1}, we can thus apply the same argument to get that the second derivative of $\boldsymbol{f}$ is uniquely defined on $\overline{\overline{A^\circ}^\circ}$. It can be shown that $\overline{\overline{A^\circ}^\circ} = \overline{A^\circ}$. One can thus apply the same argument recursively to show that the first k derivatives of $\boldsymbol{f}$ are uniquely defined on $\overline{A^\circ}$. ■

Definition 7.13 (C^k-diffeomorphism onto its image)**.** *Let $A \subseteq \mathbb{R}^n$. A map $\boldsymbol{f} : A \to \mathbb{R}^m$ is said to be a C^k-diffeomorphism onto its image if the restriction $\boldsymbol{f}$ to its image $\tilde{\boldsymbol{f}} : A \to \boldsymbol{f}(A)$ is a C^k-diffeomorphism.*

Remark 7.3. *If $S \subseteq A \subseteq \mathbb{R}^n$ and $\boldsymbol{f} : A \to \mathbb{R}^m$ is a C^k-diffeomorphism on its image, then the restriction of $\boldsymbol{f}$ to S, i.e. $\boldsymbol{f}\big|_S$, is also a C^k diffeomorphism on its image. That is because $\boldsymbol{f}\big|_S$ is clearly bijective, is C^k (simply take the C^k extension of $\boldsymbol{f}$) and so is its inverse (simply take the C^k extension of $\boldsymbol{f}^{-1}$).*

A.2. Relationship between additive decoders and the diagonal Hessian penalty

Proposition 7.1 (Equivalence between additivity and diagonal Hessian)**.** *Let $\boldsymbol{f} : \mathbb{R}^{d_z} \to \mathbb{R}^{d_x}$ be a C^2 function. Then,*

$$\begin{array}{l} \forall \boldsymbol{z} \in \mathbb{R}^{d_z},\ \boldsymbol{f}(\boldsymbol{z}) = \sum_{B\in\mathcal{B}} \boldsymbol{f}^{(B)}(\boldsymbol{z}_B) \\ \textit{where } \boldsymbol{f}^{(B)} : \mathbb{R}^{|B|} \to \mathbb{R}^{d_x} \textit{ is } C^2. \end{array} \iff \begin{array}{l} \forall k \in [d_x],\ \boldsymbol{z} \in \mathbb{R}^{d_z},\ \ D^2\boldsymbol{f}_k(\boldsymbol{z}) \textit{ is} \\ \textit{block diagonal with blocks in } \mathcal{B}. \end{array} \tag{7.19}$$

Proof We start by showing the " $\implies$ " direction. Let B and B' be two distinct blocks of $\mathcal{B}$. Let $i \in B$ and $i' \in B'$. We can compute the derivative of $\boldsymbol{f}_k$ w.r.t. $\boldsymbol{z}_i$:

$$D_i \boldsymbol{f}_k(\boldsymbol{z}) = \sum_{\bar{B} \in \mathcal{B}} D_i \boldsymbol{f}_k^{(\bar{B})}(\boldsymbol{z}_{\bar{B}}) = D_i \boldsymbol{f}_k^{(B)}(z_B)\,, \tag{7.20}$$

where the last equality holds because $i \in B$ and not in any other block $\bar{B}$. Furthermore,

$$D^2_{i,i'} \boldsymbol{f}_k(\boldsymbol{z}) = D^2_{i,i'} \boldsymbol{f}_k^{(B)}(\boldsymbol{z}_B) = 0\,, \tag{7.21}$$

where the last equality holds because $i' \notin B$. This shows that $D^2 \boldsymbol{f}_k(\boldsymbol{z})$ is block diagonal.

We now show the " $\impliedby$ " direction. Fix $k \in [d_x]$, $B \in \mathcal{B}$. We know that $D^2_{B,B^c} \boldsymbol{f}_k(\boldsymbol{z}) = 0$ for all $\boldsymbol{z} \in \mathbb{R}^{d_z}$. Fix $\boldsymbol{z} \in \mathbb{R}^{d_z}$. Consider a continuously differentiable path $\boldsymbol{\phi} : [0,1] \to \mathbb{R}^{|B^c|}$ such that $\boldsymbol{\phi}(0) = 0$ and $\boldsymbol{\phi}(1) = \boldsymbol{z}_{B^c}$. As $D^2_{B,B^c} \boldsymbol{f}_k(\boldsymbol{z})$ is a continuous function of $\boldsymbol{z}$, we can use the fundamental theorem of calculus for line integrals to get that

$$D_B \boldsymbol{f}_k(\boldsymbol{z}_B, \boldsymbol{z}_{B^c}) - D_B \boldsymbol{f}_k(\boldsymbol{z}_B, 0) = \int_0^1 \underbrace{D^2_{B,B^c} \boldsymbol{f}_k(\boldsymbol{z}_B, \boldsymbol{\phi}(t))}_{=0} \boldsymbol{\phi}'(t) dt = 0\,, \tag{7.22}$$

(where $D^2_{B,B^c} \boldsymbol{f}_k(\boldsymbol{z}_B, \boldsymbol{\phi}(t)) \boldsymbol{\phi}'(t)$ denotes a matrix-vector product) which implies that

$$D_B \boldsymbol{f}_k(\boldsymbol{z}) = D_B \boldsymbol{f}_k(\boldsymbol{z}_B, 0)\,. \tag{7.23}$$

And the above equality holds for all $B \in \mathcal{B}$ and all $\boldsymbol{z} \in \mathbb{R}^{d_z}$.

Choose an arbitrary $\boldsymbol{z} \in \mathbb{R}^{d_z}$. Consider a continously differentiable path $\boldsymbol{\psi} : [0,1] \to \mathbb{R}^{d_z}$ such that $\boldsymbol{\psi}(0) = 0$ and $\boldsymbol{\psi}(1) = \boldsymbol{z}$. By applying the fundamental theorem of calculus for line integrals once more, we have that

$$\begin{aligned}
\boldsymbol{f}_k(\boldsymbol{z}) - \boldsymbol{f}_k(0) &= \int_0^1 D \boldsymbol{f}_k(\boldsymbol{\psi}(t)) \boldsymbol{\psi}'(t) dt && (7.24)\\
&= \int_0^1 \sum_{B \in \mathcal{B}} D_B \boldsymbol{f}_k(\boldsymbol{\psi}(t)) \boldsymbol{\psi}'_B(t) dt && (7.25)\\
&= \sum_{B \in \mathcal{B}} \int_0^1 D_B \boldsymbol{f}_k(\boldsymbol{\psi}(t)) \boldsymbol{\psi}'_B(t) dt && (7.26)\\
&= \sum_{B \in \mathcal{B}} \int_0^1 D_B \boldsymbol{f}_k(\boldsymbol{\psi}_B(t), 0) \boldsymbol{\psi}'_B(t) dt\,, && (7.27)
\end{aligned}$$

where the last equality holds by (7.23). We can further apply the fundamental theorem of calculus for line integrals to each term $\int_0^1 D_B \boldsymbol{f}_k(\boldsymbol{\psi}_B(t), 0)\boldsymbol{\psi}'_B(t)dt$ to get

$$\boldsymbol{f}_k(\boldsymbol{z}) - \boldsymbol{f}_k(0) = \sum_{B\in\mathcal{B}} (\boldsymbol{f}_k(\boldsymbol{z}_B, 0) - \boldsymbol{f}_k(0, 0)) \tag{7.28}$$

$$\implies \boldsymbol{f}_k(\boldsymbol{z}) = \boldsymbol{f}_k(0) + \sum_{B\in\mathcal{B}} (\boldsymbol{f}_k(\boldsymbol{z}_B, 0) - \boldsymbol{f}_k(0)) \tag{7.29}$$

$$= \sum_{B\in\mathcal{B}} \underbrace{\left(\boldsymbol{f}_k(\boldsymbol{z}_B, 0) - \frac{|\mathcal{B}| - 1}{|\mathcal{B}|}\boldsymbol{f}_k(0)\right)}_{\boldsymbol{f}_k^{(B)}(\boldsymbol{z}_B) :=} . \tag{7.30}$$

and since $\boldsymbol{z}$ was arbitrary, the above holds for all $\boldsymbol{z} \in \mathbb{R}^{d_z}$. Note that the functions $\boldsymbol{f}_k^{(B)}(\boldsymbol{z}_B)$ must be C^2 because $\boldsymbol{f}_k$ is C^2. This concludes the proof. ■

A.3. Additive decoders form a superset of compositional decoders [Brady et al., 2023]

Compositional decoders were introduced by Brady et al. [2023] as a suitable class of functions to perform object-centric representation learning with identifiability guarantees. They are also interested in block-disentanglement, but, contrarily to our work, they assume that the latent vector $\boldsymbol{z}$ is fully supported, i.e. $\mathcal{Z} = \mathbb{R}^{d_z}$. We now rewrite the definition of compositional decoders in the notation used in this work:

Definition 7.14 (Compositional decoders, adapted from Brady et al. [2023])**.** *Given a partition $\mathcal{B}$, a differentiable decoder $\boldsymbol{f} : \mathbb{R}^{d_z} \to \mathbb{R}^{d_x}$ is said to be compositional w.r.t. $\mathcal{B}$ whenever the Jacobian $D\boldsymbol{f}(\boldsymbol{z})$ is such that for all $i \in [d_z], B \in \mathcal{B}, \boldsymbol{z} \in \mathbb{R}^{d_z}$, we have*

$$D_B \boldsymbol{f}_i(\boldsymbol{z}) \neq \mathbf{0} \implies D_{B^c} \boldsymbol{f}_i(\boldsymbol{z}) = \mathbf{0}\,,$$

where B^c is the complement of $B \in \mathcal{B}$.

In other words, each line of the Jacobian can have nonzero values only in one block $B \in \mathcal{B}$. Note that this nonzero block can change with different values of $\boldsymbol{z}$.

The next result shows that additive decoders form a superset of C^2 compositional decoders (Brady et al. [2023] assumed only C^1). Note that additive decoders are *strictly* more expressive than C^2 compositional decoders because some additive functions are not compositional, like Example 7.3 for instance.

Proposition 7.2 (Compositional implies additive)**.** *Given a partition $\mathcal{B}$, if $\boldsymbol{f} : \mathbb{R}^{d_z} \to \mathbb{R}^{d_x}$ is compositional (Definition 7.14) and C^2, then it is also additive (Definition 7.1).*

Proof Choose any $i \in [d_x]$. Our strategy will be to show that $D^2 \boldsymbol{f}_i$ is block diagonal everywhere on $\mathbb{R}^{d_z}$ and use Proposition 7.1 to conclude that $\boldsymbol{f}_i$ is additive.

Choose an arbitrary $\boldsymbol{z}_0 \in \mathbb{R}^{d_z}$. By compositionality, there exists a block $B \in \mathcal{B}$ such that $D_{B^c} \boldsymbol{f}_i(\boldsymbol{z}_0) = \mathbf{0}$. We consider two cases separately:

Case 1 Assume $D_B \boldsymbol{f}_i(\boldsymbol{z}_0) \neq \mathbf{0}$. By continuity of $D_B \boldsymbol{f}_i$, there exists an open neighborhood of $\boldsymbol{z}_0$, U, s.t. for all $\boldsymbol{z} \in U$, $D_B \boldsymbol{f}_i(\boldsymbol{z}) \neq \mathbf{0}$. By compositionality, this means that, for all $\boldsymbol{z} \in U$, $D_{B^c} \boldsymbol{f}_i(\boldsymbol{z}) = \mathbf{0}$. When a function is zero on an open set, its derivative must also be zero, hence $D D_{B^c} \boldsymbol{f}_i(\boldsymbol{z}_0) = \mathbf{0}$. Because $\boldsymbol{f}$ is C^2, the Hessian is symmetric so that we also have $D_{B^c} D \boldsymbol{f}_i(\boldsymbol{z}_0) = \mathbf{0}$. We can thus conclude that the Hessian $D^2 \boldsymbol{f}_i(\boldsymbol{z}_0)$ is such that all entries are zero except possibly for $D^2 \boldsymbol{f}_i(\boldsymbol{z}_0)_{B,B}$. Hence, $D^2 \boldsymbol{f}_i(\boldsymbol{z}_0)$ is block diagonal with blocks in $\mathcal{B}$.

Case 2: Assume $D_B \boldsymbol{f}_i(\boldsymbol{z}_0) = \mathbf{0}$. This means the whole row of the Jacobian is zero, i.e. $D \boldsymbol{f}_i(\boldsymbol{z}_0) = \mathbf{0}$. By continuity of $D \boldsymbol{f}_i$, we have that the set $V := (D \boldsymbol{f}_i)^{-1}(\{0\})$ is closed. Thus this set decomposes as $V = V^\circ \cup \partial V$ where V° and ∂V are the interior and boundary of V, respectively.

Case 2.1: Suppose $\boldsymbol{z}_0 \in V^\circ$. Then we can take a derivative so that $D^2 \boldsymbol{f}_i(\boldsymbol{z}_0) = \mathbf{0}$, which of course means that $D^2 \boldsymbol{f}_i(\boldsymbol{z}_0)$ is diagonal.

Case 2.2: Suppose $\boldsymbol{z}_0 \in \partial V$. By the definition of boundary, for all open set U containing $\boldsymbol{z}_0$, U intersects with the complement of V, i.e. $(D \boldsymbol{f}_i)^{-1}(\mathbb{R}^{d_z} \setminus \{0\})$. This means we can construct a sequence $\{\boldsymbol{z}_k\}_{k=1}^{\infty} \subseteq V^c$ which converges to $\boldsymbol{z}_0$. By **Case 1**, we have that for all $k \geq 1$, $D^2 \boldsymbol{f}_i(\boldsymbol{z}_k)$ is block diagonal. This means that $\lim_{k \to \infty} D^2 \boldsymbol{f}_i(\boldsymbol{z}_k)$ is block diagonal. Moreover, by continuity of $D^2 \boldsymbol{f}_i$, we have that $\lim_{k \to \infty} D^2 \boldsymbol{f}_i(\boldsymbol{z}_k) = D^2 \boldsymbol{f}_i(\boldsymbol{z}_0)$. Hence $D^2 \boldsymbol{f}_i(\boldsymbol{z}_0)$ is block diagonal.

We showed that for all $\boldsymbol{z}_0 \in \mathbb{R}^{d_z}$, $D^2 \boldsymbol{f}_i(\boldsymbol{z}_0)$ is block diagonal. Hence, $\boldsymbol{f}$ is additive by Proposition 7.1. ■

A.4. Examples of local but non-global disentanglement

In this section, we provide examples of mapping $\boldsymbol{v} : \hat{\mathcal{Z}}^{\text{train}} \to \mathcal{Z}^{\text{train}}$ that satisfy the *local* disentanglement property of Definition 7.4, but not the *global* disentanglement property of Definition 7.3. Note that these notions are defined for pairs of decoders $\boldsymbol{f}$ and $\hat{\boldsymbol{f}}$, but here we construct directly the function $\boldsymbol{v}$ which is usually defined as $\boldsymbol{f}^{-1} \circ \hat{\boldsymbol{f}}$. However, given $\boldsymbol{v}$ we can always define $\boldsymbol{f}$ and $\hat{\boldsymbol{f}}$ to be such that $\boldsymbol{f}^{-1} \circ \hat{\boldsymbol{f}} = \boldsymbol{v}$: Simply take $\boldsymbol{f}(\boldsymbol{z}) := [\boldsymbol{z}_1, \ldots, \boldsymbol{z}_{d_z}, 0, \ldots, 0]^\top \in \mathbb{R}^{d_x}$ and $\hat{\boldsymbol{f}} := \boldsymbol{f} \circ \boldsymbol{v}$. This construction however yields a decoder $\boldsymbol{f}$ that is not sufficiently nonlinear (Assumption 7.2). Clearly the mappings $\boldsymbol{v}$ that we provide in the following examples cannot be written as compositions of decoders $\boldsymbol{f}^{-1} \circ \hat{\boldsymbol{f}}$ where $\boldsymbol{f}$ and $\hat{\boldsymbol{f}}$ satisfy all assumptions of Theorem 7.2, as this would contradict the theorem. In Examples 7.5 & 7.6, the path-connected assumption of Theorem 7.2 is violated. In Example 7.7, it is less obvious to see which assumptions would be violated.

Example 7.5 (Disconnected support with changing permutation)**.** *Let* $\boldsymbol{v} : \hat{\mathcal{Z}} \to \mathbb{R}^2$ *s.t.* $\hat{\mathcal{Z}} = \hat{\mathcal{Z}}^{(1)} \cup \hat{\mathcal{Z}}^{(2)} \subseteq \mathbb{R}^2$ *where* $\hat{\mathcal{Z}}^{(1)} = \{\boldsymbol{z} \in \mathbb{R}^2 \mid \boldsymbol{z}_1 \leq 0 \text{ and } \boldsymbol{z}_2 \leq 0\}$ *and* $\hat{\mathcal{Z}}^{(2)} = \{\boldsymbol{z} \in \mathbb{R}^2 \mid \boldsymbol{z}_1 \geq 1 \text{ and } \boldsymbol{z}_2 \geq 1\}$*. Assume*

$$\boldsymbol{v}(\boldsymbol{z}) := \begin{cases} (\boldsymbol{z}_1, \boldsymbol{z}_2), & \text{if } \boldsymbol{z} \in \hat{\mathcal{Z}}^{(1)} \\ (\boldsymbol{z}_2, \boldsymbol{z}_1), & \text{if } \boldsymbol{z} \in \hat{\mathcal{Z}}^{(2)} \end{cases} . \tag{7.31}$$

Step 1: $\boldsymbol{v}$ ***is a diffeomorphism.*** *Note that* $\boldsymbol{v}$ *is its own inverse. Indeed,*

$$\boldsymbol{v}(\boldsymbol{v}(\boldsymbol{z})) = \begin{cases} \boldsymbol{v}(\boldsymbol{z}_1, \boldsymbol{z}_2) = (\boldsymbol{z}_1, \boldsymbol{z}_2), & \text{if } \boldsymbol{z} \in \hat{\mathcal{Z}}^{(1)} \\ \boldsymbol{v}(\boldsymbol{z}_2, \boldsymbol{z}_1) = (\boldsymbol{z}_1, \boldsymbol{z}_2), & \text{if } \boldsymbol{z} \in \hat{\mathcal{Z}}^{(2)} \end{cases} .$$

Thus, $\boldsymbol{v}$ *is bijective on its image. Clearly,* $\boldsymbol{v}$ *is* C^2*, thus* $\boldsymbol{v}^{-1} = \boldsymbol{v}$ *is also* C^2*. Hence,* $\boldsymbol{v}$ *is a* C^2*-diffeomorphism.*

Step 2: $\boldsymbol{v}$ ***is locally disentangled.*** *The Jacobian of* $\boldsymbol{v}$ *is given by*

$$D\boldsymbol{v}(\boldsymbol{z}) := \begin{cases} \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}, & \text{if } \boldsymbol{z} \in \hat{\mathcal{Z}}^{(1)} \\ \\ \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}, & \text{if } \boldsymbol{z} \in \hat{\mathcal{Z}}^{(2)} \end{cases} , \tag{7.32}$$

which is everywhere a permutation matrix, hence $\boldsymbol{v}$ *is locally disentangled.*

Step 3: $\boldsymbol{v}$ ***is not globally disentangled.*** *That is because* $\boldsymbol{v}_1(\boldsymbol{z}_1, \boldsymbol{z}_2)$ *depends on both* $\boldsymbol{z}_1$ *and* $\boldsymbol{z}_2$*. Indeed, if* $\boldsymbol{z}_2 = 0$*, we have that* $\boldsymbol{v}_1(-1, 0) = -1 \neq 0 = \boldsymbol{v}_1(0, 0)$*. Also, if* $\boldsymbol{z}_1 = 1$*, we have that* $\boldsymbol{v}_1(1, 1) = 1 \neq 2 = \boldsymbol{v}_1(1, 2)$*.*

Example 7.6 (Disconnected support with fixed permutation)**.** *Let* $\boldsymbol{v} : \hat{\mathcal{Z}} \to \mathbb{R}^2$ *s.t.* $\hat{\mathcal{Z}} = \hat{\mathcal{Z}}^{(1)} \cup \hat{\mathcal{Z}}^{(2)} \subseteq \mathbb{R}^2$ *where* $\hat{\mathcal{Z}}^{(1)} = \{\boldsymbol{z} \in \mathbb{R}^2 \mid \boldsymbol{z}_2 \leq 0\}$ *and* $\hat{\mathcal{Z}}^{(2)} = \{\boldsymbol{z} \in \mathbb{R}^2 \mid \boldsymbol{z}_2 \geq 1\}$*. Assume* $\boldsymbol{v}(\boldsymbol{z}) := \boldsymbol{z} + \mathbb{1}(\boldsymbol{z} \in \hat{\mathcal{Z}}^{(2)})$*.*

Step 1: $\boldsymbol{v}$ ***is a diffeomorphism.*** *The image of* $\boldsymbol{v}$ *is the union of the following two sets:* $\mathcal{Z}^{(1)} := \boldsymbol{v}(\hat{\mathcal{Z}}^{(1)}) = \hat{\mathcal{Z}}^{(1)}$ *and* $\mathcal{Z}^{(2)} := \boldsymbol{v}(\hat{\mathcal{Z}}^{(2)}) = \{\boldsymbol{z} \in \mathbb{R}^2 \mid \boldsymbol{z}_2 \geq 2\}$*. Consider the map* $\boldsymbol{w} : \mathcal{Z}^{(1)} \cup \mathcal{Z}^{(2)} \to \hat{\mathcal{Z}}$ *defined as* $\boldsymbol{w}(\boldsymbol{z}) := \boldsymbol{z} - \mathbb{1}(\boldsymbol{z} \in \mathcal{Z}^{(2)})$*. We now show that* $\boldsymbol{w}$ *is the inverse of* $\boldsymbol{v}$*:*

$$\boldsymbol{w}(\boldsymbol{v}(\boldsymbol{z})) = \boldsymbol{v}(\boldsymbol{z}) - \mathbb{1}(\boldsymbol{v}(\boldsymbol{z}) \in \mathcal{Z}^{(2)}) \tag{7.33}$$

$$= \boldsymbol{z} + \mathbb{1}(\boldsymbol{z} \in \hat{\mathcal{Z}}^{(2)}) - \mathbb{1}(\boldsymbol{z} + \mathbb{1}(\boldsymbol{z} \in \hat{\mathcal{Z}}^{(2)}) \in \mathcal{Z}^{(2)}) . \tag{7.34}$$

If $\boldsymbol{z} \in \hat{\mathcal{Z}}^{(2)}$*, we have*

$$\boldsymbol{w}(\boldsymbol{v}(\boldsymbol{z})) = \boldsymbol{z} + \mathbb{1} - \mathbb{1}(\boldsymbol{z} + \mathbb{1} \in \mathcal{Z}^{(2)}) \tag{7.35}$$

$$= \boldsymbol{z} + \mathbb{1} - \mathbb{1}(\boldsymbol{z} \in \hat{\mathcal{Z}}^{(2)}) = \boldsymbol{z} . \tag{7.36}$$

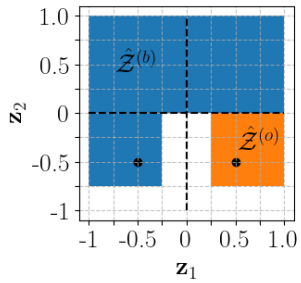


Figure 7.6. Illustration of $\hat{\mathcal{Z}} = \hat{\mathcal{Z}}^{(b)} \cup \hat{\mathcal{Z}}^{(o)}$ in Example 7.7 where $\hat{\mathcal{Z}}^{(b)}$ is the blue region and $\hat{\mathcal{Z}}^{(o)}$ is the orange region. The two black dots correspond to $(-1/2, -1/2)$ and $(1/2, -1/2)$, where the function $\boldsymbol{v}_2(\boldsymbol{z}_1, \boldsymbol{z}_2)$ is evaluated to show that it is not constant in $\boldsymbol{z}_1$.

If $\boldsymbol{z} \in \hat{\mathcal{Z}}^{(1)}$, we have

$$\boldsymbol{w}(\boldsymbol{v}(\boldsymbol{z})) = \boldsymbol{z} - \mathbb{1}(\boldsymbol{z} \in \mathcal{Z}^{(2)}) = \boldsymbol{z}\,. \tag{7.37}$$

A similar argument can be made to show that $\boldsymbol{v}(\boldsymbol{w}(\boldsymbol{z})) = \boldsymbol{z}$. Thus $\boldsymbol{w}$ is the inverse of $\boldsymbol{v}$. Both $\boldsymbol{v}$ and its inverse $\boldsymbol{w}$ are C^2, thus $\boldsymbol{v}$ is a C^2-diffeomorphism on its image.

Step 2: $\boldsymbol{v}$ is locally disentangled. *This is clear since $D\boldsymbol{v}(\boldsymbol{z}) = \boldsymbol{I}$ everywhere.*

Step 3: $\boldsymbol{v}$ is not globally disentangled. *Indeed, the function $\boldsymbol{v}_1(\boldsymbol{z}_1, \boldsymbol{z}_2) = \boldsymbol{z}_1 + \mathbb{1}(\boldsymbol{z} \in \hat{\mathcal{Z}}^{(2)})$ is not constant in $\boldsymbol{z}_2$.*

Example 7.7 (Connected support)**.** *Let $\boldsymbol{v} : \hat{\mathcal{Z}} \to \mathbb{R}^2$ s.t. $\hat{\mathcal{Z}} = \hat{\mathcal{Z}}^{(b)} \cup \hat{\mathcal{Z}}^{(o)}$ where $\hat{\mathcal{Z}}^{(b)}$ and $\hat{\mathcal{Z}}^{(o)}$ are respectively the blue and orange regions of Figure 7.6. Both regions contain their boundaries. The function $\boldsymbol{v}$ is defined as follows:*

$$\boldsymbol{v}_1(\boldsymbol{z}) := \boldsymbol{z}_1 \tag{7.38}$$

$$\boldsymbol{v}_2(\boldsymbol{z}) := \begin{cases} \frac{(\boldsymbol{z}_2+1)^2+1}{2}, & \text{if } \boldsymbol{z} \in \hat{\mathcal{Z}}^{(b)} \\ e^{\boldsymbol{z}_2}, & \text{if } \boldsymbol{z} \in \hat{\mathcal{Z}}^{(o)} \end{cases}. \tag{7.39}$$

Step 1: $\boldsymbol{v}$ is a diffeomorphism. *Clearly, $\boldsymbol{v}_1$ is C^2. To show that $\boldsymbol{v}_2$ also is, we must verify that $\boldsymbol{v}_2(\boldsymbol{z})$ is C^2 at the frontier between $\hat{\mathcal{Z}}^{(b)}$ and $\hat{\mathcal{Z}}^{(o)}$, i.e. when $\boldsymbol{z} \in [1/4, 1] \times \{0\}$.*

*$\boldsymbol{v}_2(\boldsymbol{z})$ **is continuous** since*

$$\left.\frac{(\boldsymbol{z}_2+1)^2+1}{2}\right|_{\boldsymbol{z}_2=0} = 1 = \left. e^{\boldsymbol{z}_2}\right|_{\boldsymbol{z}_2=0}. \tag{7.40}$$

*$\boldsymbol{v}_2(\boldsymbol{z})$ **is** C^1 since*

$$\left(\frac{(\boldsymbol{z}_2+1)^2+1}{2}\right)'\Bigg|_{\boldsymbol{z}_2=0} = (\boldsymbol{z}_2+1)|_{\boldsymbol{z}_2=0} = 1 = \left. e^{\boldsymbol{z}_2}\right|_{\boldsymbol{z}_2=0} = (e^{\boldsymbol{z}_2})'|_{\boldsymbol{z}_2=0}\,. \tag{7.41}$$

*$\boldsymbol{v}_2(\boldsymbol{z})$ **is** C^2 since*

$$\left(\frac{(\boldsymbol{z}_2+1)^2+1}{2}\right)''\Big|_{\boldsymbol{z}_2=0} = 1|_{\boldsymbol{z}_2=0} = 1 = e^{\boldsymbol{z}_2}|_{\boldsymbol{z}_2=0} = (e^{\boldsymbol{z}_2})''|_{\boldsymbol{z}_2=0}\,. \tag{7.42}$$

We will now find an explicit expression for the inverse of $\boldsymbol{v}$. Define

$$\boldsymbol{w}_1(\boldsymbol{z}) := \boldsymbol{z}_1 \tag{7.43}$$

$$\boldsymbol{w}_2(\boldsymbol{z}) := \begin{cases} \sqrt{2\boldsymbol{z}_2 - 1} - 1, & \text{if } \boldsymbol{z} \in \boldsymbol{v}(\hat{\mathcal{Z}}^{(b)}) \\ \log(\boldsymbol{z}_2), & \text{if } \boldsymbol{z} \in \boldsymbol{v}(\hat{\mathcal{Z}}^{(o)}) \end{cases}. \tag{7.44}$$

It is straightforward to see that $\boldsymbol{w}(\boldsymbol{v}(\boldsymbol{z})) = \boldsymbol{z}$ for all $\boldsymbol{z} \in \hat{\mathcal{Z}}$. One can also show that $\boldsymbol{w}$ is C^2 at the boundary between both regions $\boldsymbol{v}(\hat{\mathcal{Z}}^{(b)})$ and $\boldsymbol{v}(\hat{\mathcal{Z}}^{(o)})$, i.e. when $\boldsymbol{z} \in [1/4, 1] \times \{1\}$.

Since both $\boldsymbol{v}$ and its inverse $\boldsymbol{w}$ are C^2, $\boldsymbol{v}$ is a C^2-diffeomorphism.

***Step 2: $\boldsymbol{v}$ is locally disentangled.** The Jacobian of $\boldsymbol{v}$ is*

$$D\boldsymbol{v}(\boldsymbol{z}) := \begin{cases} \begin{bmatrix} 1 & 0 \\ 0 & \boldsymbol{z}_2 + 1 \end{bmatrix}, & \text{if } \boldsymbol{z} \in \hat{\mathcal{Z}}^{(b)} \\ \begin{bmatrix} 1 & 0 \\ 0 & e^{\boldsymbol{z}_2} \end{bmatrix}, & \text{if } \boldsymbol{z} \in \hat{\mathcal{Z}}^{(o)} \end{cases}, \tag{7.45}$$

which is a permutation-scaling matrix everywhere on $\hat{\mathcal{Z}}$. Thus local disentanglement holds.

***Step 3: $\boldsymbol{v}$ is not globally disentangled.** However, $\boldsymbol{v}_2(\boldsymbol{z}_1, \boldsymbol{z}_2)$ is not constant in $\boldsymbol{z}_1$. Indeed,*

$$\boldsymbol{v}_2(-\frac{1}{2}, -\frac{1}{2}) = \frac{(\boldsymbol{z}_2+1)^2+1}{2}\Big|_{\boldsymbol{z}_2=-1/2} = \frac{5}{8} \neq e^{-1/2} = \boldsymbol{v}_2(\frac{1}{2}, -\frac{1}{2})\,. \tag{7.46}$$

Thus global disentanglement does not hold.

A.5. Proof of Theorem 7.1

Proposition 7.3. *Suppose that the data-generating process satisfies Assumption 7.1, that the learned decoder $\hat{\boldsymbol{f}} : \mathbb{R}^{d_z} \to \mathbb{R}^{d_x}$ is a C^2-diffeomorphism onto its image and that the encoder $\hat{\boldsymbol{g}} : \mathbb{R}^{d_x} \to \mathbb{R}^{d_z}$ is continuous. Then, if $\hat{\boldsymbol{f}}$ and $\hat{\boldsymbol{g}}$ solve the reconstruction problem on the training distribution, i.e. $\mathbb{E}^{\text{train}}||\boldsymbol{x} - \hat{\boldsymbol{f}}(\hat{\boldsymbol{g}}(\boldsymbol{x}))||^2 = 0$, we have that $\boldsymbol{f}(\mathcal{Z}^{\text{train}}) = \hat{\boldsymbol{f}}(\hat{\mathcal{Z}}^{\text{train}})$ and the map $\boldsymbol{v} := \boldsymbol{f}^{-1} \circ \hat{\boldsymbol{f}}$ is a C^2-diffeomorphism from $\hat{\mathcal{Z}}^{\text{train}}$ to $\mathcal{Z}^{\text{train}}$.*

Proof First note that

$$\mathbb{E}^{\text{train}}||\boldsymbol{x} - \hat{\boldsymbol{f}}(\hat{\boldsymbol{g}}(\boldsymbol{x}))||^2 = \mathbb{E}^{\text{train}}||\boldsymbol{f}(\boldsymbol{z}) - \hat{\boldsymbol{f}}(\hat{\boldsymbol{g}}(\boldsymbol{f}(\boldsymbol{z})))||^2 = 0\,, \tag{7.47}$$

which implies that, for $\mathbb{P}_{\boldsymbol{z}}^{\text{train}}$-almost every $\boldsymbol{z} \in \mathcal{Z}^{\text{train}}$,

$$\boldsymbol{f}(\boldsymbol{z}) = \hat{\boldsymbol{f}}(\hat{\boldsymbol{g}}(\boldsymbol{f}(\boldsymbol{z})))\,.$$

But since the functions on both sides of the equations are continuous, the equality holds for all $\boldsymbol{z} \in \mathcal{Z}^{\text{train}}$. This implies that $\boldsymbol{f}(\mathcal{Z}^{\text{train}}) = \hat{\boldsymbol{f}} \circ \hat{\boldsymbol{g}} \circ \boldsymbol{f}(\mathcal{Z}^{\text{train}}) = \hat{\boldsymbol{f}}(\hat{\mathcal{Z}}^{\text{train}})$.

By Remark 7.3, the restrictions $\boldsymbol{f} : \mathcal{Z}^{\text{train}} \to \boldsymbol{f}(\mathcal{Z}^{\text{train}})$ and $\hat{\boldsymbol{f}} : \hat{\mathcal{Z}}^{\text{train}} \to \hat{\boldsymbol{f}}(\hat{\mathcal{Z}}^{\text{train}})$ are C^2-diffeomorphisms and, because $\boldsymbol{f}(\mathcal{Z}^{\text{train}}) = \hat{\boldsymbol{f}}(\hat{\mathcal{Z}}^{\text{train}})$, their composition $\boldsymbol{v} := \boldsymbol{f}^{-1} \circ \hat{\boldsymbol{f}} : \hat{\mathcal{Z}}^{\text{train}} \to \mathcal{Z}^{\text{train}}$ is a well defined C^2-diffeomorphism (since C^2-diffeomorphisms are closed under composition). ■

Theorem 7.1 (Local disentanglement via additive decoders)**.** *Suppose that the data-generating process satisfies Assumption 7.1, that the learned decoder $\hat{\boldsymbol{f}} : \mathbb{R}^{d_z} \to \mathbb{R}^{d_x}$ is a C^2-diffeomorphism, that the encoder $\hat{\boldsymbol{g}} : \mathbb{R}^{d_x} \to \mathbb{R}^{d_z}$ is continuous, that both $\boldsymbol{f}$ and $\hat{\boldsymbol{f}}$ are additive (Definition 7.1) and that $\boldsymbol{f}$ is sufficiently nonlinear as formalized by Assumption 7.2. Then, if $\hat{\boldsymbol{f}}$ and $\hat{\boldsymbol{g}}$ solve the reconstruction problem on the training distribution, i.e. $\mathbb{E}^{\text{train}}||\boldsymbol{x} - \hat{\boldsymbol{f}}(\hat{\boldsymbol{g}}(\boldsymbol{x}))||^2 = 0$, we have that $\hat{\boldsymbol{f}}$ is locally $\mathcal{B}$-disentangled w.r.t. $\boldsymbol{f}$ (Definition 7.4).*

Proof We can apply Proposition 7.3 and have that the map $\boldsymbol{v} := \boldsymbol{f}^{-1} \circ \hat{\boldsymbol{f}}$ is a C^2-diffeomorphism from $\hat{\mathcal{Z}}^{\text{train}}$ to $\mathcal{Z}^{\text{train}}$. This allows one to write

$$\boldsymbol{f} \circ \boldsymbol{v}(\boldsymbol{z}) = \hat{\boldsymbol{f}}(\boldsymbol{z}) \; \forall \boldsymbol{z} \in \hat{\mathcal{Z}}^{\text{train}} \tag{7.48}$$

$$\sum_{B \in \mathcal{B}} \boldsymbol{f}^{(B)}(\boldsymbol{v}_B(\boldsymbol{z})) = \sum_{B \in \mathcal{B}} \hat{\boldsymbol{f}}^{(B)}(\boldsymbol{z}_B) \; \forall \boldsymbol{z} \in \hat{\mathcal{Z}}^{\text{train}} \,. \tag{7.49}$$

Since $\mathcal{Z}^{\text{train}}$ is regularly closed and is diffeomorphic to $\hat{\mathcal{Z}}^{\text{train}}$, by Lemma 7.1, we must have that $\hat{\mathcal{Z}}^{\text{train}} \subseteq \overline{(\hat{\mathcal{Z}}^{\text{train}})^\circ}$. Moreover, the left and right hand side of (7.49) are C^2, which means they have uniquely defined first and second derivatives on $\overline{(\hat{\mathcal{Z}}^{\text{train}})^\circ}$ by Lemma 7.3. This means the derivatives are uniquely defined on $\hat{\mathcal{Z}}^{\text{train}}$.

Let $\boldsymbol{z} \in \hat{\mathcal{Z}}^{\text{train}}$. Choose some $J \in \mathcal{B}$ and some $j \in J$. Differentiate both sides of the above equation with respect to $\boldsymbol{z}_j$, which yields:

$$\sum_{B \in \mathcal{B}} \sum_{i \in B} D_i \boldsymbol{f}^{(B)}(\boldsymbol{v}_B(\boldsymbol{z})) D_j \boldsymbol{v}_i(\boldsymbol{z}) = D_j \hat{\boldsymbol{f}}^{(J)}(\boldsymbol{z}_J) \,. \tag{7.50}$$

Choose $J' \in \mathcal{B} \setminus \{J\}$ and $j' \in J'$. Differentiating the above w.r.t. $\boldsymbol{z}_{j'}$ yields

$$\begin{aligned}
&\sum_{B \in \mathcal{B}} \sum_{i \in B} \left[D_i \boldsymbol{f}^{(B)}(\boldsymbol{v}_B(\boldsymbol{z})) D^2_{j,j'} \boldsymbol{v}_i(\boldsymbol{z}) + \sum_{i' \in B} D^2_{i,i'} \boldsymbol{f}^{(B)}(\boldsymbol{v}_B(\boldsymbol{z})) D_{j'} \boldsymbol{v}_{i'}(\boldsymbol{z}) D_j \boldsymbol{v}_i(\boldsymbol{z}) \right] = 0 \\
&\sum_{B \in \mathcal{B}} \Bigg[\sum_{i \in B} \left[D_i \boldsymbol{f}^{(B)}(\boldsymbol{v}_B(\boldsymbol{z})) D^2_{j,j'} \boldsymbol{v}_i(\boldsymbol{z}) + D^2_{i,i} \boldsymbol{f}^{(B)}(\boldsymbol{v}_B(\boldsymbol{z})) D_{j'} \boldsymbol{v}_i(\boldsymbol{z}) D_j \boldsymbol{v}_i(\boldsymbol{z}) \right] + \\
&\qquad \sum_{(i,i') \in B^2_<} D^2_{i,i'} \boldsymbol{f}^{(B)}(\boldsymbol{v}_B(\boldsymbol{z})) (D_{j'} \boldsymbol{v}_{i'}(\boldsymbol{z}) D_j \boldsymbol{v}_i(\boldsymbol{z}) + D_{j'} \boldsymbol{v}_i(\boldsymbol{z}) D_j \boldsymbol{v}_{i'}(\boldsymbol{z})) \Bigg] = 0 \,,
\end{aligned} \tag{7.51}$$

where $B^2_< := B^2 \cap \{(i,i') \mid i' < i\}$. For the sake of notational conciseness, we are going to refer to $S_{\mathcal{B}}$ and $S^c_{\mathcal{B}}$ as S and S^c (Definition 7.11). Also, define

$$S_< := \bigcup_{B \in \mathcal{B}} B^2_< \,. \tag{7.52}$$

Let us define the vectors

$$\forall i \in \{1, ...d_z\},\ \vec{a}_i(\boldsymbol{z}) := (D^2_{j,j'} \boldsymbol{v}_i(\boldsymbol{z}))_{(j,j') \in S^c} \tag{7.53}$$

$$\forall i \in \{1, ...d_z\},\ \vec{b}_i(\boldsymbol{z}) := (D_{j'} \boldsymbol{v}_i(\boldsymbol{z}) D_j \boldsymbol{v}_i(\boldsymbol{z}))_{(j,j') \in S^c} \tag{7.54}$$

$$\forall B \in \mathcal{B},\ \forall (i,i') \in B^2_<,\ \vec{c}_{i,i'}(\boldsymbol{z}) := (D_{j'} \boldsymbol{v}_{i'}(\boldsymbol{z}) D_j \boldsymbol{v}_i(\boldsymbol{z}) + D_{j'} \boldsymbol{v}_i(\boldsymbol{z}) D_j \boldsymbol{v}_{i'}(\boldsymbol{z}))_{(j,j') \in S^c} \tag{7.55}$$

This allows us to rewrite, for all $k \in \{1, ..., d_x\}$

$$\sum_{B \in \mathcal{B}} \left[\sum_{i \in B} \left[D_i \boldsymbol{f}^{(B)}_k(\boldsymbol{v}_B(\boldsymbol{z})) \vec{a}_i(\boldsymbol{z}) + D^2_{i,i} \boldsymbol{f}^{(B)}_k(\boldsymbol{v}_B(\boldsymbol{z})) \vec{b}_i(\boldsymbol{z}) \right] + \sum_{(i,i') \in B^2_<} D^2_{i,i'} \boldsymbol{f}^{(B)}_k(\boldsymbol{v}_B(\boldsymbol{z})) \vec{c}_{i,i'}(\boldsymbol{z}) \right] = 0 \,. \tag{7.56}$$

We define

$$\boldsymbol{w}(\boldsymbol{z}, k) := ((D_i \boldsymbol{f}^{(B)}_k(\boldsymbol{z}_B))_{i \in B}, (D^2_{i,i} \boldsymbol{f}^{(B)}_k(\boldsymbol{z}_B))_{i \in B}, (D^2_{i,i'} \boldsymbol{f}^{(B)}_k(\boldsymbol{z}_B))_{(i,i') \in B^2_<})_{B \in \mathcal{B}} \tag{7.57}$$

$$\boldsymbol{M}(\boldsymbol{z}) := [[\vec{a}_i(\boldsymbol{z})]_{i \in B}, [\vec{b}_i(\boldsymbol{z})]_{i \in B}, [\vec{c}_{i,i'}(\boldsymbol{z})]_{(i,i') \in B^2_<}]_{B \in \mathcal{B}} \,, \tag{7.58}$$

which allows us to write, for all $k \in \{1, ..., d_z\}$

$$\boldsymbol{M}(\boldsymbol{z}) \boldsymbol{w}(\boldsymbol{v}(\boldsymbol{z}), k) = 0 \,. \tag{7.59}$$

We can now recognize that the matrix $\boldsymbol{W}(\boldsymbol{v}(\boldsymbol{z}))$ of Assumption 7.2 is given by

$$\boldsymbol{W}(\boldsymbol{v}(\boldsymbol{z}))^\top = [\boldsymbol{w}(\boldsymbol{v}(\boldsymbol{z}), 1) \ \dots \ \boldsymbol{w}(\boldsymbol{v}(\boldsymbol{z}), d_x)] \tag{7.60}$$

which allows us to write

$$\boldsymbol{M}(\boldsymbol{z}) \boldsymbol{W}(\boldsymbol{v}(\boldsymbol{z}))^\top = 0 \tag{7.61}$$

$$\boldsymbol{W}(\boldsymbol{v}(\boldsymbol{z})) \boldsymbol{M}(\boldsymbol{z})^\top = 0 \tag{7.62}$$

Since $\boldsymbol{W}(\boldsymbol{v}(\boldsymbol{z}))$ has full column-rank (by Assumption 7.2 and the fact that $\boldsymbol{v}(\boldsymbol{z}) \in \mathcal{Z}^{\text{train}}$), there exists q rows that are linearly independent. Let K be the index set of these rows. This means $\boldsymbol{W}(\boldsymbol{v}(\boldsymbol{z}))_{K,\cdot}$ is an invertible matrix. We can thus write

$$\boldsymbol{W}(\boldsymbol{v}(\boldsymbol{z}))_{K,\cdot} \boldsymbol{M}(\boldsymbol{z})^\top = 0 \tag{7.63}$$

$$(\boldsymbol{W}(\boldsymbol{v}(\boldsymbol{z}))_{K,\cdot})^{-1} \boldsymbol{W}(\boldsymbol{v}(\boldsymbol{z}))_{K,\cdot} \boldsymbol{M}(\boldsymbol{z})^\top = (\boldsymbol{W}(\boldsymbol{v}(\boldsymbol{z}))_{K,\cdot})^{-1} 0 \tag{7.64}$$

$$\boldsymbol{M}(\boldsymbol{z})^\top = 0 \,, \tag{7.65}$$

which means, in particular, that, $\forall i \in \{1, \dots, d_z\}$, $\vec{b}_i(\boldsymbol{z}) = 0$, i.e.,

$$\forall i \in \{1, \dots, d_z\}, \forall (j, j') \in S^c, D_j \boldsymbol{v}_i(\boldsymbol{z}) D_{j'} \boldsymbol{v}_i(\boldsymbol{z}) = 0 \tag{7.66}$$

Since the $\boldsymbol{v}$ is a diffeomorphism, its Jacobian matrix $D\boldsymbol{v}(\boldsymbol{z})$ is invertible everywhere. By Lemma 7.2, this means there exists a permutation π such that, for all j, $D_j \boldsymbol{v}_{\pi(j)}(\boldsymbol{z}) \neq 0$. This and (7.66) imply that

$$\forall (j, j') \in S^c, \;\; D_j \boldsymbol{v}_{\pi(j')}(\boldsymbol{z}) \underbrace{D_{j'} \boldsymbol{v}_{\pi(j')}(\boldsymbol{z})}_{\neq 0} = 0, \tag{7.67}$$

$$\implies \forall (j, j') \in S^c, \;\; D_j \boldsymbol{v}_{\pi(j')}(\boldsymbol{z}) = 0 \,. \tag{7.68}$$

To show that $D\boldsymbol{v}(\boldsymbol{z})$ is a $\mathcal{B}$-block permutation matrix, the only thing left to show is that π respects $\mathcal{B}$. For this, we use the fact that, $\forall B \in \mathcal{B}, \forall (i, i') \in B^2_<$, $\vec{c}_{i,i'}(\boldsymbol{z}) = 0$ (recall $\boldsymbol{M}(\boldsymbol{z}) = 0$). Because $\vec{c}_{i,i'}(\boldsymbol{z}) = \vec{c}_{i',i}(\boldsymbol{z})$, we can write

$$\forall (i, i') \in S \text{ s.t. } i \neq i', \forall (j, j') \in S^c, D_{j'} \boldsymbol{v}_{i'}(\boldsymbol{z}) D_j \boldsymbol{v}_i(\boldsymbol{z}) + D_{j'} \boldsymbol{v}_i(\boldsymbol{z}) D_j \boldsymbol{v}_{i'}(\boldsymbol{z}) = 0 \,. \tag{7.69}$$

We now show that if $(j, j') \in S^c$ (indices belong to different blocks), then $(\pi(j), \pi(j')) \in S^c$ (they also belong to different blocks). Assume this is false, i.e. there exists $(j_0, j_0') \in S^c$ such that $(\pi(j_0), \pi(j_0')) \in S$. Then we can apply (7.69) (with $i := \pi(j_0)$ and $i' := \pi(j_0')$) and get

$$\underbrace{D_{j_0'} \boldsymbol{v}_{\pi(j_0')}(\boldsymbol{z}) D_{j_0} \boldsymbol{v}_{\pi(j_0)}(\boldsymbol{z})}_{\neq 0} + D_{j_0'} \boldsymbol{v}_{\pi(j_0)}(\boldsymbol{z}) D_{j_0} \boldsymbol{v}_{\pi(j_0')}(\boldsymbol{z}) = 0 \,, \tag{7.70}$$

where the left term in the sum is different of 0 because of the definition of π. This implies that

$$D_{j_0'} \boldsymbol{v}_{\pi(j_0)}(\boldsymbol{z}) D_{j_0} \boldsymbol{v}_{\pi(j_0')}(\boldsymbol{z}) \neq 0 \,, \tag{7.71}$$

otherwise (7.70) cannot hold. But (7.71) contradicts (7.68). Thus, we have that,

$$(j, j') \in S^c \implies (\pi(j), \pi(j')) \in S^c \,. \tag{7.72}$$

The contraposed is

$$(\pi(j), \pi(j')) \in S \implies (j, j') \in S \tag{7.73}$$

$$(j, j') \in S \implies (\pi^{-1}(j), \pi^{-1}(j')) \in S \,. \tag{7.74}$$

From the above, it is clear that π^{-1} respects $\mathcal{B}$ which implies that π respects $\mathcal{B}$ (Lemma 7.4). Thus $D\boldsymbol{v}(\boldsymbol{z})$ is a $\mathcal{B}$-block permutation matrix. ■

Lemma 7.4 ($\mathcal{B}$-respecting permutations form a group)**.** *Let $\mathcal{B}$ be a partition of $\{1, \dots, d_z\}$ and let π and $\bar{\pi}$ be a permutation of $\{1, \dots, d_z\}$ that respect $\mathcal{B}$. The following holds:*

(1) The identity permutation e respects $\mathcal{B}$.

(2) The composition $\pi \circ \bar{\pi}$ respects $\mathcal{B}$.

(3) The inverse permutation π^{-1} respects $\mathcal{B}$.

Proof The first statement is trivial, since for all $B \in \mathcal{B}$, $e(B) = B \in \mathcal{B}$.

The second statement follows since for all $B \in \mathcal{B}$, $\bar{\pi}(B) \in \mathcal{B}$ and thus $\pi(\bar{\pi}(B)) \in \mathcal{B}$.

We now prove the third statement. Let $B \in \mathcal{B}$. Since π is surjective and respects $\mathcal{B}$, there exists a $B' \in \mathcal{B}$ such that $\pi(B') = B$. Thus, $\pi^{-1}(B) = \pi^{-1}(\pi(B')) = B' \in \mathcal{B}$. ■

A.6. Sufficient nonlinearity v.s. sufficient variability in nonlinear ICA with auxiliary variables

In Section 7.3.1, we introduced the "sufficient nonlinearity" condition (Assumption 7.2) and highlighted its resemblance to the "sufficient variability" assumptions often found in the nonlinear ICA literature [Hyvarinen and Morioka, 2016, 2017, Hyvärinen et al., 2019, Khemakhem et al., 2020a,b, Lachapelle et al., 2022, Zheng et al., 2022]. We now clarify this connection. To make the discussion more concrete, we consider the sufficient variability assumption found in Hyvärinen et al. [2019]. In this work, the latent variable $\boldsymbol{z}$ is assumed to be distributed according to

$$p(\boldsymbol{z} \mid \boldsymbol{u}) := \prod_{i=1}^{d_z} p_i(\boldsymbol{z}_i \mid \boldsymbol{u}) \,. \tag{7.75}$$

In other words, the latent factors $\boldsymbol{z}_i$ are mutually conditionally independent given an observed auxiliary variable $\boldsymbol{u}$. Define

$$\boldsymbol{w}(\boldsymbol{z}, \boldsymbol{u}) := \left(\left(\frac{\partial}{\partial \boldsymbol{z}_i} \log p_i(\boldsymbol{z}_i \mid \boldsymbol{u}) \right)_{i \in [d_z]} \left(\frac{\partial^2}{\partial \boldsymbol{z}_i^2} \log p_i(\boldsymbol{z}_i \mid \boldsymbol{u}) \right)_{i \in [d_z]} \right) \in \mathbb{R}^{2d_z} \,. \tag{7.76}$$

We now recall the assumption of sufficient variability of Hyvärinen et al. [2019]:

Assumption 7.3 (Assumption of variability from Hyvärinen et al. [2019, Theorem 1])**.** *For any $\boldsymbol{z} \in \mathbb{R}^{d_z}$, there exists $2d_z + 1$ values of $\boldsymbol{u}$, denoted by $\boldsymbol{u}^{(0)}, \boldsymbol{u}^{(1)}, \ldots, \boldsymbol{u}^{(2d_z)}$ such that the $2d_z$ vectors*

$$\boldsymbol{w}(\boldsymbol{z}, \boldsymbol{u}^{(1)}) - \boldsymbol{w}(\boldsymbol{z}, \boldsymbol{u}^{(0)}), \ldots, \boldsymbol{w}(\boldsymbol{z}, \boldsymbol{u}^{(2d_z)}) - \boldsymbol{w}(\boldsymbol{z}, \boldsymbol{u}^{(0)}) \tag{7.77}$$

are linearly independent.

To emphasize the resemblance with our assumption of sufficient nonlinearity, we rewrite it in the special case where the partition $\mathcal{B} := \{\{1\}, \ldots, \{d_z\}\}$. Note that, in that case, $q := d_z + \sum_{B \in \mathcal{B}} \frac{|B|(|B|+1)}{2} = 2d_z$.

Assumption 7.4 (Sufficient nonlinearity (trivial partition))**.** *For all $\boldsymbol{z} \in \mathcal{Z}^{\text{train}}$, $\boldsymbol{f}$ is such that the following matrix has independent columns (i.e. full column-rank):*

$$\boldsymbol{W}(\boldsymbol{z}) := \left[\left[D_i \boldsymbol{f}^{(i)}(\boldsymbol{z}_i) \right]_{i \in [d_z]} \left[D^2_{i,i} \boldsymbol{f}^{(i)}(\boldsymbol{z}_i) \right]_{i \in [d_z]} \right] \in \mathbb{R}^{d_x \times 2d_z} . \tag{7.78}$$

One can already see the resemblance between Assumptions 7.3 & 7.4, e.g. both have something to do with first and second derivatives. To make the connection even more explicit, define $\boldsymbol{w}(\boldsymbol{z}, k)$ to be the kth row of $\boldsymbol{W}(\boldsymbol{z})$ (do not conflate with $\boldsymbol{w}(\boldsymbol{z}, \boldsymbol{u})$). Also, recall the basic fact from linear algebra that the column-rank is always equal to the row-rank. This means that $\boldsymbol{W}(\boldsymbol{z})$ is full column-rank if and only if there exists k_1, ..., $k_{2d_z} \in [d_x]$ such that the vectors $\boldsymbol{w}(\boldsymbol{z}, k_1), \ldots, \boldsymbol{w}(\boldsymbol{z}, k_{2d_z})$ are linearly independent. It is then easy to see the correspondance between $\boldsymbol{w}(\boldsymbol{z}, k)$ and $\boldsymbol{w}(\boldsymbol{z}, \boldsymbol{u}) - \boldsymbol{w}(\boldsymbol{z}, \boldsymbol{u}^{(0)})$ (from Assumption 7.3) and between the pixel index $k \in [d_x]$ and the auxiliary variable $\boldsymbol{u}$.

We now look at why Assumption 7.2 is likely to be satisfied when $d_x >> d_z$. Informally, one can see that when d_x is much larger than $2d_z$, the matrix $\boldsymbol{W}(z)$ has much more rows than columns and thus it becomes more likely that we will find $2d_z$ rows that are linearly independent, thus satisfying Assumption 7.2.

A.7. Examples of sufficiently nonlinear additive decoders

Example 7.8 (A sufficiently nonlinear $\boldsymbol{f}$ - Example 7.3 continued)**.** *Consider the additive function*

$$\boldsymbol{f}(\boldsymbol{z}) := \begin{bmatrix} \boldsymbol{z}_1 \\ \boldsymbol{z}_1^2 \\ \boldsymbol{z}_1^3 \\ \boldsymbol{z}_1^4 \end{bmatrix} + \begin{bmatrix} (\boldsymbol{z}_2 + 1) \\ (\boldsymbol{z}_2 + 1)^2 \\ (\boldsymbol{z}_2 + 1)^3 \\ (\boldsymbol{z}_2 + 1)^4 \end{bmatrix} . \tag{7.79}$$

We will provide a numerical verification that this function is a diffeomorphism from the square $[-1, 0] \times [0, 1]$ to its image that satisfies Assumption 7.2.

The Jacobian of $\boldsymbol{f}$ is given by

$$D\boldsymbol{f}(\boldsymbol{z}) = \begin{bmatrix} 1 & 1 \\ 2\boldsymbol{z}_1 & 2(\boldsymbol{z}_2 + 1) \\ 3\boldsymbol{z}_1^2 & 3(\boldsymbol{z}_2 + 1)^2 \\ 4\boldsymbol{z}_1^3 & 4(\boldsymbol{z}_2 + 1)^3 \end{bmatrix} , \tag{7.80}$$

and the matrix $\boldsymbol{W}(\boldsymbol{z})$ from Assumption 7.2 is given by

$$\boldsymbol{W}(\boldsymbol{z}) = \begin{bmatrix} 1 & 0 & 1 & 0 \\ 2\boldsymbol{z}_1 & 2 & 2(\boldsymbol{z}_2 + 1) & 2 \\ 3\boldsymbol{z}_1^2 & 6\boldsymbol{z}_1 & 3(\boldsymbol{z}_2 + 1)^2 & 6(\boldsymbol{z}_2 + 1) \\ 4\boldsymbol{z}_1^3 & 12\boldsymbol{z}_1^2 & 4(\boldsymbol{z}_2 + 1)^3 & 12(\boldsymbol{z}_2 + 1)^2 \end{bmatrix} . \tag{7.81}$$

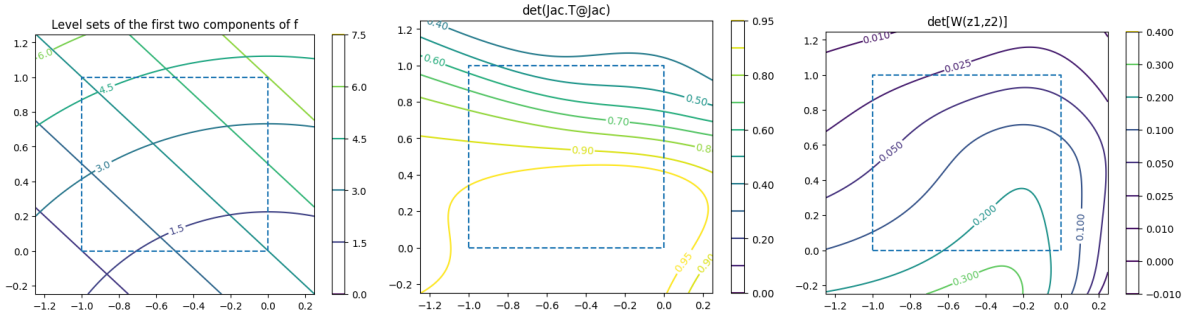


Figure 7.7. Numerical verification that $\boldsymbol{f} : [-1, 0] \times [0, 1] \to \mathbb{R}^4$ from Example 7.8 is injective (**left**), has a full rank Jacobian (**middle**) and satisfies Assumption 7.2 (**right**). The **left** figure shows that $\boldsymbol{f}$ is injective on the square $[-1, 0] \times [0, 1]$ since one can recover $\boldsymbol{z}$ uniquely by knowing the values of $\boldsymbol{f}_1(\boldsymbol{z})$ and $\boldsymbol{f}_2(\boldsymbol{z})$, i.e. knowing the level sets. The **middle** figure reports the $\det(D\boldsymbol{f}(\boldsymbol{z})^\top D\boldsymbol{f}(\boldsymbol{z}))$ (columns of the Jacobian are normalized to have norm 1) and shows that it is nonzero in the square $[-1, 0] \times [0, 1]$, which means the Jacobian is full rank. The **right** figure shows the determinant of the matrix $\boldsymbol{W}(\boldsymbol{z})$ (from Assumption 7.2, but with normalized columns), we can see that it is nonzero everywhere on the square $[-1, 0] \times [0, 1]$. We normalized the columns of $D\boldsymbol{f}$ and $\boldsymbol{W}$ so that the determinant is between 0 and 1.

Figure 7.7 presents a numerical verification that $\boldsymbol{f}$ is injective, has a full rank Jacobian and satisfies Assumption 7.2. Injective $\boldsymbol{f}$ with full rank Jacobian is enough to conclude that $\boldsymbol{f}$ is a diffeomorphism onto its image.

Example 7.9 (Smooth balls dataset is sufficiently nonlinear - Example 7.4 continued)**.** *We implemented a ground-truth additive decoder $\boldsymbol{f} : [0, 5]^2 \to \mathbb{R}^{64*64*3}$ which maps to 64x64 RGB images consisting of two colored balls where $\boldsymbol{z}_1$ and $\boldsymbol{z}_2$ control their respective heights (Figure 7.8a). The analytical form of $\boldsymbol{f}$ can be found in our code base. The decoder $\boldsymbol{f}$ is implemented in JAX [Bradbury et al., 2018] which allows for its automatic differentiation to compute $D\boldsymbol{f}$ and $D^2\boldsymbol{f}$ (Figures 7.8b & 7.8c). This allows us to verify numerically that $\boldsymbol{f}$ is sufficiently nonlinear (Assumption 7.2). Recall that this assumption requires that $\boldsymbol{W}(\boldsymbol{z})$ (defined in Assumption 7.2) has independent columns everywhere. To test this, we compute $Vol(\boldsymbol{z}) := \sqrt{|\det(\boldsymbol{W}(\boldsymbol{z})^\top \boldsymbol{W}(\boldsymbol{z}))|}$ over a grid of values of $\boldsymbol{z}$ and verify that $Vol(\boldsymbol{z}) > 0$ everywhere (Figure 7.8d). Note that $Vol(\boldsymbol{z})$ corresponds to the $4D$ volume of the parallelepiped embedded in $\mathbb{R}^{64*64*3}$ spanned by the four columns of $\boldsymbol{W}(\boldsymbol{z})$. This volume is > 0 if and only if the columns are linearly independent. Note that we normalize the columns of $\boldsymbol{W}(\boldsymbol{z})$ so that they have a norm of one. It follows that $Vol(\boldsymbol{z})$ is between 0 and 1 where 1 means the vectors are orthogonal, i.e. maximally independent. The minimal value of $Vol(\boldsymbol{z})$ over the domain of $\boldsymbol{f}$ is ≈ 0.97, indicating that Assumption 7.2 holds.*

A.8. Proof of Theorem 7.2

We start with a simple definition:

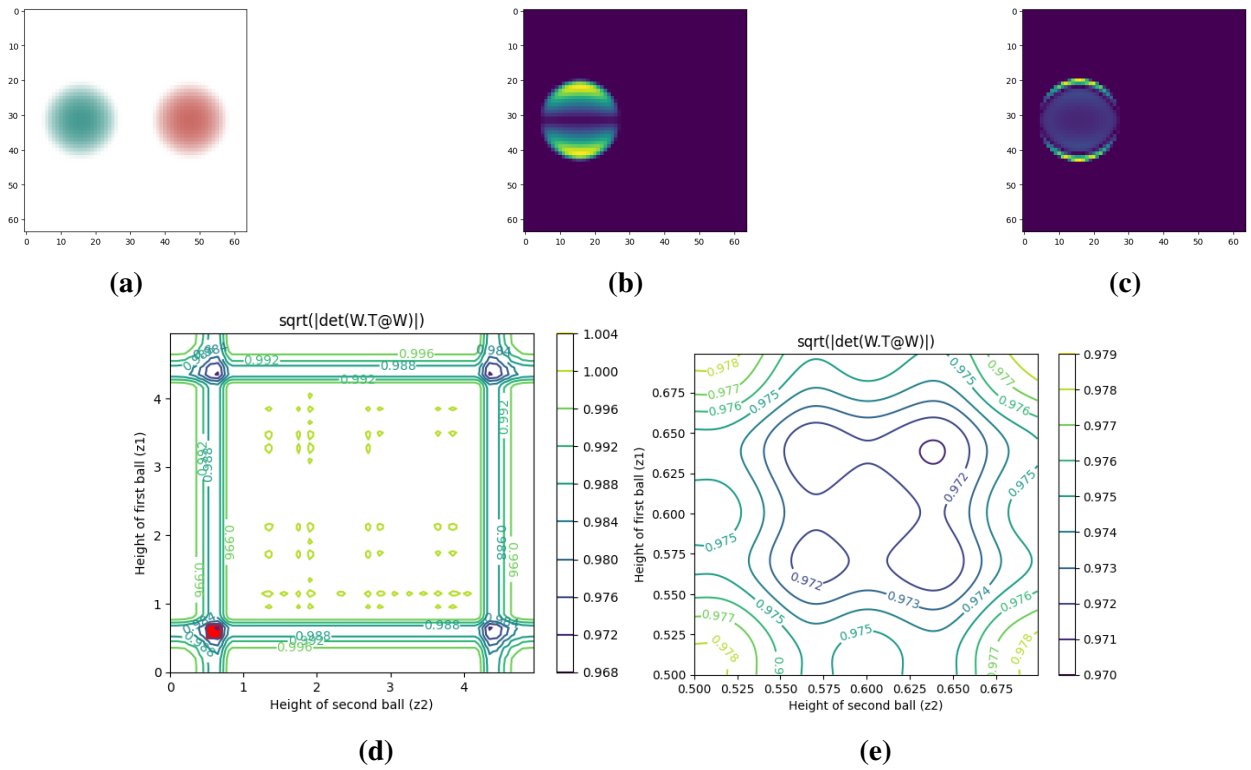


Figure 7.8. Figure (a) shows an image the synthetic dataset of Example 7.9. Figure (b) shows the derivative of the image w.r.t. $\boldsymbol{z}_1$ (the height of the left ball) where the color intensity of each pixel corresponds to the Euclidean norm along the RGB axis. Figure (c) similarly shows the second derivative of the image w.r.t. $\boldsymbol{z}_1$. Figure (d) is a contour plot of the function $\sqrt{|\det(\boldsymbol{W}(\boldsymbol{z})^\top \boldsymbol{W}(\boldsymbol{z}))|}$ where $\boldsymbol{W}(\boldsymbol{z})$ is defined in Assumption 7.2 (here columns are normalized to have unit norm). The smallest value of $\sqrt{|\det(\boldsymbol{W}(\boldsymbol{z})^\top \boldsymbol{W}(\boldsymbol{z}))|}$ across domain is ≈ 0.97, indicating that Assumption 2 is satisfied. See Example 7.9 and code for details. Figure 7.8e is a higher resolution rendering of the red region of Figure 7.8d (to make sure there is no singularity there).

Definition 7.15 ($\mathcal{B}$-block permutation matrices)**.** *A matrix $\boldsymbol{A} \in \mathbb{R}^{d\times d}$ is a $\mathcal{B}$-block permutation matrix if it is invertible and can be written as $\boldsymbol{A} = \boldsymbol{C}\boldsymbol{P}_\pi$ where $\boldsymbol{P}_\pi$ is the matrix representing the $\mathcal{B}$-respecting permutation π ($P_\pi \boldsymbol{e}_i = \boldsymbol{e}_{\pi(i)}$) and $\boldsymbol{C} \in \mathbb{R}^{d\times d}_{S_\mathcal{B}}$ (See Definitions 7.10 & 7.11).*

The following technical lemma leverages continuity and path-connectedness to show that the block-permutation structure must remain the same across the whole domain. It can be skipped at first read.

Lemma 7.5. *Let $\mathcal{C}$ be a connected topological space and let $\boldsymbol{M} : \mathcal{C} \to \mathbb{R}^{d\times d}$ be a continuous function. Suppose that, for all $c \in \mathcal{C}$, $\boldsymbol{M}(c)$ is an invertible $\mathcal{B}$-block permutation matrix (Definition 7.15). Then, there exists a $\mathcal{B}$-respecting permutation π such that for all $c \in \mathcal{C}$ and all distinct $B, B' \in \mathcal{B}$, $\boldsymbol{M}(c)_{\pi(B'),B} = 0$.*

Proof The reason this result is not trivial, is that, even if $\boldsymbol{M}(c)$ is a $\mathcal{B}$-block permutation for all c, the permutation might change for different c. The goal of this lemma is to show that, if $\mathcal{C}$ is

connected and the map $\boldsymbol{M}(\cdot)$ is continuous, then one can find a single permutation that works for all $c \in \mathcal{C}$.

First, since $\mathcal{C}$ is connected and $\boldsymbol{M}$ is continuous, its image, $\boldsymbol{M}(\mathcal{C})$, must be connected (by [Munkres, 2000, Theorem 23.5]).

Second, from the hypothesis of the lemma, we know that

$$\boldsymbol{M}(\mathcal{C}) \subseteq \mathcal{A} := \left(\bigcup_{\pi \in \mathfrak{S}(\mathcal{B})} \mathbb{R}^{d \times d}_{S_{\mathcal{B}}} \boldsymbol{P}_\pi \right) \setminus \{\text{singular matrices}\}\,, \tag{7.82}$$

where $\mathfrak{S}(\mathcal{B})$ is the set of $\mathcal{B}$-respecting permutations and $\mathbb{R}^{d \times d}_{S_{\mathcal{B}}} \boldsymbol{P}_\pi = \{\boldsymbol{M}\boldsymbol{P}_\pi \mid \boldsymbol{M} \in \mathbb{R}^{d \times d}_{S_{\mathcal{B}}}\}$. We can rewrite the set $\mathcal{A}$ above as

$$\mathcal{A} = \bigcup_{\pi \in \mathfrak{S}(\mathcal{B})} \left(\mathbb{R}^{d \times d}_{S_{\mathcal{B}}} \boldsymbol{P}_\pi \setminus \{\text{singular matrices}\} \right)\,, \tag{7.83}$$

We now define an equivalence relation $\sim$ over $\mathcal{B}$-respecting permutation: $\pi \sim \pi'$ iff for all $B \in \mathcal{B}$, $\pi(B) = \pi'(B)$. In other words, two $\mathcal{B}$-respecting permutations are equivalent if they send every block to the same block (note that they can permute elements of a given block differently). We notice that

$$\pi \sim \pi' \implies \mathbb{R}^{d \times d}_{S_{\mathcal{B}}} \boldsymbol{P}_\pi = \mathbb{R}^{d \times d}_{S_{\mathcal{B}}} \boldsymbol{P}_{\pi'}\,. \tag{7.84}$$

Let $\mathfrak{S}(\mathcal{B})/\sim$ be the set of equivalence classes induce by $\sim$ and let Π stand for one such equivalence class. Thanks to (7.84), we can define, for all $\Pi \in \mathfrak{S}(\mathcal{B})/\sim$, the following set:

$$V_\Pi := \mathbb{R}^{d \times d}_{S_{\mathcal{B}}} \boldsymbol{P}_\pi \setminus \{\text{singular matrices}\},\ \text{for some } \pi \in \Pi\,, \tag{7.85}$$

where the specific choice of $\pi \in \Pi$ is arbitrary (any $\pi' \in \Pi$ would yield the same definition, by (7.84)). This construction allows us to write

$$\mathcal{A} = \bigcup_{\Pi \in \mathfrak{S}(\mathcal{B})/\sim} V_\Pi\,, \tag{7.86}$$

We now show that $\{V_\Pi\}_{\Pi \in \mathfrak{S}(\mathcal{B})/\sim}$ forms a partition of $\mathcal{A}$. Choose two distinct equivalence classes of permutations Π and Π' and let $\pi \in \Pi$ and $\pi' \in \Pi'$ be representatives. We note that

$$\mathbb{R}^{d \times d}_{S_{\mathcal{B}}} \boldsymbol{P}_\pi \cap \mathbb{R}^{d \times d}_{S_{\mathcal{B}}} \boldsymbol{P}_{\pi'} \subseteq \{\text{singular matrices}\}\,, \tag{7.87}$$

since any matrix that is both in $\mathbb{R}^{d \times d}_{S_{\mathcal{B}}} \boldsymbol{P}_\pi$ and $\mathbb{R}^{d \times d}_{S_{\mathcal{B}}} \boldsymbol{P}_{\pi'}$ must have at least one row filled with zeros. This implies that

$$V_\Pi \cap V_{\Pi'} = \emptyset\,, \tag{7.88}$$

which shows that $\{V_\Pi\}_{\Pi \in \mathfrak{S}(\mathcal{B})/\sim}$ is indeed a partition of $\mathcal{A}$.

Each V_Π is closed in $\mathcal{A}$ (wrt the relative topology) since

$$V_\Pi = \mathbb{R}^{d\times d}_{S_\mathcal{B}} \boldsymbol{P}_\pi \setminus \{\text{singular matrices}\} = \mathcal{A} \cap \underbrace{\mathbb{R}^{d\times d}_{S_\mathcal{B}} \boldsymbol{P}_\pi}_{\text{closed in } \mathbb{R}^{d\times d}} . \tag{7.89}$$

Moreover, V_Π is open in $\mathcal{A}$, since

$$V_\Pi = \mathcal{A} \setminus \underbrace{\bigcup_{\Pi' \neq \Pi} V_{\Pi'}}_{\text{closed in } \mathcal{A}} . \tag{7.90}$$

Thus, for any $\Pi \in \mathfrak{S}(\mathcal{B})/\sim$, the sets V_Π and $\bigcup_{\Pi' \neq \Pi} V_{\Pi'}$ forms a *separation* (see [Munkres, 2000, Section 23]). Since $\boldsymbol{M}(\mathcal{C})$ is a connected subset of $\mathcal{A}$, it must lie completely in V_Π or $\bigcup_{\Pi' \neq \Pi} V_{\Pi'}$, by [Munkres, 2000, Lemma 23.2]. Since this is true for all Π, it must follow that there exists a Π^* such that $\boldsymbol{M}(\mathcal{C}) \subseteq V_{\Pi^*}$, which completes the proof. ■

Theorem 7.2 (From local to global disentanglement)**.** *Suppose that all the assumptions of Theorem 7.1 hold. Additionally, assume $\mathcal{Z}^{\text{train}}$ is path-connected (Definition 7.8) and that the block-specific decoders $\boldsymbol{f}^{(B)}$ and $\hat{\boldsymbol{f}}^{(B)}$ are injective for all blocks $B \in \mathcal{B}$. Then, if $\hat{\boldsymbol{f}}$ and $\hat{\boldsymbol{g}}$ solve the reconstruction problem on the training distribution, i.e. $\mathbb{E}^{\text{train}}||\boldsymbol{x} - \hat{\boldsymbol{f}}(\hat{\boldsymbol{g}}(\boldsymbol{x}))||^2 = 0$, we have that $\hat{\boldsymbol{f}}$ is (globally) $\mathcal{B}$-disentangled w.r.t. $\boldsymbol{f}$ (Definition 7.3) and, for all $B \in \mathcal{B}$,*

$$\hat{\boldsymbol{f}}^{(B)}(\boldsymbol{z}_B) = \boldsymbol{f}^{(\pi(B))}(\bar{\boldsymbol{v}}_{\pi(B)}(\boldsymbol{z}_B)) + \boldsymbol{c}^{(B)}, \text{ for all } \boldsymbol{z}_B \in \hat{\mathcal{Z}}^{\text{train}}_B , \tag{7.8}$$

where the functions $\bar{\boldsymbol{v}}_{\pi(B)}$ are from Defintion 7.3 and the vectors $\boldsymbol{c}^{(B)} \in \mathbb{R}^{d_x}$ are constants such that $\sum_{B\in\mathcal{B}} \boldsymbol{c}^{(B)} = 0$. We also have that the functions $\bar{\boldsymbol{v}}_{\pi(B)} : \hat{\mathcal{Z}}^{\text{train}}_B \to \mathcal{Z}^{\text{train}}_{\pi(B)}$ are C^2-diffeomorphisms and have the following form:

$$\bar{\boldsymbol{v}}_{\pi(B)}(\boldsymbol{z}_B) = (\boldsymbol{f}^{\pi(B)})^{-1}(\hat{\boldsymbol{f}}^{(B)}(\boldsymbol{z}_B) - \boldsymbol{c}^{(B)}), \text{ for all } \boldsymbol{z}_B \in \hat{\mathcal{Z}}^{\text{train}}_B . \tag{7.9}$$

Proof

Step 1 - Showing the permutation π does not change for different $\boldsymbol{z}$. Theorem 7.1 showed local $\mathcal{B}$-disentanglement, i.e. for all $\boldsymbol{z} \in \hat{\mathcal{Z}}^{\text{train}}$, $D\boldsymbol{v}(\boldsymbol{z})$ has a $\mathcal{B}$-block permutation structure. The first step towards showing global disentanglement is to show that this block structure is the same for all $\boldsymbol{z} \in \hat{\mathcal{Z}}^{\text{train}}$ (*a priori*, π could be different for different $\boldsymbol{z}$). Since $\boldsymbol{v}$ is C^2, its Jacobian $D\boldsymbol{v}(\boldsymbol{z})$ is continuous. Since $\mathcal{Z}^{\text{train}}$ is path-connected, $\hat{\mathcal{Z}}^{\text{train}}$ must also be since both sets are diffeomorphic. By Lemma 7.5, this means the $\mathcal{B}$-block permutation structure of $D\boldsymbol{v}(\boldsymbol{z})$ is the same for all $\boldsymbol{z} \in \hat{\mathcal{Z}}^{\text{train}}$ (implicitly using the fact that path-connected implies connected). In other words, there exists a permutation π respecting $\mathcal{B}$ such that, for all $\boldsymbol{z} \in \hat{\mathcal{Z}}^{\text{train}}$ and all distinct $B, B' \in \mathcal{B}$, $D_B \boldsymbol{v}_{\pi(B')}(\boldsymbol{z}) = 0$.

Step 2 - Linking object-specific decoders. We now show that, for all $B \in \mathcal{B}$, $\hat{\boldsymbol{f}}^{(B)}(\boldsymbol{z}_B) = \boldsymbol{f}^{(\pi(B))}(\boldsymbol{v}_{\pi(B)}(\boldsymbol{z})) + \boldsymbol{c}^{(B)}$ for all $\boldsymbol{z} \in \hat{\mathcal{Z}}^{\text{train}}$. To do this, we rewrite (7.50) as

$$D\hat{\boldsymbol{f}}^{(J)}(\boldsymbol{z}_J) = \sum_{B \in \mathcal{B}} D\boldsymbol{f}^{(B)}(\boldsymbol{v}_B(\boldsymbol{z}))D_J\boldsymbol{v}_B(\boldsymbol{z})\,, \tag{7.91}$$

but because $B \neq \pi(J) \implies D_J\boldsymbol{v}_B(\boldsymbol{z}) = 0$ (block-permutation structure), we get

$$D\hat{\boldsymbol{f}}^{(J)}(\boldsymbol{z}_J) = D\boldsymbol{f}^{(\pi(J))}(\boldsymbol{v}_{\pi(J)}(\boldsymbol{z}))D_J\boldsymbol{v}_{\pi(J)}(\boldsymbol{z})\,. \tag{7.92}$$

The above holds for all $J \in \mathcal{B}$. We simply change J by B in the following equation.

$$D\hat{\boldsymbol{f}}^{(B)}(\boldsymbol{z}_B) = D\boldsymbol{f}^{(\pi(B))}(\boldsymbol{v}_{\pi(B)}(\boldsymbol{z}))D_B\boldsymbol{v}_{\pi(B)}(\boldsymbol{z})\,. \tag{7.93}$$

Now notice that the r.h.s. of the above equation is equal to $D(\boldsymbol{f}^{(\pi(B))} \circ \boldsymbol{v}_{\pi(B)})$. We can thus write

$$D\hat{\boldsymbol{f}}^{(B)}(\boldsymbol{z}_B) = D(\boldsymbol{f}^{(\pi(B))} \circ \boldsymbol{v}_{\pi(B)})(\boldsymbol{z})\,, \text{for all } \boldsymbol{z} \in \hat{\mathcal{Z}}^{\text{train}}\,. \tag{7.94}$$

Now choose distinct $\boldsymbol{z}, \boldsymbol{z}^0 \in \hat{\mathcal{Z}}^{\text{train}}$. Since $\mathcal{Z}^{\text{train}}$ is path-connected, $\hat{\mathcal{Z}}^{\text{train}}$ also is since they are diffeomorphic. Hence, there exists a continuously differentiable function $\boldsymbol{\phi} : [0,1] \to \hat{\mathcal{Z}}^{\text{train}}$ such that $\boldsymbol{\phi}(0) = \boldsymbol{z}^0$ and $\boldsymbol{\phi}(1) = \boldsymbol{z}$. We can now use (7.94) together with the gradient theorem, a.k.a. the fundamental theorem of calculus for line integrals, to show the following

$$\int_0^1 D\hat{\boldsymbol{f}}^{(B)}(\boldsymbol{\phi}_B(\boldsymbol{z})) \cdot \boldsymbol{\phi}_B(t)dt = \int_0^1 D(\boldsymbol{f}^{(\pi(B))} \circ \boldsymbol{v}_{\pi(B)})(\boldsymbol{\phi}(\boldsymbol{z})) \cdot \boldsymbol{\phi}(t)dt \tag{7.95}$$

$$\hat{\boldsymbol{f}}^{(B)}(\boldsymbol{z}_B) - \hat{\boldsymbol{f}}^{(B)}(\boldsymbol{z}_B^0) = \boldsymbol{f}^{(\pi(B))} \circ \boldsymbol{v}_{\pi(B)}(\boldsymbol{z}) - \boldsymbol{f}^{(\pi(B))} \circ \boldsymbol{v}_{\pi(B)}(\boldsymbol{z}^0) \tag{7.96}$$

$$\hat{\boldsymbol{f}}^{(B)}(\boldsymbol{z}_B) = \boldsymbol{f}^{(\pi(B))} \circ \boldsymbol{v}_{\pi(B)}(\boldsymbol{z}) + \underbrace{(\hat{\boldsymbol{f}}^{(B)}(\boldsymbol{z}_B^0) - \boldsymbol{f}^{(\pi(B))} \circ \boldsymbol{v}_{\pi(B)}(\boldsymbol{z}^0))}_{\text{constant in } \boldsymbol{z}} \tag{7.97}$$

$$\hat{\boldsymbol{f}}^{(B)}(\boldsymbol{z}_B) = \boldsymbol{f}^{(\pi(B))} \circ \boldsymbol{v}_{\pi(B)}(\boldsymbol{z}) + \boldsymbol{c}^{(B)}\,, \tag{7.98}$$

which holds for all $\boldsymbol{z} \in \hat{\mathcal{Z}}^{\text{train}}$.

We now show that $\sum_{B \in \mathcal{B}} \boldsymbol{c}^{(B)} = 0$. Take some $\boldsymbol{z}^0 \in \hat{\mathcal{Z}}^{\text{train}}$. Equations (7.49) & (7.98) tell us that

$$\sum_{B \in \mathcal{B}} \boldsymbol{f}^{(B)}(\boldsymbol{v}_B(\boldsymbol{z}^0)) = \sum_{B \in \mathcal{B}} \hat{\boldsymbol{f}}^{(B)}(\boldsymbol{z}_B^0) \tag{7.99}$$

$$= \sum_{B \in \mathcal{B}} \boldsymbol{f}^{(\pi(B))}(\boldsymbol{v}_{\pi(B)}(\boldsymbol{z}^0)) + \sum_{B \in \mathcal{B}} \boldsymbol{c}^{(B)} \tag{7.100}$$

$$= \sum_{B \in \mathcal{B}} \boldsymbol{f}^{(B)}(\boldsymbol{v}_B(\boldsymbol{z}^0)) + \sum_{B \in \mathcal{B}} \boldsymbol{c}^{(B)} \tag{7.101}$$

$$\implies 0 = \sum_{B \in \mathcal{B}} \boldsymbol{c}^{(B)} \tag{7.102}$$

Step 3 - From local to global disentanglement. By assumption, the functions $\boldsymbol{f}^{(B)} : \mathcal{Z}_B^{\text{train}} \to \mathbb{R}^{d_x}$ are injective. This will allow us to show that $\boldsymbol{v}_{\pi(B)}(\boldsymbol{z})$ depends only on $\boldsymbol{z}_B$. We proceed by contradiction. Suppose there exists $(\boldsymbol{z}_B, \boldsymbol{z}_{B^c}) \in \hat{\mathcal{Z}}^{\text{train}}$ and $\boldsymbol{z}_{B^c}^0$ such that $(\boldsymbol{z}_B, \boldsymbol{z}_{B^c}^0) \in \hat{\mathcal{Z}}^{\text{train}}$ and $\boldsymbol{v}_{\pi(B)}(\boldsymbol{z}_B, \boldsymbol{z}_{B^c}) \neq \boldsymbol{v}_{\pi(B)}(\boldsymbol{z}_B, \boldsymbol{z}_{B^c}^0)$. This means

$$\boldsymbol{f}^{(\pi(B))} \circ \boldsymbol{v}_{\pi(B)}(\boldsymbol{z}_B, \boldsymbol{z}_{B^c}) + \boldsymbol{c}^{(B)} = \hat{\boldsymbol{f}}^{(B)}(\boldsymbol{z}_B) = \boldsymbol{f}^{(\pi(B))} \circ \boldsymbol{v}_{\pi(B)}(\boldsymbol{z}_B, \boldsymbol{z}_{B^c}^0) + \boldsymbol{c}^{(B)}$$

$$\boldsymbol{f}^{(\pi(B))}(\boldsymbol{v}_{\pi(B)}(\boldsymbol{z}_B, \boldsymbol{z}_B)) = \boldsymbol{f}^{(\pi(B))}(\boldsymbol{v}_{\pi(B)}(\boldsymbol{z}_B, \boldsymbol{z}_B^0))$$

which is a contradiction with the fact that $\boldsymbol{f}^{(\pi(B))}$ is injective. Hence, $\boldsymbol{v}_{\pi(B)}(\boldsymbol{z})$ depends only on $\boldsymbol{z}_B$. We also get an explicit form for $\boldsymbol{v}_{\pi(B)}$:

$$(\boldsymbol{f}^{\pi(B)})^{-1}(\hat{\boldsymbol{f}}^{(B)}(\boldsymbol{z}_B) - \boldsymbol{c}^{(B)}) = \boldsymbol{v}_{\pi(B)}(\boldsymbol{z}) \text{ for all } \boldsymbol{z} \in \mathcal{Z}^{\text{train}} . \tag{7.103}$$

We define the map $\bar{\boldsymbol{v}}_{\pi(B)}(\boldsymbol{z}_B) := (\boldsymbol{f}^{\pi(B)})^{-1}(\hat{\boldsymbol{f}}^{(B)}(\boldsymbol{z}_B) - \boldsymbol{c}^{(B)})$ which is from $\hat{\mathcal{Z}}_B^{\text{train}}$ to $\mathcal{Z}_{\pi(B)}^{\text{train}}$. This allows us to rewrite (7.98) as

$$\hat{\boldsymbol{f}}^{(B)}(\boldsymbol{z}_B) = \boldsymbol{f}^{(\pi(B))} \circ \bar{\boldsymbol{v}}_{\pi(B)}(\boldsymbol{z}_B) + \boldsymbol{c}^{(B)} , \text{ for all } \boldsymbol{z}_B \in \mathcal{Z}_B^{\text{train}} . \tag{7.104}$$

Because $\hat{\boldsymbol{f}}^{(B)}$ is also injective, we must have that $\bar{\boldsymbol{v}}_{\pi(B)} : \hat{\mathcal{Z}}_B^{\text{train}} \to \mathcal{Z}_{\pi(B)}^{\text{train}}$ is injective as well.

We now show that $\bar{\boldsymbol{v}}_{\pi(B)}$ is surjective. Choose some $\boldsymbol{z}_{\pi(B)} \in \mathcal{Z}_{\pi(B)}^{\text{train}}$. We can always find $\boldsymbol{z}_{\pi(B)^c}$ such that $(\boldsymbol{z}_{\pi(B)}, \boldsymbol{z}_{\pi(B)^c}) \in \mathcal{Z}^{\text{train}}$. Because $\boldsymbol{v} : \hat{\mathcal{Z}}^{\text{train}} \to \mathcal{Z}^{\text{train}}$ is surjective (it is a diffeomorphism), there exists a $\boldsymbol{z}^0 \in \hat{\mathcal{Z}}^{\text{train}}$ such that $\boldsymbol{v}(\boldsymbol{z}^0) = (\boldsymbol{z}_{\pi(B)}, \boldsymbol{z}_{\pi(B)^c})$. By (7.103), we have that

$$\bar{\boldsymbol{v}}_{\pi(B)}(\boldsymbol{z}_B^0) = \boldsymbol{v}_{\pi(B)}(\boldsymbol{z}^0) . \tag{7.105}$$

which means $\bar{\boldsymbol{v}}_{\pi(B)}(\boldsymbol{z}_B^0) = \boldsymbol{z}_{\pi(B)}$.

We thus have that $\bar{\boldsymbol{v}}_{\pi(B)}$ is bijective. It is a diffeomorphism because

$$\det D\bar{\boldsymbol{v}}_{\pi(B)}(\boldsymbol{z}_B) = \det D_B \boldsymbol{v}_{\pi(B)}(\boldsymbol{z}) \neq 0 \; \forall \boldsymbol{z} \in \hat{\mathcal{Z}}^{\text{train}} \tag{7.106}$$

where the first equality holds by (7.103) and the second holds because $\boldsymbol{v}$ is a diffeomorphism and has block-permutation structure, which means it has a nonzero determinant everywhere on $\hat{\mathcal{Z}}^{\text{train}}$ and is equal to the product of the determinants of its blocks, which implies each block $D_B \boldsymbol{v}_{\pi(B)}$ must have nonzero determinant everywhere.

Since $\bar{\boldsymbol{v}}_{\pi(B)} : \hat{\mathcal{Z}}_B^{\text{train}} \to \mathcal{Z}_{\pi(B)}^{\text{train}}$ bijective and has invertible Jacobian everywhere, it must be a diffeomorphism. ■

A.9. Injectivity of object-specific decoders v.s. injectivity of their sum

We want to explore the relationship between the injectivity of individual object-specific decoders $\boldsymbol{f}^{(B)}$ and the injectivity of their sum, i.e. $\sum_{B \in \mathcal{B}} \boldsymbol{f}^{(B)}$.

We first show the simple fact that having each $\boldsymbol{f}^{(B)}$ injective is not sufficient to have $\sum_{B\in\mathcal{B}} \boldsymbol{f}^{(B)}$ injective. Take $\boldsymbol{f}^{(B)}(\boldsymbol{z}_B) = \boldsymbol{W}^{(B)}\boldsymbol{z}_B$ where $\boldsymbol{W}^{(B)} \in \mathbb{R}^{d_x \times |B|}$ has full column-rank for all $B \in \mathcal{B}$. We have that

$$\sum_{B\in\mathcal{B}} \boldsymbol{f}^{(B)}(\boldsymbol{z}_B) = \sum_{B\in\mathcal{B}} \boldsymbol{W}^{(B)}\boldsymbol{z}_B = [\boldsymbol{W}^{(B_1)} \ \cdots \ \boldsymbol{W}^{(B_\ell)}]\boldsymbol{z}\,, \tag{7.107}$$

where it is clear that the matrix $[\boldsymbol{W}^{(B_1)} \ \cdots \ \boldsymbol{W}^{(B_\ell)}] \in \mathbb{R}^{d_x \times d_z}$ is not necessarily injective even if each $\boldsymbol{W}^{(B)}$ is. This is the case, for instance, if all $\boldsymbol{W}^{(B)}$ have the same image.

We now provide conditions such that $\sum_{B\in\mathcal{B}} \boldsymbol{f}^{(B)}$ injective implies each $\boldsymbol{f}^{(B)}$ injective. We start with a simple lemma:

Lemma 7.6. *If $g \circ h$ is injective, then h is injective.*

Proof By contradiction, assume that h is not injective. Then, there exists distinct $x_1, x_2 \in \mathrm{Dom}(h)$ such that $h(x_1) = h(x_2)$. This implies $g \circ h(x_1) = g \circ h(x_2)$, which violates injectivity of $g \circ h$. ■

The following Lemma provides a condition on the domain of the function $\sum_{B\in\mathcal{B}} \boldsymbol{f}^{(B)}$, $\mathcal{Z}^{\text{train}}$, so that its injectivity implies injectivity of the functions $\boldsymbol{f}^{(B)}$.

Lemma 7.7. *Assume that, for all $B \in \mathcal{B}$ and for all distinct $\boldsymbol{z}_B, \boldsymbol{z}'_B \in \mathcal{Z}^{\text{train}}_B$, there exists $\boldsymbol{z}_{B^c}$ such that $(\boldsymbol{z}_B, \boldsymbol{z}_{B^c}), (\boldsymbol{z}'_B, \boldsymbol{z}_{B^c}) \in \mathcal{Z}^{\text{train}}$. Then, whenever $\sum_{B\in\mathcal{B}} \boldsymbol{f}^{(B)}$ is injective, each $\boldsymbol{f}^{(B)}$ must be injective.*

Proof Notice that $\boldsymbol{f}(\boldsymbol{z}) := \sum_{B\in\mathcal{B}} \boldsymbol{f}^{(B)}(\boldsymbol{z}_B)$ can be written as $\boldsymbol{f} := \text{SumBlocks} \circ \bar{\boldsymbol{f}}(\boldsymbol{z})$ where

$$\bar{\boldsymbol{f}}(\boldsymbol{z}) := \begin{bmatrix} \boldsymbol{f}^{(B_1)}(\boldsymbol{z}_{B_1}) \\ \vdots \\ \boldsymbol{f}^{(B_\ell)}(\boldsymbol{z}_{B_\ell}) \end{bmatrix}, \text{ and } \text{SumBlocks}(\boldsymbol{x}^{(B_1)}, \ldots, \boldsymbol{x}^{(B_\ell)}) := \sum_{B\in\mathcal{B}} \boldsymbol{x}^{(B)} \tag{7.108}$$

Since $\boldsymbol{f}$ is injective, by Lemma 7.6 $\bar{\boldsymbol{f}}$ must be injective.

We now show that each $\boldsymbol{f}^{(B)}$ must also be injective. Take $\boldsymbol{z}_B, \boldsymbol{z}'_B \in \mathcal{Z}^{\text{train}}_B$ such that $\boldsymbol{f}^{(B)}(\boldsymbol{z}_B) = \boldsymbol{f}^{(B)}(\boldsymbol{z}'_B)$. By assumption, we know there exists a $\boldsymbol{z}_{B^c}$ s.t. $(\boldsymbol{z}_B, \boldsymbol{z}_{B^c})$ and $(\boldsymbol{z}'_B, \boldsymbol{z}_{B^c})$ are in $\mathcal{Z}^{\text{train}}$. By construction, we have that $\bar{\boldsymbol{f}}((\boldsymbol{z}_B, \boldsymbol{z}_{B^c})) = \bar{\boldsymbol{f}}((\boldsymbol{z}'_B, \boldsymbol{z}_{B^c}))$. By injectivity of $\bar{\boldsymbol{f}}$, we have that $(\boldsymbol{z}_B, \boldsymbol{z}_{B^c}) \neq (\boldsymbol{z}'_B, \boldsymbol{z}_{B^c})$, which implies $\boldsymbol{z}_B \neq \boldsymbol{z}'_B$, i.e. $\boldsymbol{f}^{(B)}$ is injective. ■

A.10. Proof of Corollary 7.1

Corollary 7.1 (Cartesian-product extrapolation)**.** *Suppose the assumptions of Theorem 7.2 holds. Then,*

$$\textit{for all } \boldsymbol{z} \in \text{CPE}_{\mathcal{B}}(\hat{\mathcal{Z}}^{\text{train}}),\ \sum_{B\in\mathcal{B}} \hat{\boldsymbol{f}}^{(B)}(\boldsymbol{z}_B) = \sum_{B\in\mathcal{B}} \boldsymbol{f}^{(\pi(B))}(\bar{\boldsymbol{v}}_{\pi(B)}(\boldsymbol{z}_B))\,. \tag{7.11}$$

Furthermore, if $\text{CPE}_{\mathcal{B}}(\mathcal{Z}^{\text{train}}) \subseteq \mathcal{Z}^{\text{test}}$*, then* $\hat{\boldsymbol{f}}(\text{CPE}_{\mathcal{B}}(\hat{\mathcal{Z}}^{\text{train}})) \subseteq \boldsymbol{f}(\mathcal{Z}^{\text{test}})$.

Proof Pick $\boldsymbol{z} \in \text{CPE}(\hat{\mathcal{Z}}^{\text{train}})$. By definition, this means that, for all $B \in \mathcal{B}$, $\boldsymbol{z}_B \in \hat{\mathcal{Z}}_B^{\text{train}}$. We thus have that, for all $B \in \mathcal{B}$,

$$\hat{\boldsymbol{f}}^{(B)}(\boldsymbol{z}_B) = \boldsymbol{f}^{(\pi(B))} \circ \bar{\boldsymbol{v}}_{\pi(B)}(\boldsymbol{z}_B) + \boldsymbol{c}^{(B)} \,. \tag{7.109}$$

We can thus sum over B to obtain

$$\sum_{B \in \mathcal{B}} \hat{\boldsymbol{f}}^{(B)}(\boldsymbol{z}_B) = \sum_{B \in \mathcal{B}} \boldsymbol{f}^{(\pi(B))} \circ \bar{\boldsymbol{v}}_{\pi(B)}(\boldsymbol{z}_B) + \underbrace{\sum_{B \in \mathcal{B}} \boldsymbol{c}^{(B)}}_{=0} \,. \tag{7.110}$$

Since $\boldsymbol{z} \in \text{CPE}(\hat{\mathcal{Z}}^{\text{train}})$ was arbitrary, we have

$$\text{for all } \boldsymbol{z} \in \text{CPE}(\hat{\mathcal{Z}}^{\text{train}}),\ \sum_{B \in \mathcal{B}} \hat{\boldsymbol{f}}^{(B)}(\boldsymbol{z}_B) = \sum_{B \in \mathcal{B}} \boldsymbol{f}^{(\pi(B))} \circ \bar{\boldsymbol{v}}_{\pi(B)}(\boldsymbol{z}_B) \tag{7.111}$$

$$\hat{\boldsymbol{f}}(\boldsymbol{z}) = \boldsymbol{f} \circ \bar{\boldsymbol{v}}(\boldsymbol{z}) \,, \tag{7.112}$$

where $\bar{\boldsymbol{v}} : \text{CPE}_{\mathcal{B}}(\hat{\mathcal{Z}}^{\text{train}}) \to \text{CPE}_{\mathcal{B}}(\mathcal{Z}^{\text{train}})$ is defined as

$$\bar{\boldsymbol{v}}(\boldsymbol{z}) := \begin{bmatrix} \bar{\boldsymbol{v}}_{B_1}(\boldsymbol{z}_{\pi^{-1}(B_1)}) \\ \vdots \\ \bar{\boldsymbol{v}}_{B_\ell}(\boldsymbol{z}_{\pi^{-1}(B_\ell)}) \end{bmatrix} \,, \tag{7.113}$$

The map $\bar{\boldsymbol{v}}$ is a diffeomorphism since each $\bar{\boldsymbol{v}}_{\pi(B)}$ is a diffeomorphism from $\hat{\mathcal{Z}}_B^{\text{train}}$ to $\mathcal{Z}_{\pi(B)}^{\text{train}}$.

By (7.112) we get

$$\hat{\boldsymbol{f}}(\text{CPE}_{\mathcal{B}}(\hat{\mathcal{Z}}^{\text{train}})) = \boldsymbol{f} \circ \bar{\boldsymbol{v}}(\text{CPE}_{\mathcal{B}}(\hat{\mathcal{Z}}^{\text{train}})) \,, \tag{7.114}$$

and since the map $\bar{\boldsymbol{v}}$ is surjective we have $\bar{\boldsymbol{v}}(\text{CPE}_{\mathcal{B}}(\hat{\mathcal{Z}}^{\text{train}})) = \text{CPE}_{\mathcal{B}}(\mathcal{Z}^{\text{train}})$ and thus

$$\hat{\boldsymbol{f}}(\text{CPE}_{\mathcal{B}}(\hat{\mathcal{Z}}^{\text{train}})) = \boldsymbol{f}(\text{CPE}_{\mathcal{B}}(\mathcal{Z}^{\text{train}})) \,. \tag{7.115}$$

Hence if $\text{CPE}_{\mathcal{B}}(\mathcal{Z}^{\text{train}}) \subseteq \mathcal{Z}^{\text{test}}$, then $\boldsymbol{f}(\text{CPE}_{\mathcal{B}}(\mathcal{Z}^{\text{train}})) \subseteq \boldsymbol{f}(\mathcal{Z}^{\text{test}})$. ■

A.11. Will all extrapolated images make sense?

Here is a minimal example where the assumption $\text{CPE}_{\mathcal{B}}(\mathcal{Z}^{\text{train}}) \not\subseteq \mathcal{Z}^{\text{test}}$ is violated.

Example 7.10 (Violation of $\text{CPE}_{\mathcal{B}}(\mathcal{Z}^{\text{train}}) \not\subseteq \mathcal{Z}^{\text{test}}$)**.** *Imagine* $\boldsymbol{z} = (\boldsymbol{z}_1, \boldsymbol{z}_2)$ *where* $\boldsymbol{z}_1$ *and* $\boldsymbol{z}_2$ *are the* x*-positions of two distinct balls. It does not make sense to have two balls occupying the same location in space and thus whenever* $\boldsymbol{z}_1 = \boldsymbol{z}_2$ *we have* $(\boldsymbol{z}_1, \boldsymbol{z}_2) \notin \mathcal{Z}^{\text{test}}$*. But if* $(1, 2)$ *and* $(2, 1)$ *are both in* $\mathcal{Z}^{\text{train}}$*, it implies that* $(1, 1)$ *and* $(2, 2)$ *are in* $\text{CPE}(\mathcal{Z}^{\text{train}})$*, which is a violation of* $\text{CPE}_{\mathcal{B}}(\mathcal{Z}^{\text{train}}) \subseteq \mathcal{Z}^{\text{test}}$.

A.12. Additive decoders cannot model occlusion

We now explain why additive decoders cannot model occlusion. Occlusion occurs when an object is partially hidden behind another one. Intuitively, the issue is the following: Consider two images consisting of two objects, A and B (each image shows both objects). In both images, the position of object A is the same and in exactly one of the images, object B partially occludes object A. Since the position of object A did not change, its corresponding latent block $\boldsymbol{z}_A$ is also unchanged between both images. However, the pixels occupied by object A do change between both images because of occlusion. The issue is that, because of additivity, $\boldsymbol{z}_A$ and $\boldsymbol{z}_B$ cannot interact to make some pixels that belonged to object A "disappear" to be replaced by pixels of object B. In practice, object-centric representation learning methods rely a masking mechanism which allows interactions between $\boldsymbol{z}_A$ and $\boldsymbol{z}_B$ (See Equation 7.1 in Section 7.2). This highlights the importance of studying this class of decoders in future work.

B. Experiments

B.1. Training Details

Loss Function. We use the standard reconstruction objective of mean squared error loss between the ground truth data and the reconstructed/generated data.
Hyperparameters. For both the ScalarLatents and the BlockLatents dataset, we used the Adam optimizer with the hyperparameters defined below. Note that we maintain consistent hyperparameters across both the Additive decoder and the Non-Additive decoder method.

ScalarLatents Dataset.

- Batch Size: 64
- Learning Rate: 1×10^{-3}
- Weight Decay: 5×10^{-4}
- Total Epochs: 4000

BlockLatents Dataset.

- Batch Size: 1024
- Learning Rate: 1×10^{-3}
- Weight Decay: 5×10^{-4}
- Total Epochs: 6000

Model Architecture. We use the following architectures for Encoder and Decoder across both the datasets (ScalarLatents, BlockLatents). Note that for the ScalarLatents dataset we train with latent dimension $d_z = 2$, and for the BlockLatents dataset we train with latent dimension $d_z = 4$, which corresponds to the dimensionalities of the ground-truth data generating process for both datasets.

Encoder Architecture:

- RestNet-18 Architecture till the penultimate layer (512 dimensional feature output)
- Stack of 5 fully-connected layer blocks, with each block consisting of Linear Layer (dimensions: 512×512), Batch Normalization layer, and Leaky ReLU activation (negative slope: 0.01).
- Final Linear Layer (dimension: $512 \times d_z$) followed by Batch Normalization Layer to output the latent representation.

Decoder Architecture (Non-additive):

- Fully connected layer block with input as latent representation, consisting of Linear Layer (dimension: $d_z \times 512$), Batch Normalization layer, and Leaky ReLU activation (negative slope: 0.01).
- Stack of 5 fully-connected layer blocks, with each block consisting of Linear Layer (dimensions: 512×512), Batch Normalization layer, and Leaky ReLU activation (negative slope: 0.01).
- Series of DeConvolutional layers, where each DeConvolutional layer is follwed by Leaky ReLU (negative slope: 0.01) activation.
  - DeConvolution Layer (c_{in}: 64, c_{out}: 64, kernel: 4; stride: 2; padding: 1)
  - DeConvolution Layer (c_{in}: 64, c_{out}: 32, kernel: 4; stride: 2; padding: 1)
  - DeConvolution Layer (c_{in}: 32, c_{out}: 32, kernel: 4; stride: 2; padding: 1)
  - DeConvolution Layer (c_{in}: 32, c_{out}: 3, kernel: 4; stride: 2; padding: 1)

Decoder Architecture (Additive): Recall that an additive decoder has the form $\boldsymbol{f}(\boldsymbol{z}) = \sum_{B \in \mathcal{B}} \boldsymbol{f}^{(B)}(\boldsymbol{z}_B)$. Each $\boldsymbol{f}^{(B)}$ has the same architecture as the one presented above for the non-additive case, but the input has dimensionality $|B|$ (which is 1 or 2, depending on the dataset). Note that we do not share parameters among the functions $\boldsymbol{f}^{(B)}$.

B.2. Datasets Details

We use the moving balls environment from Ahuja et al. [2022b] with images of dimension $64 \times 64 \times 3$, with latent vector ($\boldsymbol{z}$) representing the position coordinates of each balls. We consider only two balls. The rendered images have pixels in the range [0, 255].

ScalarLatents Dataset. We fix the x-coordinate of each ball to 0.25 and 0.75. The only factors varying are the y-coordinates of both balls. Thus, $\boldsymbol{z} \in \mathbb{R}^2$ and $\mathcal{B} = \{\{1\}, \{2\}\}$ where $\boldsymbol{z}_1$ and $\boldsymbol{z}_2$ designate the y-coordinates of both balls. We sample the y-coordinate of the first ball from a continuous uniform distribution as follows: $\boldsymbol{z}_1 \sim$ Uniform(0, 1). Then we sample the y-coordinate of the second ball as per the following scheme:

$$z_2 \sim \begin{cases} \text{Uniform}(0,1) & \text{if } z_1 \leq 0.5 \\ \text{Uniform}(0,0.5) & \text{else} \end{cases}$$

Hence, this leads to the L-shaped latent support, i.e., $\mathcal{Z}^{\text{train}} := [0,1] \times [0,1] \setminus [0.5,1] \times [0.5,1]$.

We use $50k$ samples for the test dataset, while we use $20k$ samples for the train dataset along with $5k$ samples (25% of the train sample size) for the validation dataset.

BlockLatents Dataset. For this dataset, we allow the balls to move in both the x, y directions, so that $\boldsymbol{z} \in \mathbb{R}^4$ and $\mathcal{B} = \{\{1,2\},\{3,4\}\}$. For the case of **independent latents**, we sample each latent component independently and identically distributed according to a uniform distribution over $(0,1)$, i.e. $z_i \sim$ Uniform(0, 1). We rejected the images that present occlusion, i.e. when one ball hides another one.[2]

For the case of **dependent latents**, we sample the latents corresponding to the first ball similarly from the same continuous uniform distribution, i.e, $z_1, z_2 \sim$ Uniform (0, 1). However, the latents of the second ball are a function of the latents of the first ball, as described in what follows:

$$z_3 \sim \begin{cases} \text{Uniform}(0,0.5) & \text{if } 1.25 \times (z_1^2 + z_2^2) \geq 1.0 \\ \text{Uniform}(0.5,1) & \text{if } 1.25 \times (z_1^2 + z_2^2) < 1.0 \end{cases}$$

$$z_4 \sim \begin{cases} \text{Uniform}(0.5,1) & \text{if } 1.25 \times (z_1^2 + z_2^2) \geq 1.0 \\ \text{Uniform}(0,0.5) & \text{if } 1.25 \times (z_1^2 + z_2^2) < 1.0 \end{cases}$$

Intuitively, this means the second ball will be placed in either the top-left or the bottom-right quadrant based on the position of the first ball. We also exclude from the dataset the images presenting occlusion.

Note that our dependent BlockLatent setup is same as the non-linear SCM case from Ahuja et al. [Ahuja et al., 2023].

We use $50k$ samples for both the train and the test dataset, along with $12.5k$ samples (25% of the train sample size) for the validation dataset.

Disconnected Support Dataset. For this dataset, we have setup similar to the **ScalarLatents** dataset; we fix the x-coordinates of both balls to 0.25 and 0.75 and only vary the y-coordinates so that $\boldsymbol{z} \in \mathbb{R}^2$. We sample the y-coordinate of the first ball (z_1) from Uniform(0, 1). Then we sample the y-coordinate of the second ball (z_2) from either of the following continuous uniform distribution with equal probability; Uniform(0, 0.25) and Uniform(0.75, 1). This leads to a disconnected support given by $\mathcal{Z}^{\text{train}} := [0,1] \times [0,1] \setminus [0.25,0.75] \times [0.25,0.75]$.

[2]Note that, in the independent latents case, the latents are not actually independent because of the rejection step which prevents occlusion from happening.

We use $50k$ samples for the test dataset, while we use $20k$ samples for the train dataset along with $5k$ samples (25% of the train sample size) for the validation dataset.

B.3. Evaluation Metrics

Recall that, to evaluate disentanglement, we compute a matrix of scores $(s_{B,B'}) \in \mathbb{R}^{\ell \times \ell}$ where ℓ is the number of blocks in $\mathcal{B}$ and $s_{B,B'}$ is a score measuring how well we can predict the ground-truth block $\boldsymbol{z}_B$ from the learned latent block $\hat{\boldsymbol{z}}_{B'} = \hat{\boldsymbol{g}}_{B'}(\boldsymbol{x})$ outputted by the encoder. The final Latent Matching Score (LMS) is computed as $\text{LMS} = \arg\max_{\pi \in \mathfrak{S}_\mathcal{B}} \frac{1}{\ell} \sum_{B \in \mathcal{B}} s_{B,\pi(B)}$, where $\mathfrak{S}_\mathcal{B}$ is the set of permutations respecting $\mathcal{B}$ (Definition 7.2). These scores are always computed on the test set.
Metric $\text{LMS}_{\text{Spear}}$: As mentioned in the main paper, this metric is used for the **ScalarLatents** dataset where each block is 1-dimensional. Hence, this metric is almost the same as the mean correlation coefficient (MCC), which is widely used in the nonlinear ICA literature [Hyvarinen and Morioka, 2016, 2017, Hyvärinen et al., 2019, Khemakhem et al., 2020a, Lachapelle et al., 2022], with the only difference that we use Spearman correlation instead of Pearson correlation as a score $s_{B,B'}$. The Spearman correlation can capture nonlinear monotonous relations, unlike Pearson which can only capture linear dependencies. We favor Spearman over Pearson because our identifiability result (Theorem 7.2) guarantees we can recover the latents only up to permutation and element-wise invertible transformations, which can be nonlinear.
Metric LMS_{tree}: This metric is used for the **BlockLatents** dataset. For this metric, we take $s_{B,B'}$ to be the R^2 score of a Regression Tree with maximal depth of 10. For this, we used the class `sklearn.tree.DecisionTreeRegressor` from the `sklearn` library. We learn the parameters of the Decision Tree using the train dataset and then use it to evaluate LMS_{tree} metric on the test dataset. For the additive decoder, it is easy to compute this metric since the additive structure already gives a natural partition $\mathcal{B}$ which matches the ground-truth. However, for the non-additive decoder, there is no natural partition and thus we cannot compute LMS_{tree} directly. To go around this problem, for the non-additive decoder, we compute LMS_{tree} for all possible partitions of d_z latent variables into blocks of size $|B| = 2$ (assuming all blocks have the same dimension), and report the best LMS_{tree}. This procedure is tractable in our experiments due to the small dimensionality of the problem we consider.

B.4. Boxplots for main experiments (Table 7.1)

Since the standard error in the main results (Table 7.1) was high, we provide boxplots in Figures 7.9 & 7.10 to have a better visibility on what is causing this. We observe that the high standard error for the Additive approach was due to bad performance for a few bad random initializations for the ScalarLatents dataset; while we have nearly perfect latent identification for the others. Figure 7.14e shows the latent space learned by the worst case seed, which somehow learned a disconnected

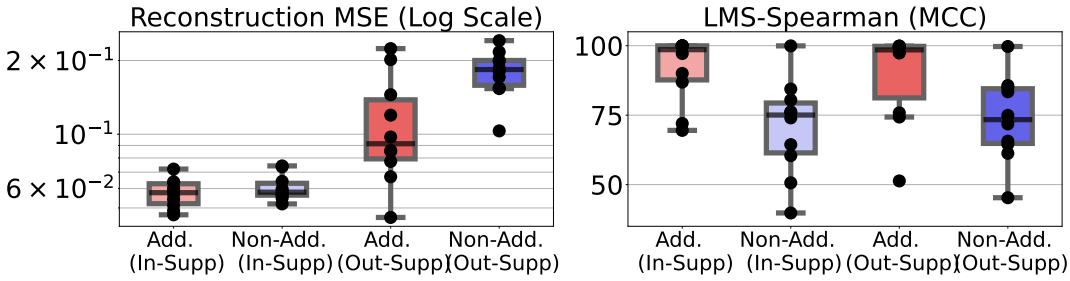
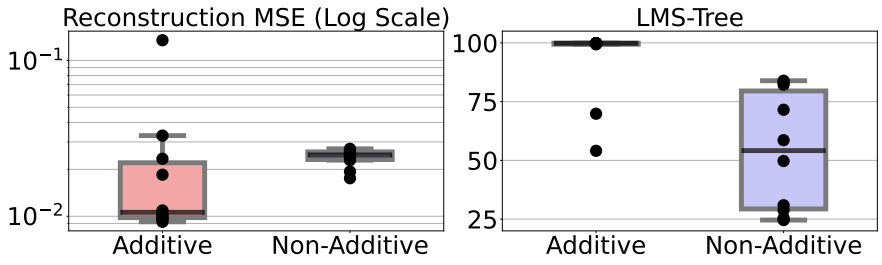
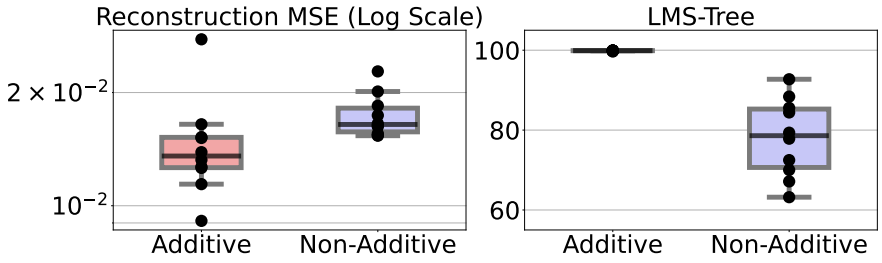


Figure 7.9. Reconstruction mean squared error (MSE) (↓) and Latent Matching Score (LMS) (↑) over 10 different random initializations for **ScalarLatents** dataset.



(a) Independent Latent Case



(b) Dependent Latent Case

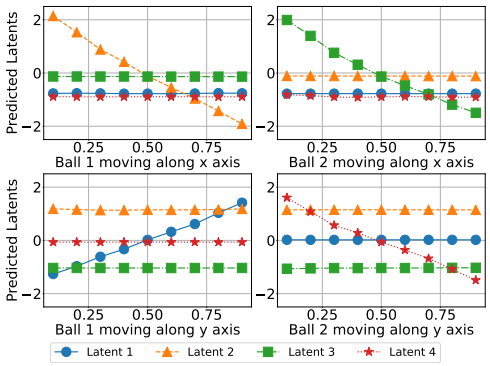
Figure 7.10. Reconstruction mean squared error (MSE) (↓) and Latent Matching Score (LMS) (↑) for 10 different initializations for **BlockLatents** dataset.

support even if the ground-truth support was connected. Similarly, for the case of Independent BlockLatents, there are only a couple of bad random initializations and the rest of the cases have perfect identification.

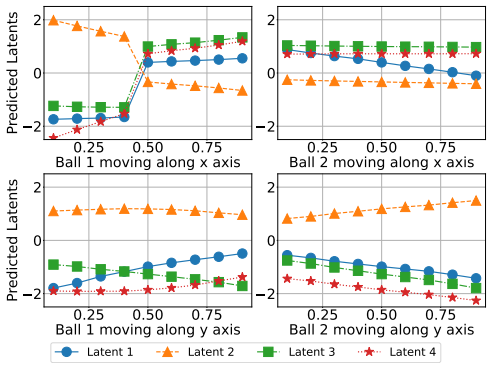
B.5. Additional Results: BlockLatents Dataset

To get a qualitative understanding of latent identification in the BlockLatents dataset, we plot the response of each predicted latent as we change a particular ground-truth latent factor. We describe the following cases of changing the ground-truth latents:

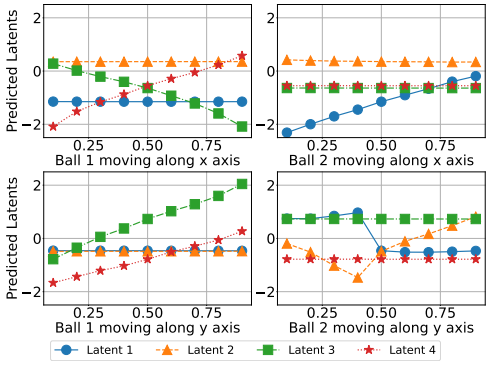
- **Ball 1 moving along x-axis:** We sample 10 equally spaced points for $\boldsymbol{z}_1$ from $[0, 1]$; while keeping other latents fixed as follows: $\boldsymbol{z}_2 = 0.25, \boldsymbol{z}_3 = 0.50, \boldsymbol{z}_4 = 0.75$. We will never have occlusion since the balls are separated along the y-axis $\boldsymbol{z}_4 - \boldsymbol{z}_2 > 0$.



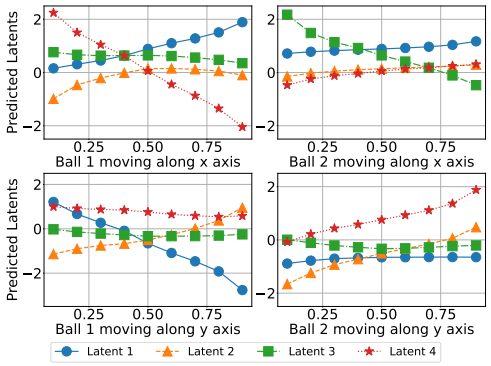
(a) Additive Decoder (Best) (LMS_{Tree} : 99.9)



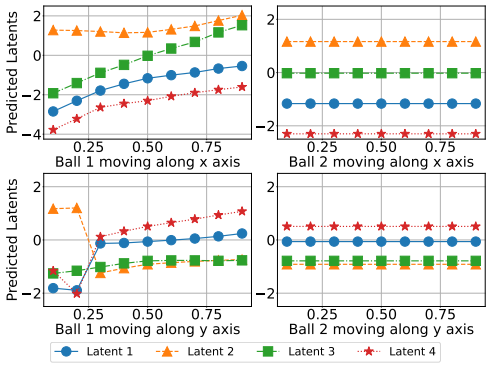
(b) Non-Additive Decoder (Best) (LMS_{Tree} : 83.9)



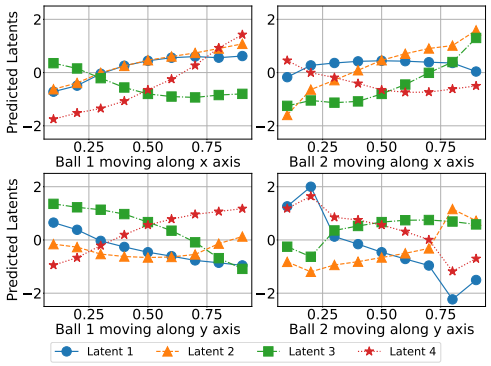
(c) Additive Decoder (Median) (LMS_{Tree} : 99.8)



(d) Non-Additive Decoder (Median) (LMS_{Tree} : 58.6)



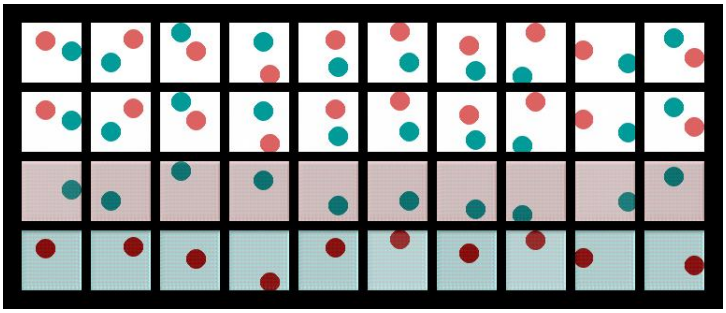
(e) Additive Decoder (Worst) (LMS_{Tree} : 54.1)



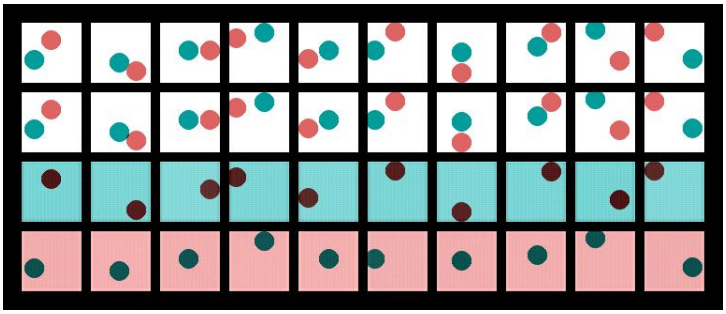
(f) Non-Additive Decoder (Worst) (LMS_{Tree} : 24.6)

Figure 7.11. Latent responses for the cases with the **best/median/worst** LMS_{Tree} among runs performed on the **BlockLatent** dataset with independent latents. In each plot, we report the latent factors predicted from multiple images where one ball moves along only one axis at a time.

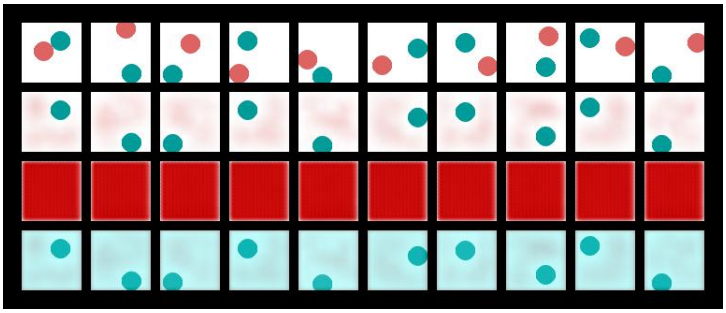
- **Ball 2 moving along x-axis:** We sample 10 equally spaced points for z_3 from $[0, 1]$; while keeping other latents fixed as follows: $z_1 = 0.50, z_2 = 0.25, z_4 = 0.75$. We will never have occlusion since the balls are separated along the y-axis $z_4 - z_2 > 0$.
- **Ball 1 moving along y-axis:** We sample 10 equally spaced points for z_2 from $[0, 1]$; while keeping other latents fixed as follows: $z_1 = 0.25, z_3 = 0.75, z_4 = 0.50$. We will never have occlusion since the balls are separated along the x-axis $z_3 - z_1 > 0$.



(a) Additive Decoder (Best)



(b) Additive Decoder (Median)



(c) Additive Decoder (Worst)

Figure 7.12. Object-specific renderings with the **best/median/worst** LMS_{tree} among runs performed on the **BlockLatents** dataset with independent latents. In each plot, the first row is the original image, the second row is the reconstruction and the third and fourth rows are the output of the object-specific decoders. In the best and median cases, each object-specific decoder corresponds to one and only one object, e.g. the third row of the best case always corresponds to the red ball. However, in the worst case, there are issues with reconstruction as only one of the balls is generated. Note that the visual artefacts are due to the additive constant indeterminacy we saw in Theorem 7.2, which cancel each other as is suggested by the absence of artefacts in the reconstruction.

- **Ball 2 moving along y-axis:** We sample 10 equally spaced points for $\boldsymbol{z}_4$ from $[0, 1]$; while keeping other latents fixed as follows: $\boldsymbol{z}_1 = 0.25, \boldsymbol{z}_2 = 0.50, \boldsymbol{z}_3 = 0.75$. We will never have occlusion since the balls are separated along the x-axis $\boldsymbol{z}_3 - \boldsymbol{z}_1 > 0$.

Figure 7.5 in the main paper presents the latent responses plot for the median LMS_{tree} case among random initializations. In Figure 7.11, we provide the results for the case of best and the

worst LMS_{tree} among random seeds. We find that Additive Decoder fails for only for the worst case random seed, while Non-Additive Decoder fails for all the cases.

Additionally, we provide the object-specific reconstructions for the Additive Decoder in Figure 7.12. This helps us better understand the failure of Additive Decoder for the worst case random seed (Figure 7.12c), where the issue arises due to bad reconstruction error.

B.6. Disconnected Support Experiments

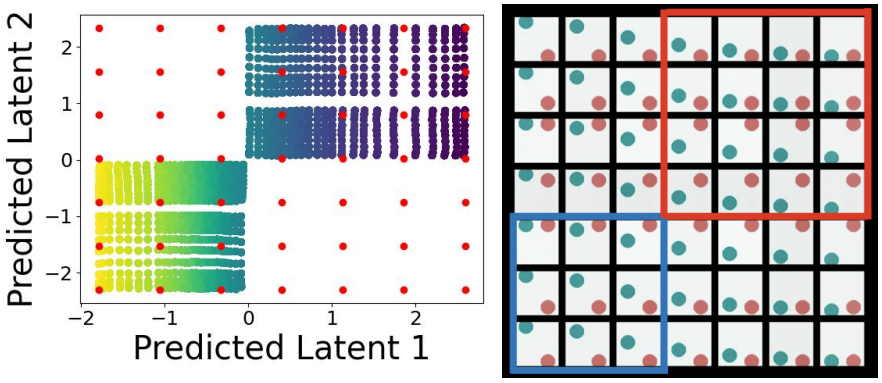


Figure 7.13. Learned latent space, $\hat{\mathcal{Z}}^{\text{train}}$, and the corresponding reconstructed images of the additive decoder with the **median** $\text{LMS}_{\text{Spear}}$ among runs performed on the **Disconnected Support** dataset. The red dots correspond to latent factors used to generate the images.

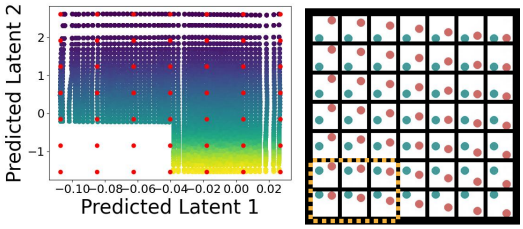
Since path-connected latent support is an important assumption for latent identification with additive decoders (Theorem 7.2), we provide results for the case where the assumption is not satisfied. We experiment with the **Disconnected Support** dataset (Section B.2) and find that we obtain much worse $\text{LMS}_{\text{Spear}}$ as compared to the case of training with L-shaped support in the **ScalarLatents** dataset. Over 10 different random initializations, we find mean $\text{LMS}_{\text{Spear}}$ performance of 69.5 with standard error of 6.69.

For better qualitative understanding, we provide visualization of the latent support and the extrapolated images for the median $\text{LMS}_{\text{Spear}}$ among 10 random seeds in Figure 7.13. Somewhat surprisingly, the representation appears to be aligned in the sense that the first predicted latent corresponds to the blue ball while the second predicted latent correspond to the red ball. Also surprisingly, extrapolation occurs (we can see images of both balls high). That being said, we observe that the relationship between the predicted latent 2 ($\hat{z}_2$) and y-coordinate of second (red) ball is not monotonic, which explains why the Spearman correlation is so low (Spearman correlation scores are high when there is a monotonic relationship between both variables).

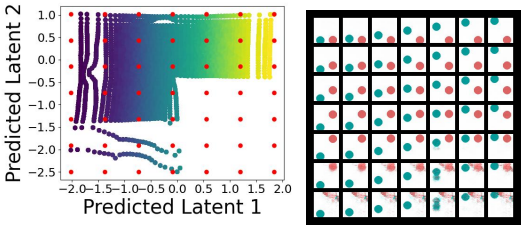
B.7. Additional Results: ScalarLatents Dataset

To get a qualitative understanding of extrapolation, we plot the latent support on the test dataset and sample a grid of equally spaced points from the support of each predicted latent on the test dataset. The grid represents the cartesian-product of the support of predicted latents and would contain novel combinations of latents that were unseen during training. We show the reconstructed images for each point from the cartesian-product grid to see whether the model is able to reconstruct well the novel latent combinations.

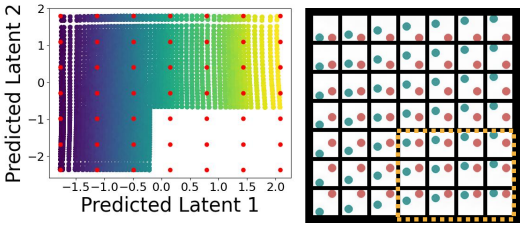
Figure 7.4 in the main paper presents visualizations of the latent support and the extrapolated images for the median $\text{LMS}_{\text{Spear}}$ case among random seeds. In Figure 7.14, we provide the results for the case of best and the worst $\text{LMS}_{\text{Spear}}$ among random seeds. We find that even for the best case (Figure 7.14b), Non-Additive Decoder does not generate good quality extrapolated images, while Additive Decoder generates extrapoalted images for the best and median case. The worst-case run for the Additive Decoder has disconnected support, which explains why it is not able to extrapolate.



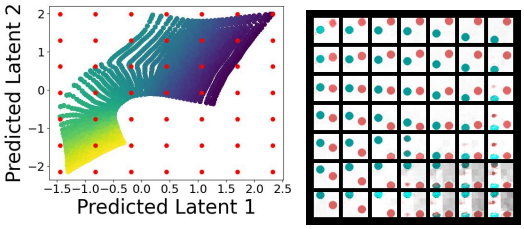
(a) Additive Decoder (Best) ($\text{LMS}_{\text{Spear}}$: 99.9)



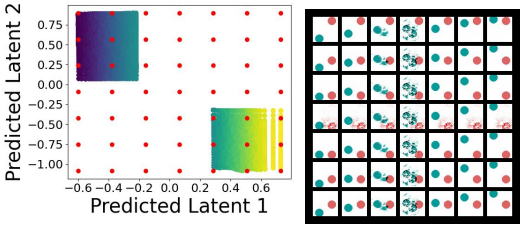
(b) Non-Additive Decoder (Best) ($\text{LMS}_{\text{Spear}}$: 99.9)



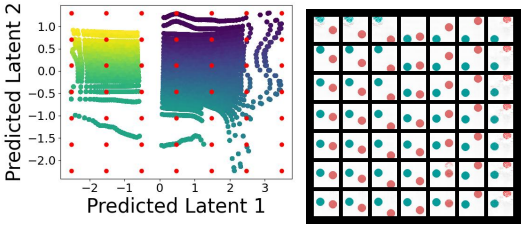
(c) Additive Decoder (Median) ($\text{LMS}_{\text{Spear}}$: 99.9)



(d) Non-Additive Decoder (Median) ($\text{LMS}_{\text{Spear}}$: 76.1)



(e) Additive Decoder (Worst) ($\text{LMS}_{\text{Spear}}$: 69.5)



(f) Non-Additive Decoder (Worst) ($\text{LMS}_{\text{Spear}}$: 39.8)

Figure 7.14. Figure (a, c, e) shows the learned latent space, $\hat{\mathcal{Z}}^{\text{train}}$, and the corresponding reconstructed images of the additive decoder with the **best/median/worst** $\text{LMS}_{\text{Spear}}$ among runs performed on the **ScalarLatents** dataset. Figure (b, d, f) shows the same thing for the non-additive decoder. The red dots correspond to latent factors used to generate the images and the yellow square highlights extrapolated images.

Chapter 8

An End in Itself? Interpretations of Identifiability and Motivations for Generalization Guarantees

> *Suppose you have two theories, A and B, which look completely different psychologically, with different ideas in them and so on, but that all the consequences that are computed from each are exactly the same, and both agree with experiment. [...] How are we going to decide which one is right? There is no way by science, because they both agree with experiment to the same extent. [...] However, for psychological reasons, in order to guess new theories, these two may be very far from equivalent, because one gives a man different ideas from the other. [...] There are certain ways of changing one which looks natural which will not look natural in the other.*
>
> — Feynman [1965, p. 168]

This thesis has focused on the question of identifiability both in causal discovery (Chapters 3 & 4) and representation learning (Chapters 5, 6 & 7). In this chapter, I motivate the study of identifiability further both as *an end in itself* (Section 8.1) and as *a means to an end* (Section 8.2).

In Section 8.1, I list three possible ways identifiability guarantees can be interpreted, namely the *realist interpretation*, the *independent-learner interpretation*, and the *interpretability interpretation*, of which the first two are largely based on the work of Gresele [2023]. All three interpretations correspond to different reasons one might care about identifiability *as an end in itself*. Indeed, the first interpretation views identifiability as a way to uncover causal or physical structure about how the world works. The second one sees identifiability as a desirable property of a statistical model that informs us about how multiple reruns of the same algorithm yield more or less similar models/representations. The third one sees identifiability as a way to obtain models that can easily be interpreted by a machine learning practitioner.

In Section 8.2, I provide multiple mathematically concrete settings, mainly based on the contributions of this thesis, where an identifiability result is instrumental to prove a downstream

performance guarantee. We will see that identifiability appears as an intermediate step when going from assumptions about the data to generalization guarantees (Figure 8.1). In other words, identifiability is seen as *a means to an end*, as opposed to something desirable in and of itself. I consider four seemingly unrelated problem settings, namely causal discovery, additive decoders for extrapolation, sparse multitask learning and semi-supervised learning via clustering; and unify them under the umbrella of statistical decision theory (Section 2.2). Moreover, I identify three general steps that appear in all four settings provided: (i) Choose assumptions suitable for the problem at hand postulating that some unknown structure is present in the data, (ii) show that this structure can be recovered from data via an identifiability analysis, and (iii) leverage the learned structure to provide generalization guarantees on the downstream task. These three steps as well as how they apply to the four settings considered here are summarized in Table 8.1.

8.1. Three interpretations of identifiability

Probability theory is a mathematical framework aimed at describing the uncertainty we face in the real world. It can be used to describe different types of uncertainty, i.e. it can be interpreted in various ways. For instance, the *frequentist interpretation* of probability states that uncertain experiments can be repeated multiple times and that a probability of some event occurring is the proportion of times the event would occur if one would repeat the experiment infinitely many times. A standard example of such an experiment is a coin flip which, intuitively, can be repeated as many times as one has patience for. In contrast, the *Bayesian interpretation* sees a probability as a precise description of a belief. For instance, someone might believe that there is 90% chance that tomorrow will be rainy in Montreal. In this case, the probability of rain captures a subjective belief and not the result of an infinitely repeatable experiment (one could argue that tomorrow occurs only once and is thus not repeatable). The same mathematical framework is used in two different ways to describe the world.

In this section, I argue that identifiability guarantees, which are mathematical statements, can be interpreted in different ways, analogously to how probability theory has different interpretations. I describe three interpretations of identifiability: the *realist interpretation*, the *independent learners interpretation* and the *interpretability interpretation*. I emphasize that the first two correspond respectively to the cocktail-party and the independent-listeners metaphors discussed by Gresele [2023].

To make the discussion more concrete, I focus on identifiability in representation learning, especially up to permutation and element-wise transformations, although I believe this discussion applies more broadly for instance in causal discovery or clustering. Recall the formal definition of

identifiability in this context, which was introduced in Section 2.5.1:

$$\forall(\boldsymbol{f}, \mathbb{P}_{\boldsymbol{z}}) \in \mathcal{F} \times \mathcal{P}, (\hat{\boldsymbol{f}}, \hat{\mathbb{P}}_{\boldsymbol{z}}) \in \hat{\mathcal{F}} \times \hat{\mathcal{P}}, \mathbb{P}_{(\boldsymbol{f}, \mathbb{P}_{\boldsymbol{z}})} = \mathbb{P}_{(\hat{\boldsymbol{f}}, \hat{\mathbb{P}}_{\boldsymbol{z}})} \implies \boldsymbol{f} = \hat{\boldsymbol{f}} \circ \boldsymbol{d} \circ \boldsymbol{P}\,, \tag{8.1}$$

where $\mathcal{F}$ and $\hat{\mathcal{F}}$ and function classes for the decoder $\boldsymbol{f}$ and $\mathcal{P}$ and $\hat{\mathcal{P}}$ are hypothesis classes for the distribution over the latent vector $\boldsymbol{z}$. As discussed in Section 2.5.1, identifiability guarantees are typically asymmetric in the sense that $\mathcal{F} \times \mathcal{P}$ is a proper subset of $\hat{\mathcal{F}} \times \hat{\mathcal{P}}$.

8.1.1. The realist interpretation

In the realist interpretation, the model $(\boldsymbol{f}, \mathbb{P}_z) \in \mathcal{F} \times \mathcal{P}$ is thought of as a ground-truth model that represents faithfully a physical or causal process giving rise to the distribution over observations $\mathbb{P}_{(\boldsymbol{f}, \mathbb{P}_{\boldsymbol{z}})}$, while $(\hat{\boldsymbol{f}}, \hat{\mathbb{P}}_{\boldsymbol{z}}) \in \hat{\mathcal{F}} \times \hat{\mathcal{P}}$ is thought of as a model fitted to the observations such that $\mathbb{P}_{(\hat{\boldsymbol{f}}, \hat{\mathbb{P}}_{\boldsymbol{z}})} = \mathbb{P}_{(\boldsymbol{f}, \mathbb{P}_{\boldsymbol{z}})}$. The identifiability guarantee then states that the learned representation $\hat{\boldsymbol{f}}$ is the same as the ground-truth representation $\boldsymbol{f}$ up to some indeterminacy. In this interpretation, the learned model has discovered something "physical" about how the data came about, which of course might be of scientific interest.

Gresele [2023] discussed this interpretation and brought up the well-known cocktail party problem as an example where it applies. In the cocktail party problem, the learner must separate audio signals produced by multiple people based on audio signals captured by multiple microphones positioned in the room where the party is taking place. The signals produced by the attendees are then modelled by the latent vector $\boldsymbol{z}$ (each dimension corresponds to an attendee) and the recorded signal is given by $\boldsymbol{x}$ (each dimension corresponds to a microphone).[1] The underlying assumption is that, the signal recorded by one microphone is a linear combination of the signals emitted by the attendees, where the coefficients are functions of the distance between an attendee and a microphone. We can thus say that $\boldsymbol{x} = \boldsymbol{A}\boldsymbol{z}$ for some matrix $\boldsymbol{A}$. Assuming the signal emitted by the attendees are independent and non-Gaussian, one can apply the standard linear ICA result to guarantee identifiability of the latent factors up to permutation and rescaling.

In this previous example, the ground-truth model $\boldsymbol{x} = \boldsymbol{A}\boldsymbol{z}$ is an actual physical description of what the microphones record as a function of the signals $\boldsymbol{z}$. In that sense, there is truly a ground-truth model that one aims to recover. The assumptions on this ground-truth model is given by $\mathcal{F}$ and $\mathcal{P}$ and one can think of these as assumptions about how the world *is*. The assumption that $\boldsymbol{A}$ is invertible is in fact an assumption about the spatial configuration of the microphones relative to the attendees. The assumption that the latent factors are non-Gaussian is an assumption about what the attendees are actually saying.

[1] Audio signals usually unroll in time so that $\boldsymbol{x}_t = \boldsymbol{A}\boldsymbol{z}_t$ where t is a time index. This slight complication is avoided to simplify the discussion.

8.1.2. The independent-learners interpretation

Gresele [2023] suggested a second interpretation based on a variant of the cocktail-party problem in which the same recorded audio signals are processed by different learners, or "listeners". In this interpretation, $(\boldsymbol{f}, \mathbb{P}_{\boldsymbol{z}})$ and $(\hat{\boldsymbol{f}}, \hat{\mathbb{P}}_{\boldsymbol{z}})$ have similar status, they both correspond to models outputted by two different learners fitted on the same ground-truth distribution $\mathbb{P}_{\boldsymbol{x}}$. More precisely, we can imagine that both algorithms consist in searching in the space $\hat{\mathcal{F}} \times \hat{\mathcal{P}}$ for a model that fits the ground-truth distribution exactly. If $\hat{\mathcal{F}} \times \hat{\mathcal{P}}$ is expressive enough to represent the ground-truth distribution $\mathbb{P}_{\boldsymbol{x}}$, we must have that $\mathbb{P}_{(\hat{\boldsymbol{f}}, \hat{\mathbb{P}}_{\boldsymbol{z}})} = \mathbb{P}_{\boldsymbol{x}} = \mathbb{P}_{(\boldsymbol{f}, \mathbb{P}_{\boldsymbol{z}})}$. Note that even if both models represent the data distribution exactly, this does not mean that the generative process assumed by the models represent anything physical or causal about how the data came about, in contrast with the realist interpretation of Section 8.1.1. Now, to apply the identifiability guarantee, we must assume that one of the models is in $\mathcal{F} \times \mathcal{P}$, which is typically a proper subset of $\hat{\mathcal{F}} \times \hat{\mathcal{P}}$. Again, this is not an assumption about how the data is generated, it is an assumption about one of the model outputted by both instantiations of the same algorithm. In that case, the identifiability result informs us that both models are "the same" up to some equivalence relation. In some sense, this interpretation is more down-to-earth and operational in the sense that it does not claim to uncover something "real" about how the data is generated, but only states that, if two instantiations of the same algorithm end up with the same observation model, then we can relate their representations via a simple transformation, like permutation-scaling for instance. I claim that such a property is necessary for a model to be interpretable since if it does not hold, different reruns of the same algorithm (say with different initialization) might end up with drastically different representations and thus different interpretations.[2]

8.1.3. The interpretability interpretation

I propose a third interpretation which sees identifiability as a guarantee that a model will be *interpretable*. Let us start by postulating that a machine learning practitioner considers some representation to be *natural* and let $\boldsymbol{f}$ be the mapping from what they consider to be the natural factors of variations, $\boldsymbol{z}$, to the observations, $\boldsymbol{x}$. This mapping can be thought of as existing in the mind of the practitioner, without necessarily representing a causal or physical process that is truly present in the real world. Then, a candidate decoder $\hat{\boldsymbol{f}}$ (for instance learned with data) is said to be interpretable if it can be related to the decoder of the practitioner by a permutation composed with an element-wise invertible transformation, i.e. $\boldsymbol{f} = \hat{\boldsymbol{f}} \circ \boldsymbol{d} \circ \boldsymbol{P}$. In other words, the representation of a model is interpretable if its coordinates can be permuted and transformed (bijectively) to match the factors of variations the practitioner considers natural. This captures the idea that someone

[2] I am sweeping under the rug important considerations such as whether the algorithm reaches the global optimum and the fact that we train only on finite datasets.

considers a representation interpretable when it relates to their own internal representation via a simple transformation. We can then go further and assume that $\boldsymbol{f}$ is part of the world model of the practitioner given by $\mathbb{P}_{(\boldsymbol{f},\mathbb{P}_{\boldsymbol{z}})}$. We further assume that this internal model exactly matches the ground-truth distribution over observations, i.e. $\mathbb{P}_{(\boldsymbol{f},\mathbb{P}_{\boldsymbol{z}})} = \mathbb{P}_{\boldsymbol{x}}$. Similarly to the second interpretation, there is no need to assume that the model $\boldsymbol{x} = \boldsymbol{f}(\boldsymbol{z})$ is "causal" or represents anything physical about the world. By assuming that the model also fits the data exactly, i.e. $\mathbb{P}_{(\hat{\boldsymbol{f}},\hat{\mathbb{P}}_{\boldsymbol{z}})} = \mathbb{P}_{\boldsymbol{x}}$, identifiability guarantees that the resulting representation $\hat{\boldsymbol{f}}$ will be interpretable in the sense that $\boldsymbol{f} = \hat{\boldsymbol{f}} \circ \boldsymbol{d} \circ \boldsymbol{P}$. In this view, assuming that $(\boldsymbol{f}, \mathbb{P}_{\boldsymbol{z}}) \in \mathcal{F} \times \mathcal{P}$ is actually an assumption about the internal world model of the practitioner. More precisely, $\boldsymbol{f} \in \mathcal{F}$ states that the relationship between what the practitioner considers to be natural factors of variations and the observations satisfies some specific constraint, e.g. diffeomorphism, additive or sparse. Additionally, $\mathbb{P}_{\boldsymbol{z}} \in \mathcal{P}$ enforces constraints on how the natural factors of variation are distributed in the internal world model of the practitioner, e.g. independent, non-Gaussian or sparsely connected in time.

Note that, even if we avoided the need to assume the existence of a specific ground-truth data-generating process, the assumption that $\mathbb{P}_{(\boldsymbol{f},\mathbb{P}_{\boldsymbol{z}})} = \mathbb{P}_{\boldsymbol{x}}$ is strong since it postulates that the ground-truth distribution $\mathbb{P}_{\boldsymbol{x}}$ *can* be expressed with a model of the form $\boldsymbol{x} = \boldsymbol{f}(\boldsymbol{z})$. But I stress the fact that this is different from saying that $\boldsymbol{x} = \boldsymbol{f}(\boldsymbol{z})$ is a causal model describing how the distribution $\mathbb{P}_{\boldsymbol{x}}$ came about, as was the case in the realist interpretation. I hypothesize that an approximate fit such as $\mathbb{P}_{(\boldsymbol{f},\mathbb{P}_{\boldsymbol{z}})} \approx \mathbb{P}_{\boldsymbol{x}} \approx \mathbb{P}_{(\hat{\boldsymbol{f}},\hat{\mathbb{P}}_{\boldsymbol{z}})}$ might be enough to guarantee "approximate interpretability" of the form $\boldsymbol{f} \approx \hat{\boldsymbol{f}} \circ \boldsymbol{d} \circ \boldsymbol{P}$. However, proving such approximate identifiability guarantees remains an important open question.

One could argue that the interpretability interpretation is in fact a special case of the independent-learners interpretation. After all, the machine learning practitioner can be viewed as a second "independent learner" applying their own algorithm to model the world.[3] This is a valid point, but I still consider the interpretability interpretation to be useful on its own as it provides a more human-centric perspective to identifiability which may guide us differently as we explore the space of possible assumptions for such generative models. It leads to questions such as "What are the properties of the representations we humans deem natural and/or interpretable?" for which we can propose answers that can be encoded into both $\mathcal{F}$ and $\mathcal{P}$.

8.2. From identifiability to generalization guarantees

As pointed out in the very first citation of this thesis from David Hume, inductive reasoning is impossible without assuming some kind of uniformity in nature, i.e. without making assumptions about how the world works. Once we postulate that some set of assumption holds, we can sometimes

[3]It was Luigi Gresele who raised this concern in a discussion we had at the Third Bellairs Workshop on Causality in Barbados (2024).

use deductive reasoning to prove that a given algorithm is guaranteed to perform well in a given setting. The goal of this section is to argue that identifiability guarantees, which are the focus of this thesis, can be a useful intermediate step along the chain of deductions linking assumptions to generalization guarantees (Figure 8.1). To support this claim, we identify this pattern across four seemingly different problem settings, namely causal discovery (Section 8.2.1), additive decoders for extrapolation (Section 8.2.2), sparse multitask learning (Section 8.2.3) and semi-supervised learning via clustering (Section 8.2.4). I spell out three simple "steps" that are common to all four examples and clarify where and how identifiability can be used to prove generalization guarantees. I now describe these three steps abstractly and delay how they apply to each problem settings to later in the section (also see Table 8.1 for a summary).

- **Step 1: Choose assumptions.** The first step is to choose some set of assumptions reasonable for the task at hand. These assumptions postulate that some unknown structure gives rise to both the observed data and the downstream task we wish to solve.
- **Step 2: Prove identifiability.** Using the assumptions from **Step 1**, we prove that the structure can be recovered from the data available, up to some equivalence class. This is the identifiability guarantee.
- **Step 3: Prove generalization.** Leverage the assumptions of **Step 1** and the knowledge of the identified structure from **Step 2** to conclude that the algorithm will perform well on some downstream task.

The meaning of these different steps will be become more transparent once we see how they apply to more concrete examples, as summarized in Table 8.1. Before going further, notice that the standard supervised machine learning setting with i.i.d. samples fits in this framework. Indeed, **Step 1** consists in assuming that all samples $(\boldsymbol{x}^{(i)}, y^{(i)})$ from the training set are independently and identically distributed and, crucially, that this is also true of future samples encountered at testing time. **Step 2** here is fairly trivial, since it is clear that one can identify $p(y \mid \boldsymbol{x})$ from $p(\boldsymbol{x}, y)$, which is enough to get perfect prediction. In **Step 3**, we leverage the knowledge of $p(y \mid \boldsymbol{x})$ and the assumption that the distribution does not change between training and test time to make good predictions at test time, which here is the downstream task. Although a lot can be said about this setting in the finite-data regime (e.g. about the bias-variance trade-off), the infinite-data regime is rather trivial, since the only structure that one leverages is the fact that the data is i.i.d. at both training and testing time, which makes the identifiability question obvious (we can clearly identify $p(y \mid \boldsymbol{x})$ from $p(\boldsymbol{x}, y)$). In this section, I concentrate on settings with more interesting structure that can be identified and leveraged for downstream performance, where "downstream" typically means "a task that somewhat differs from the one(s) observed at training time".

I will analyze all four settings using the framework of statistical decision theory encountered in Section 2.2, which we briefly review next.

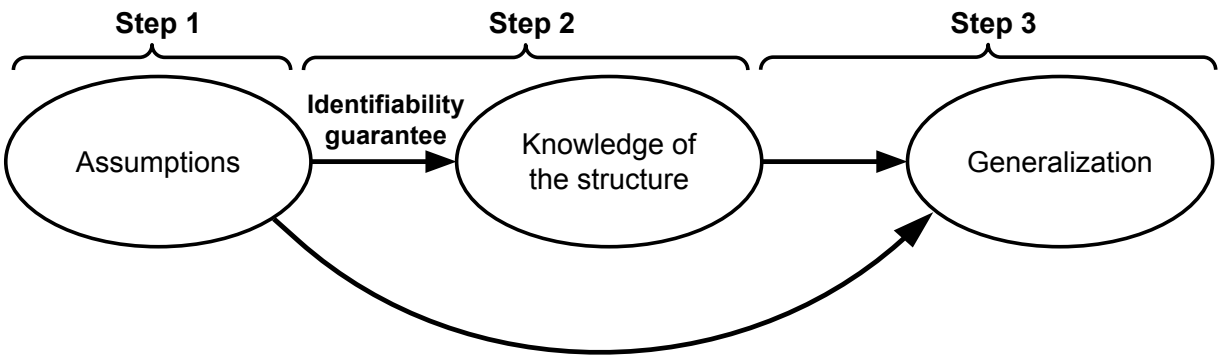


Figure 8.1. Graphical representation of the deduction steps to obtain generalization guarantees from identifiability guarantees.

Statistical decision theory. The *state of the world* is captured by $\boldsymbol{\theta}$ which is assumed to live in some space Θ. A decision maker must choose an action $\boldsymbol{a}$ in a space of actions A. A loss function $\ell(\boldsymbol{\theta}, \boldsymbol{a})$ dictates which cost is incurred based on both the state of the world $\boldsymbol{\theta}$ and the action $\boldsymbol{a}$. The decision maker can base its decision on an observable O which is related to $\boldsymbol{\theta}$ in some way. The standard framework assumes that, for each state of the world $\boldsymbol{\theta} \in \Theta$, there is a distribution $\mathbb{D}_{\boldsymbol{\theta}}$ capturing the uncertainty of O, i.e. $O \sim \mathbb{D}_{\boldsymbol{\theta}}$. In this framework, O is typically a dataset of samples. Since we concentrate on identifiability, which focuses on the infinite-data regime, we will allow O to be a full distribution over observations. In some cases, O will also contain finite datasets, namely in Sections 8.2.3 & 8.2.4. Based on the observable O, a decision rule δ must output an action $\boldsymbol{a} \in A$. Putting everything together, we can evaluate a decision rule δ by analyzing $\ell(\boldsymbol{\theta}, \delta(O))$, which (i) is potentially random since O is potentially random, and (ii) depends on the state of the world $\boldsymbol{\theta}$, which is unknown. In Section 2.2, we provide a few classical statistical problems that can be framed within statistical theory and briefly discuss different ways one can aggregate $\ell(\boldsymbol{\theta}, \delta(O))$ over both the uncertainty of O and $\boldsymbol{\theta}$ in order to compare rules.

The framework of statistical decision theory will allow us to unify all settings and formalize our goal more clearly. Assumptions about the unknown structure from **Step 1** can be encoded in the state of the world $\boldsymbol{\theta}$ and how it gives rise to both the observable O and the final loss function $\ell(\boldsymbol{\theta}, \boldsymbol{a})$. The loss function $\ell(\boldsymbol{\theta}, \boldsymbol{a})$ provides a performance metric for the downstream task we care about. In each setting, we will study different decision rules δ and show how identifiability guarantees can be used to obtain performance guarantees, as measured by $\ell(\boldsymbol{\theta}, \boldsymbol{a})$.

8.2.1. Causal discovery

In Chapters 3 & 4, I proposed novel algorithms to learn a causal graph with or without interventional data. In both cases, the approach is always based on some identifiability guarantee providing sufficient conditions so that the causal graph can be recovered either fully, or up to some equivalence

Problem setting	**Step 1:** Choose assumptions	**Step 2:** Prove identifiability	**Step 3:** Prove generalization
Causal discovery (Chapters 3 & 4)	In the correct causal factorization, interventions leave many conditional distributions invariant	Causal graph is identifiable from data (potentially interventional)	Leverage causal graph to predict the effect of unseen interventions
Additive decoders for extrapolation (Chapter 7)	Images consisting of multiple objects have an (almost) additive structure	Additive structure is identifiable from data	Leverage additive structure for Cartesian-product extrapolation
Sparse multitask learning (Chapter 6)	Multiple prediction tasks can be solved using a shared representation and each task requires only a few features (sparse task)	This representation is identifiable up to permutation and rescaling	Leverage the disentangled representation to solve a novel sparse task with fewer samples
Semi-supervised learning w/ clustering	The cluster assumption: data points that belong to the same cluster are likely to have the same label	The clusters are idenfiable from a large dataset of unlabeled samples	Leverage knowledge of the clusters to learn with fewer labelled samples

Table 8.1. Summary of the 3-steps procedure to deduce generalization guarantees from assumptions via identifiability results. The table shows how they apply to the four problem settings covered in this chapter. These steps are detailed in Section 8.2.

class. In both of these works, the main method of evaluation is to measure some notion of distance between the ground-truth and estimated graphs. In this section, instead of focusing on identifying the causal graph as an end in itself, the focus is placed on what can be done with the estimated graph, namely, predicting the effect of interventions that were never observed before. As for the next problem settings considered in this chapter, I use the language of statistical decision theory to formulate our goal.

Problem setting and "state of the world" $\boldsymbol{\theta}$. We assume that the vector of observations $\boldsymbol{x} \in \mathbb{R}^{d_x}$ is modelled by a causal graphical model (CGM) with causal graph $\mathcal{G}$ and causal mechanisms $f_j(\boldsymbol{x}_j \mid \boldsymbol{x}_{\pi_j^{\mathcal{G}}})$ for all $j \in \{1, ..., d_x\}$, where we are using the notation introduced in Section 2.3. This induces an observational density function $p(\boldsymbol{x}) := \prod_{j=1}^{d} f_j(\boldsymbol{x}_j \mid \boldsymbol{x}_{\pi_j^{\mathcal{G}}})$, which we assume puts probability mass everywhere on $\mathbb{R}^{d_x}$. Note that we implicitly assume causal sufficiency, i.e. there is no hidden variable causing more than one observed variable. Here, the "state of the world" is given by the CGM itself, i.e.

$$\boldsymbol{\theta} := (\mathcal{G}, \{\boldsymbol{f}_j(\boldsymbol{x}_j \mid \boldsymbol{x}_{\pi_j^{\mathcal{G}}})\}_{j=1}^{d_x}) \,. \tag{8.2}$$

A CGM allows us to describe the effect of *interventions*. For simplicity, we consider only perfect and deterministic interventions, so that an intervention is characterized by (i) a set of targeted variables and (ii) their respective values during the intervention. Let $I \subseteq \{1, ..., d_x\}$ be a set of targets and $\boldsymbol{x}^0 \in \mathbb{R}^{d_x}$ the vector of corresponding target values (note that only $\boldsymbol{x}_I^0$ will be relevant). An intervention is thus completely characterized by $(I, \boldsymbol{x}^0)$. If $|I| = 1$, we say the intervention is

single-target, otherwise we say it is *multi-target*. The interventional distribution for $(I, \boldsymbol{x}^0)$ is given by:

$$p^{(I,\boldsymbol{x}^0)}(\boldsymbol{x}) := \prod_{j \notin I} \boldsymbol{f}_j(\boldsymbol{x}_j \mid \boldsymbol{x}_{\pi_j^{\mathcal{G}}}) \prod_{j \in I} \mathbb{1}[\boldsymbol{x}_j = \boldsymbol{x}_j^0] \,. \tag{8.3}$$

See Section 2.3 for more details on CGMs.

Loss function $\ell(\boldsymbol{\theta}, \boldsymbol{a})$. The goal of the decision maker is to output a CGM

$$\boldsymbol{a} := (\hat{\mathcal{G}}, \{\hat{\boldsymbol{f}}_j(\boldsymbol{x}_j \mid \boldsymbol{x}_{\pi_j^{\hat{\mathcal{G}}}})\}_{j=1}^{d_x}) \,,$$

which is capable of predicting the effect of *any possible intervention*. This is captured, for example, by the following loss:

$$\ell(\boldsymbol{\theta}, \boldsymbol{a}) := \sum_{I \in \mathcal{P}([d_x])} \max_{\boldsymbol{x}^0} D_{KL}(p^{(I,\boldsymbol{x}^0)} \parallel \hat{p}^{(I,\boldsymbol{x}^0)}) \tag{8.4}$$

where $\mathcal{P}([d_x])$ is the power set of $[d_x] = \{1, ..., d_z\}$ and $\hat{p}^{(I,\boldsymbol{x}^0)}$ is the interventional density of the estimated model $\boldsymbol{a} := (\hat{\mathcal{G}}, \{\hat{\boldsymbol{f}}_j(\boldsymbol{x}_j \mid \boldsymbol{x}_{\pi_j^{\hat{\mathcal{G}}}})\}_{j=1}^{d_x})$ under intervention $(I, \boldsymbol{x}^0)$. In some settings, one might not care about *all* possible interventions and might instead focus only on a subset, in which case the loss could be adjusted accordingly.

Observable O. In Chapter 3, the learner has only access to samples from the observational distribution, while in Chapter 4 it has access to multiple interventional distributions. We will focus on the latter case here. We assume the learner observes a list of interventions given by

$$O := \{(I_0, p^{(I_0,\boldsymbol{x}_0^0)}(\boldsymbol{x})), (I_1, p^{(I_1,\boldsymbol{x}_1^0)}(\boldsymbol{x})), \ldots, (I_K, p^{(I_K,\boldsymbol{x}_K^0)}(\boldsymbol{x}))\} \,, \tag{8.5}$$

where $I_0 = \emptyset$, i.e. it corresponds to the observational distribution.

Decision rule δ. In Chapter 4, we suggest the following decision rule:

$$\delta^{\lambda}(O) := \operatorname*{arg\,max}_{\hat{\mathcal{G}} \in \text{DAG}, \{\hat{f}_j\}_j} \sum_{k=0}^{K} \mathbb{E}_{\boldsymbol{x} \sim p^{(k)}} \log \hat{p}^{(k)}(\boldsymbol{x}) - \lambda |\mathcal{G}| \,, \tag{8.6}$$

where $p^{(k)}$ and $\hat{p}^{(k)}$ are the kth interventional density of the ground-truth and learned model, respectively. Furthermore, $|\mathcal{G}|$ is the number of edges in $\mathcal{G}$ and $\lambda > 0$ is a regularization coefficient.

The 3 steps to generalization. I spell out the three steps to prove generalization to unseen interventions:

- *Step 1: Choose assumptions.* We assume that interventions performed in the real world are well modelled by interventions in a CGM (Section 2.3). More precisely, we assume the existence of a graph $\mathcal{G}$ such that the conditionals $f(\boldsymbol{x}_i \mid \boldsymbol{x}_{\pi_i^{\mathcal{G}}})$ do not change across interventions in which $\boldsymbol{x}_i$ is not intervened upon.

- *Step 2: Prove identifiability.* Theorem 4.1 provides conditions so that the causal graph estimated by the decision rule δ^λ must be $\mathcal{I}$-Markov equivalent to the ground-truth (see Chapter 4 for details). When we have sufficiently many interventions, the $\mathcal{I}$-Markov equivalence class becomes a singleton, meaning we identify the ground-truth graph exactly.
- *Step 3: Prove generalization.* Since the learned causal graph $\hat{\mathcal{G}}$ matches the ground-truth graph $\mathcal{G}$ exactly, the learned model will correctly predict the effect of unseen interventions, i.e. $p^{(I,x_0)} = \hat{p}^{(I,x_0)}$ for all interventions (I, x_0), which implies $\ell(\boldsymbol{\theta}, \delta^\lambda(O)) = 0$.

8.2.2. Additive decoders for extrapolation

In Chapter 7, we showed how solving a simple reconstruction task with an *additive decoder* can yield a disentangled representation which allows for a form of extrapolation we call *Cartesian-product extrapolation*. In this section, I formulate this extrapolation problem within statistical decision theory.

Problem setting and "state of the world" $\boldsymbol{\theta}$. I briefly recall the setting of Chapter 7. The set of *possible observations*, e.g. images, is given by a lower dimensional manifold $\boldsymbol{f}(\mathcal{Z}^{\text{test}})$ embedded in $\mathbb{R}^{d_x}$ where $\mathcal{Z}^{\text{test}}$ is an open set of $\mathbb{R}^{d_z}$ and $\boldsymbol{f}$ is a C^2-diffeomorphism onto its image. We will refer to $\boldsymbol{f}$ as the ground-truth decoder. At this stage, we have not specified any distribution over the possible observations, we have simply assumed that they are supported on a lower dimensional manifold. We will further assume that, at training time, the observations $\boldsymbol{x}$ are i.i.d. samples given by $\boldsymbol{x} = \boldsymbol{f}(\boldsymbol{z})$ where $\boldsymbol{z}$, the vector of ground-truth latent factors, is distributed according to the probability measure $\mathbb{P}_{\boldsymbol{z}}^{\text{train}}$ with support $\mathcal{Z}^{\text{train}} \subseteq \mathcal{Z}^{\text{test}}$. A key aspect of the framework is that we allow for $\mathcal{Z}^{\text{train}} \neq \mathcal{Z}^{\text{test}}$, i.e. not all possible observations are revealed at training time. To allow for disentanglement and Cartesian-product extrapolation, we further assume that $\boldsymbol{f}$ is additive in the sense that $\boldsymbol{f}(\boldsymbol{z}) = \sum_{B \in \mathcal{B}} \boldsymbol{f}^{(B)}(\boldsymbol{z}_B)$ where $\mathcal{B} := \{B_1, ..., B_\ell\}$ is a partition of $\{1, 2, \dots, d_z\}$. The state of the world is thus given by

$$\boldsymbol{\theta} := (\mathbb{P}_{\boldsymbol{z}}^{\text{train}}, \boldsymbol{f} : \mathcal{Z}^{\text{test}} \to \mathbb{R}^{d_x}) \,. \tag{8.7}$$

Loss function $\ell(\boldsymbol{\theta}, \boldsymbol{a})$. The goal of the learner is to output an encoder-decoder pair $(\hat{\boldsymbol{g}}, \hat{\boldsymbol{f}})$ that can "extrapolate" beyond the support of observed data seen during training, i.e. $\boldsymbol{f}(\mathcal{Z}^{\text{train}})$. We will formalize this goal with the loss function $\ell(\boldsymbol{\theta}, (\hat{\boldsymbol{g}}, \hat{\boldsymbol{f}}))$. Let $\hat{\mathcal{Z}}^{\text{train}} := \hat{\boldsymbol{g}}(\boldsymbol{f}(\mathcal{Z}^{\text{train}}))$, i.e. the support of estimated latent factors seen during training. Furthermore, recall the definition of the *Cartesian-product extension* of a set $\mathcal{Z} \subseteq \mathbb{R}^{d_z}$:

$$\text{CPE}_{\mathcal{B}}(\mathcal{Z}) := \prod_{B \in \mathcal{B}} \mathcal{Z}_B \,, \text{where } \mathcal{Z}_B := \{\boldsymbol{z}_B \mid \boldsymbol{z} \in \mathcal{Z}\}. \tag{8.8}$$

See Figure 7.3 for an illustration of the CPE. Note that $\mathcal{Z}_B$ is the projection of $\mathcal{Z}$ on the coordinates i in B. With this notation in hand, we can specify our loss function for Cartesian-product

extrapolation:

$$\ell(\boldsymbol{\theta}, (\hat{\boldsymbol{g}}, \hat{\boldsymbol{f}})) := \max_{\boldsymbol{z} \in \text{CPE}_{\mathcal{B}}(\mathcal{Z}^{\text{train}})} \min_{\hat{\boldsymbol{z}} \in \text{CPE}_{\mathcal{B}}(\hat{\mathcal{Z}}^{\text{train}})} \|\boldsymbol{f}(\boldsymbol{z}) - \hat{\boldsymbol{f}}(\hat{\boldsymbol{z}})\|_2^2 , \tag{8.9}$$

so that the loss is zero if and only if, for all $\boldsymbol{z}$ in the Cartesian-product extension of $\mathcal{Z}^{\text{train}}$, there exists a $\hat{\boldsymbol{z}}$ in the Cartesian-product extension of $\hat{\mathcal{Z}}^{\text{train}}$ such that $\boldsymbol{f}(\boldsymbol{z}) = \hat{\boldsymbol{f}}(\hat{\boldsymbol{z}})$. Essentially, this loss measures how well the learned decoder $\hat{\boldsymbol{f}}$ can produce observations that agree with the ground-truth model, even if these observations were never seen at training time. We assume implicitly that the $\max$ and $\min$ exist, which would be the case, for example, if $\mathcal{Z}^{\text{train}}$ and $\hat{\mathcal{Z}}^{\text{train}}$ are compact and $\boldsymbol{f}$ and $\hat{\boldsymbol{f}}$ are continuous.

Note that, a priori, this loss function "does not care" whether $\hat{\boldsymbol{f}}$ is disentangled or not. In principle, one could choose $\hat{\boldsymbol{f}}$ to be entangled w.r.t. $\boldsymbol{f}$, but still have a zero loss.

Observable O. For this setting, the only thing the learner has access to is the distribution over observations, i.e. $\mathbb{P}_{\boldsymbol{x}}^{\text{train}} := \boldsymbol{f}(\mathbb{P}_{\boldsymbol{z}}^{\text{train}})$ (the pushforward measure):

$$O := \mathbb{P}_{\boldsymbol{x}}^{\text{train}} . \tag{8.10}$$

Decision rule δ. The decision rule suggested in Chapter 7 is simply to solve the reconstruction problem:

$$\delta^{\mathcal{F}}(O) := \operatorname*{arg\,min}_{(\hat{\boldsymbol{g}}, \hat{\boldsymbol{f}}) \in \mathcal{G} \times \mathcal{F}} \mathbb{E}^{\text{train}} \|\boldsymbol{x} - \hat{\boldsymbol{f}}(\hat{\boldsymbol{g}}(\boldsymbol{x}))\|_2^2 , \tag{8.11}$$

where $\mathcal{G}$ is a class of functions from $\mathbb{R}^{d_x}$ to $\mathbb{R}^{d_z}$ and $\mathcal{F}$ is a class of functions from $\mathbb{R}^{d_z}$ to $\mathbb{R}^{d_x}$.

The main contributions of Chapter 7 are to (i) provide conditions on $\mathbb{P}_{\boldsymbol{z}}^{\text{train}}$, $\boldsymbol{f}$, and $\mathcal{F}$ so that the learned decoder $\hat{\boldsymbol{f}}$ is disentangled, and (ii) further show that these conditions are also sufficient for Cartesian-product extrapolation. The central assumption making both of these possible is *additivity*. Indeed, among other assumptions, we require both the ground-truth decoder $\boldsymbol{f}$ and the function class $\mathcal{F}$ to be additive.

The 3 steps to generalization. We spell out the 3 steps leading to extrapolation guarantees in this problem setting:

- *Step 1: Choose assumptions.* Simple images consisting of multiple objects have an additive structure in the sense that they are well modelled by $\boldsymbol{x} = \sum_{B \in \mathcal{B}} \boldsymbol{f}^{(B)}(\boldsymbol{z}_B)$ where $\boldsymbol{z}$ is a random vector.
- *Step 2: Prove identifiability.* Chapter 7 shows the representation is identifiable up to permutation of the blocks and block-wise invertible transformations.
- *Step 3: Prove generalization.* This identifiability results leads to Corollary 7.1 which states that there exists a diffeomorphism $\bar{\boldsymbol{v}} : \text{CPE}_{\mathcal{B}}(\mathcal{Z}^{\text{train}}) \to \text{CPE}_{\mathcal{B}}(\hat{\mathcal{Z}}^{\text{train}})$ such that

$$\text{for all } \boldsymbol{z} \in \text{CPE}_{\mathcal{B}}(\mathcal{Z}^{\text{train}}),\ \boldsymbol{f}(\boldsymbol{z}) = \hat{\boldsymbol{f}}(\bar{\boldsymbol{v}}(\boldsymbol{z})) . \tag{8.12}$$

This is extrapolation since the learned decoder $\hat{\boldsymbol{f}}$ imitates the ground-truth one, not only on $\mathcal{Z}^{\text{train}}$ but over all of its CPE. Note that (8.12) implies the extrapolation loss $\ell(\boldsymbol{\theta}, \delta^{\mathcal{F}}(O))$ defined earlier must be zero. Indeed, for each $\boldsymbol{z} \in \text{CPE}_{\mathcal{B}}(\mathcal{Z}^{\text{train}})$, we can take $\hat{\boldsymbol{z}} := \bar{\boldsymbol{v}}(\boldsymbol{z}) \in \text{CPE}_{\mathcal{B}}(\hat{\mathcal{Z}}^{\text{train}})$ so that $\|\boldsymbol{f}(\boldsymbol{z}) - \hat{\boldsymbol{f}}(\hat{\boldsymbol{z}})\|_2^2 = 0$. Importantly, this is possible only because $\mathcal{F}$ is restricted to additive decoders. Without such a restriction, the behavior of $\hat{\boldsymbol{f}}$ outside $\hat{\mathcal{Z}}^{\text{train}}$ is unrestricted, meaning it could be arbitrarily different from $\boldsymbol{f}$.

8.2.3. Sparse multi-task learning

I review the setting of Chapter 6 and express it in the language of statistical learning theory. The contribution of Chapter 6 is separated in two major points: First, we showed that, in a few-shot learning setting, a disentangled representation combined with sparsity regularization can offer benefits in terms of sample complexity, i.e. the number of samples required for learning is reduced. Secondly, we showed how one can learn a disentangled representation via sparse multitask learning. In this section, I combine both of these into a single decision rule which:

(1) learns a disentangled representation $\hat{\boldsymbol{f}}$ using sparse multi-task learning; and
(2) leverages this representation to perform well in an unseen few-shot task.

Problem setting and "state of the world" $\boldsymbol{\theta}$. I start by recalling the setting at hand. We assume the learner observes a family of prediction tasks parameterized by $\boldsymbol{W} \in \mathcal{W} \subseteq \mathbb{R}^{k \times m}$ where the input-label pairs $(\boldsymbol{x}, y)$ are i.i.d. samples from

$$p(\boldsymbol{x}, y \mid \boldsymbol{W}) := p(y \mid \boldsymbol{x}, \boldsymbol{W}) p(\boldsymbol{x} \mid \boldsymbol{W}) \tag{8.13}$$

$$\text{with } p(y \mid \boldsymbol{x}, \boldsymbol{W}) := p(y; \boldsymbol{\eta} = \boldsymbol{W}\boldsymbol{f}(\boldsymbol{x}))\,, \tag{8.14}$$

where $p(y; \boldsymbol{\eta})$ is a density/probability mass function parameterized by $\boldsymbol{\eta} \in \mathbb{R}^k$. The key idea is that the conditional distribution of y given $\boldsymbol{x}$ is modulated via $\boldsymbol{W}\boldsymbol{f}(\boldsymbol{x})$ where $\boldsymbol{W}$ can change across tasks while the representation $\boldsymbol{f}$ is shared across tasks. We also assume that, for each task, the parameter $\boldsymbol{W}$ is sampled i.i.d. from some distribution $\mathbb{P}_{\boldsymbol{W}}$. One of the key assumption is that $\mathbb{P}_{\boldsymbol{W}}$ puts nonzero probability mass on sparse parameters $\boldsymbol{W}$, modelling the assumption that, for a given task, only a few features are useful. We will also denote by $\boldsymbol{W}_{\text{new}}$ the task-specific weight matrix for the unseen few-shot task (second step mentioned above). The learner will have access to a small dataset of n input-label pairs sampled from this task:

$$D_{\text{new}} := ((\boldsymbol{x}^i, y^i))_{i=1}^n \sim \prod_{i=1}^n p(y^i, \boldsymbol{x}^i \mid \boldsymbol{W}_{\text{new}})\,. \tag{8.15}$$

We are now in a position to specify the "state of the world" $\boldsymbol{\theta}$ for this specific setting.

$$\boldsymbol{\theta} := (\boldsymbol{f}, \mathbb{P}_{\boldsymbol{W}}, p(y; \boldsymbol{\eta}), p(\boldsymbol{x} \mid \boldsymbol{W}), \boldsymbol{W}_{\text{new}}) \tag{8.16}$$

Loss function $\ell(\boldsymbol{\theta}, \boldsymbol{a})$. The end goal here will be to decide on a representation $\hat{\boldsymbol{f}}$ and a parameter $\hat{\boldsymbol{W}}_{\text{new}}$, the latter of which is specific to the unseen task $p(\boldsymbol{x}, y \mid \boldsymbol{W}_{\text{new}})$. We will take the loss to be the negative log-likelihood averaged over the whole test distribution. More precisely, we have

$$\ell(\boldsymbol{\theta}, (\hat{\boldsymbol{f}}, \hat{\boldsymbol{W}}_{\text{new}})) := -\mathbb{E}_{p(\boldsymbol{x}, y \mid \boldsymbol{W}_{\text{new}})} \log p(y; \hat{\boldsymbol{W}}_{\text{new}} \hat{\boldsymbol{f}}(\boldsymbol{x})) \,. \tag{8.17}$$

The above is essentially a measure of how well $(\hat{\boldsymbol{f}}, \hat{\boldsymbol{W}}_{\text{new}})$ performs on the new task $\boldsymbol{W}_{\text{new}}$.

Observables O. Before specifying the observables O, we recall a crucial identifiability result from Chapter 6. This chapter introduced two similar results guaranteeing the identifiability of $\boldsymbol{f}$ up to permutation and rescaling, namely Theorems 6.1 & 6.4. Because the latter requires less hyperparameters, it will be the focus of this section, out of a desire for simplicity. We restate Theorem 6.4, which was introduced in Appendix B.2 of Chapter 6, in an adapted form:

Theorem 8.1 (Sparse multitask learning for disentanglement)**.** *Let $\hat{\boldsymbol{f}}^{\alpha}$ be a minimizer of*

$$\begin{aligned} &\min_{\hat{\boldsymbol{f}} \in \mathcal{F}} \mathbb{E}_{\mathbb{P}_{\boldsymbol{W}}} \mathbb{E}_{p(\boldsymbol{x}, y \mid \boldsymbol{W})} - \log p(y; \hat{\boldsymbol{W}}^{(\boldsymbol{W})} \hat{\boldsymbol{f}}(\boldsymbol{x})) \\ &\text{s.t.} \quad \forall\, \boldsymbol{W} \in \mathcal{W},\ \hat{\boldsymbol{W}}^{(\boldsymbol{W})} \in \arg\min_{\tilde{\boldsymbol{W}}} \mathbb{E}_{p(\boldsymbol{x}, y \mid \boldsymbol{W})} - \log p(y; \tilde{\boldsymbol{W}} \hat{\boldsymbol{f}}(\boldsymbol{x})) \\ &\qquad \mathbb{E}_{\mathbb{P}_{\boldsymbol{W}}} \|\hat{\boldsymbol{W}}^{(\boldsymbol{W})}\|_{2,0} \leq \alpha \ , \end{aligned} \tag{8.18}$$

where $\mathbb{P}_{\boldsymbol{W}}$ and $p(\boldsymbol{x}, y \mid \boldsymbol{W})$ are described above. Under Assumptions 6.3, 6.4, 6.5, 6.6, 6.7 and if all functions in $\mathcal{F}$ are continuous and $\alpha = \mathbb{E}_{\mathbb{P}_{\boldsymbol{W}}} \|\boldsymbol{W}\|_{2,0}$, $\hat{\boldsymbol{f}}^{\alpha}$ is disentangled w.r.t. $\boldsymbol{f}$ (Definition 6.1).

Without restating all assumptions precisely, we simply recall that Assumptions 6.3 to 6.7 concern the task and data-generating distributions $\mathbb{P}_{\boldsymbol{W}}$ and $p(\boldsymbol{x}, y \mid \boldsymbol{W})$, including the ground-truth representation $\boldsymbol{f}$. The theorem guarantees that $\hat{\boldsymbol{f}}^{\alpha}(\boldsymbol{x}) = \boldsymbol{D}\boldsymbol{P}\boldsymbol{f}(\boldsymbol{x})$ for all $\boldsymbol{x}$ in the support of $p(\boldsymbol{x} \mid \boldsymbol{W})$ (which we assume is shared across $\boldsymbol{W}$), for some diagonal matrix $\boldsymbol{D}$ and permutation matrix $\boldsymbol{P}$.

The framework of statistical decision theory requires us to be precise about what is observed by and hidden from the learner. The distributions $\mathbb{P}_{\boldsymbol{W}}$ and $p(\boldsymbol{x}, y \mid \boldsymbol{W})$ appear in Problem 8.18, which falsely suggests the learner must observe $\boldsymbol{W}$. If that were the case, it would be absurd to learn a $\hat{\boldsymbol{W}}$ for every $\boldsymbol{W}$ as we do here. It turns out we can rewrite Problem 8.18 in a way that makes it clear that the learner does not need to know about $\boldsymbol{W}$. The idea is to perform a change of variable $T = \boldsymbol{\phi}(\boldsymbol{W})$ where $\boldsymbol{\phi}$ is some bijective function *unknown to the learner*. This of course induces a new measure $\mathbb{Q}_T := \boldsymbol{\phi}(\mathbb{P}_{\boldsymbol{W}})$ with support $\mathcal{T} := \boldsymbol{\phi}(\mathcal{W})$ and a new conditional distribution

$q(\boldsymbol{x}, y \mid T) := p(\boldsymbol{x}, y \mid \boldsymbol{\phi}^{-1}(T))$. With this new notation, we can rewrite Problem 8.18 as

$$\begin{aligned} &\min_{\hat{\boldsymbol{f}} \in \mathcal{F}} \mathbb{E}_{\mathbb{Q}_T} \mathbb{E}_{q(\boldsymbol{x}, y|T)} - \log p(y; \hat{\boldsymbol{W}}^{(T)} \hat{\boldsymbol{f}}(\boldsymbol{x})) \\ &\text{s.t.} \quad \forall\, T \in \mathcal{T},\ \hat{\boldsymbol{W}}^{(T)} \in \arg\min_{\tilde{\boldsymbol{W}}} \mathbb{E}_{q(\boldsymbol{x}, y|T)} - \log p(y; \tilde{\boldsymbol{W}} \hat{\boldsymbol{f}}(\boldsymbol{x})) \\ &\qquad \mathbb{E}_{\mathbb{Q}_T} \|\hat{\boldsymbol{W}}^{(T)}\|_{2,0} \leq \alpha \ . \end{aligned} \tag{8.19}$$

One should think of T as a special task index belonging to an uncountable[4] space $\mathcal{T} = \boldsymbol{\phi}(\mathcal{W})$ distributed according to $\mathbb{Q}_T$. Essentially, the role of $\mathbb{Q}_T$ is to induce a measure "over tasks" $q(\boldsymbol{x}, y \mid T)$. This can be thought of as having an (uncountably) infinite number of tasks. Of course, in practice this is impossible, but we will make this assumption nonetheless.

With this formulation in mind, the observables are "a distribution over tasks" specified by both $\mathbb{Q}_T$ and $q(\boldsymbol{x}, y \mid T)$ and a finite dataset D_{new} from (8.15), i.e.

$$O := (\mathbb{Q}_T, q(\boldsymbol{x}, y \mid T), D_{\text{new}}) \,. \tag{8.20}$$

Decision rule δ. The decision rule we propose is given by

$$\delta^{\alpha,\beta}(O) := (\hat{\boldsymbol{f}}^{\alpha}, \hat{\boldsymbol{W}}^{\alpha,\beta}_{\text{new}}) \,, \tag{8.21}$$

$$\text{where } \hat{\boldsymbol{W}}^{\alpha,\beta}_{\text{new}} := \arg\max_{\|\hat{\boldsymbol{W}}\|_{2,0} \leq \beta} \frac{1}{n} \sum_{i=1}^{n} \log p(y^i; \hat{\boldsymbol{W}} \hat{\boldsymbol{f}}^{\alpha}(\boldsymbol{x}^i)) \tag{8.22}$$

and $\hat{\boldsymbol{f}}^{\alpha}$ is the solution to problem (8.19). This decision rule can be understood as a two-step procedure. First, a representation $\hat{\boldsymbol{f}}^{\alpha}$ is learned via (sparse) multi-task learning (Problem (8.19)) and, secondly, a task-specific predictor $\hat{\boldsymbol{W}}^{\alpha,\beta}_{\text{new}}$ is learned given only a small number of labelled samples from the unseen test distribution $p(\boldsymbol{x}, y \mid \boldsymbol{W}_{\text{new}})$. The hyperparameters α and β control the sparsity levels of the first and second learning stage, respectively. Note that $\alpha = \infty = \beta$ corresponds to the case without any sparsity regularization.

The 3 steps to generalization. We spell out the three steps leading to improved sample complexity on the downstream few-shot task:

- *Step 1: Choose assumptions.* We assume that the set of supervised prediction tasks of interests can be solved using a common representation $\boldsymbol{f}(\boldsymbol{x})$ where each task requires only a sparse subset of the features to be solved with a linear predictor. This assumption is formally encoded by the fact that $\mathbb{P}_{\boldsymbol{W}}$ puts nonzero probability on sparse matrices and the fact that $\boldsymbol{W}_{\text{new}}$ is assumed sparse.

[4] Assumption 6.6 requires $\mathcal{W}$ to be uncountable and, since ϕ is a bijection, so is $\phi(\mathcal{W})$.

		$\hat{\boldsymbol{f}}^{\alpha}$ disentangled?	$\hat{\boldsymbol{W}}^{\alpha,\beta}_{\text{new}}$ has zero approximation error? (no bias?)	$\hat{\boldsymbol{W}}^{\alpha,\beta}_{\text{new}}$ has lower estimation error? (low variance?)
$\alpha = \infty$	$\beta = \infty$	No	Yes	No
$\alpha = \infty$	$\beta = \|\boldsymbol{W}_{\text{new}}\|_{2,0}$	No	No	Yes
$\alpha = \mathbb{E}_{\mathbb{P}_{\boldsymbol{W}}} \|\boldsymbol{W}\|_{2,0}$	$\beta = \infty$	Yes	Yes	No
$\alpha = \mathbb{E}_{\mathbb{P}_{\boldsymbol{W}}} \|\boldsymbol{W}\|_{2,0}$	$\beta = \|\boldsymbol{W}_{\text{new}}\|_{2,0}$	Yes	Yes	Yes

Table 8.2. I summarize all four possibilities for the sparsity regularization parameters α and β. For each possibility, I describe whether the learned representation $\hat{\boldsymbol{f}}^{\alpha}$ is disentangled and whether $\hat{\boldsymbol{W}}^{\alpha,\beta}_{\text{new}}$ is unbiased and has lower variance (i.e. better sample complexity). First, as Theorem 8.1 shows, having $\alpha = \mathbb{E}_{\mathbb{P}_{\boldsymbol{W}}} \|\boldsymbol{W}\|_{2,0}$, guarantees disentanglement (under some assumptions). Adding sparsity regularization when estimating $\boldsymbol{W}_{\text{new}}$ always lowers variance (assuming $\boldsymbol{W}_{\text{new}}$ is sparse), but biases the estimator when $\hat{\boldsymbol{f}}^{\alpha}$ is not disentangled. Having the right amount of regularization for both stages of the decision rule yields the best of both worlds: no biased and an improved sample complexity. See Chapter 6 for more details.

- *Step 2: Prove identifiability.* Under suitable assumptions, Theorem 8.1 establishes that adding the proper amount of regularization, specifically setting $\alpha := \mathbb{E}_{\mathbb{P}_{\boldsymbol{W}}} \|\boldsymbol{W}\|_{2,0}$, will force the learned representation $\hat{\boldsymbol{f}}^{\alpha}$ to be disentangled.
- *Step 3: Prove generalization.* Chapter 6 also argues that adding the proper level of sparsity regularization in the second learning stage, by setting $\beta := \|\boldsymbol{W}_{\text{new}}\|_{2,0}$, will lead to improved estimation error (lower variance) without introducing any approximation error (no bias), as long as the representation learned in the first stage, $\hat{\boldsymbol{f}}^{\alpha}$, is disentangled. All four combinations of regularization are listed in Table 8.2 with the resulting effect on disentanglement, the approximation error of $\hat{\boldsymbol{W}}^{\alpha,\beta}_{\text{new}}$ and whether its estimation error is reduced (lower variance). The case where both types of regularization are active is a win-win situation: We have lower estimation error without approximation error. The table reflect ideal values for the hyperparameters, which, in practice are not known by the learner. While the selection of β could be addressed via standard cross-validation on D_{new}, the selection of α is less obvious. See Duan et al. [2020] for an unsupervised strategy to perform hyperparameter selection for disentanglement.

8.2.4. Semi-supervised learning via clustering

I now discuss the problem of semi-supervised learning [Chapelle et al., 2006] through the lens of statistical decision theory. I will concentrate on strategies based on clustering and how identifiability is absolutely crucial for these methods to work.

Problem setting and "state of the world" $\boldsymbol{\theta}$. Semi-supervised learning refers to methods that leverage unlabeled data in order to improve prediction. More formally, it is assumed that the learner observes a very large dataset of unlabeled inputs $D_{\text{unlab}} \subseteq \mathcal{X}$ and a smaller dataset of labelled inputs

$D_{\text{lab}} \subseteq \mathcal{X} \times \mathcal{Y}$, where $\mathcal{X}$ and $\mathcal{Y}$ are the input and label spaces, respectively. It is assumed that D_{lab} is sampled from some ground-truth distribution $p(\boldsymbol{x}, y)$ (i.i.d.) and D_{unlab} is sampled from the marginal $p(\boldsymbol{x})$ (i.i.d.). The hope is that the large dataset of unlabeled inputs D_{unlab} can be used to have a better predictor. To simplify our discussion, we will assume that the learner observes $p(\boldsymbol{x})$ directly (infinite data regime) but only gets a finite dataset of labelled samples:

$$D_{\text{lab}} := ((\boldsymbol{x}^i, y^i))_{i=1}^n \sim \prod_{i=1}^n p(\boldsymbol{x}^i, y^i) \,. \tag{8.23}$$

We will make further assumption about the data-generating process. We assume there exists a hidden categorical variable $z \in \{1, ..., k\}$ such that

$$p(\boldsymbol{x}, y, z) = p(\boldsymbol{x} \mid z)p(z)p(y \mid z) \,, \tag{8.24}$$

so that $p(\boldsymbol{x}, y) = \sum_{z=1}^k p(\boldsymbol{x} \mid z)p(z)p(y \mid z)$. This factorization corresponds to the graphical model $\boldsymbol{x} \leftarrow z \rightarrow y$. Intuitively, this factorization implies that all information about $\boldsymbol{x}$ relevant to predict y is completely mediated by the categorical variable z. In this setting, the marginal distribution $p(\boldsymbol{x})$ is a mixture of k components $p(\boldsymbol{x} \mid z)$ for $z \in \{1, ..., k\}$. If the conditional $p(y \mid z)$ is close to being deterministic, this model can be seen as an instantiation of the *cluster assumption* in semi-supervised learning which states that *observations belonging to the same cluster are likely to have the same label* [Chapelle et al., 2006]. This data-generating-process suggests using a clustering algorithm to cluster the large unlabeled dataset and use the cluster identities to predict y more easily.

In sum, the state of the world $\boldsymbol{\theta}$ is given by

$$\boldsymbol{\theta} := (p(\boldsymbol{x} \mid z), p(z), p(y \mid z)) \,. \tag{8.25}$$

Loss function $\ell(\boldsymbol{\theta}, \boldsymbol{a})$. The end goal in semi-supervised learning is simply to find a predictor to predict y from $\boldsymbol{x}$. One way to achieve this is to estimate $p(y \mid \boldsymbol{x})$. So we choose our loss to be

$$\ell(\boldsymbol{\theta}, \hat{p}_{y|\boldsymbol{x}}) := -\mathbb{E}_{p(\boldsymbol{x})} D_{KL}(p(y \mid \boldsymbol{x}) \parallel \hat{p}_{y|\boldsymbol{x}}(y \mid \boldsymbol{x})) \,, \tag{8.26}$$

which reaches its minimal value when $\hat{p}_{y|\boldsymbol{x}}(y \mid \boldsymbol{x}) = p(y \mid \boldsymbol{x})$ (almost) everywhere.[5] Note that, in a classification setting, one could also consider a loss ℓ that measures the accuracy of a classifier $h : \mathcal{X} \rightarrow \mathcal{Y}$, hence removing the need to estimate $p(y \mid \boldsymbol{x})$. However, the point we want to make do not require this extra complication, so we stick with (8.26).

Observables O. As already discussed, the learner has access to the following observables:

$$O := (p(\boldsymbol{x}), D_{\text{lab}}) \,. \tag{8.27}$$

[5] Notice the abuse of notation in (8.26) where we are using conditional densities inside the KL divergence, where actual probability measures should be used. I believe the meaning is clear from context.

We assume $p(\boldsymbol{x})$ is fully known, although in practice only a finite dataset of unlabeled inputs $\boldsymbol{x}$ is given. We still model the fact that the number of labelled samples is finite with the dataset D_{lab}.

The discriminative decision rule δ_{disc}. We start by giving a very simple decision rule as baseline which does not even leverage the knowledge of $p(\boldsymbol{x})$, namely *conditional maximum likelihood estimation* on the labelled dataset $\mathcal{D}_{\text{lab}}$:

$$\delta_{\text{disc}}(O) := \arg\max_{\hat{p}_{y|\boldsymbol{x}} \in \mathcal{P}_{y|\boldsymbol{x}}} \frac{1}{n} \sum_{i=1}^{n} \log \hat{p}_{y|\boldsymbol{x}}(y^i \mid \boldsymbol{x}^i)\,, \tag{8.28}$$

where $\mathcal{P}_{y|\boldsymbol{x}}$ is some hypothesis class of conditional densities $\hat{p}_{y|\boldsymbol{x}}$. We consider this rule to be *discriminative* as opposed to *generative* since it only learns $p(y \mid \boldsymbol{x})$ and not the whole joint $p(y, \boldsymbol{x})$, as the next clustering-based rule does.

The clustering-based decision rule δ_{clus}. The second decision rule we consider is based on a clustering of $p(\boldsymbol{x})$. We will essentially learn all pieces of the data-generating process, namely $p(z)$, $p(\boldsymbol{x} \mid z)$ and $p(y \mid z)$. We will provide some examples of identifiable clustering models later on. The key idea is that, if $p(z)$ and $p(\boldsymbol{x} \mid z)$ are identifiable from the marginal $p(\boldsymbol{x})$, then, the only piece of the model that requires labelled samples to be estimated is $p(y \mid z)$. Importantly, the space of possible $p(y \mid z)$ is typically less "complex" than the space of possible $p(y \mid \boldsymbol{x})$, which means we can obtain better sample complexity for the clustering approach.

The clustering-based rule starts by performing clustering. Here we consider a MLE-based clustering strategy:

$$(\hat{p}_z, \hat{p}_{\boldsymbol{x}|z}) \in \arg\max_{\hat{p}_{\boldsymbol{z}} \in \Delta_k, \hat{p}_{\boldsymbol{x}|z} \in \mathcal{P}_{\boldsymbol{x}|z}} \mathbb{E}_{p(\boldsymbol{x})} \log \underbrace{\sum\nolimits_{z=1}^{k} \hat{p}_{\boldsymbol{x}|z}(\boldsymbol{x} \mid z)\hat{p}_z(z)}_{\hat{p}(\boldsymbol{x}) :=}\,, \tag{8.29}$$

where Δ_k is the $(k-1)$-dimensional simplex and $\mathcal{P}_{\boldsymbol{x}|z}$ is some hypothesis class for the components of the mixture. Note that if the hypothesis class of the components $\mathcal{P}_{\boldsymbol{x}|z}$ is expressive enough to contain the ground-truth components $p(\boldsymbol{x} \mid z)$ and if $\mathcal{P}_{\boldsymbol{x}|z}$ is restricted enough to be identifiable, then (8.29) is guaranteed to recover the exact components, up to permutation. I give a more explicit definition of identifiability for such clustering models below:

Definition 8.1. *We say that a clustering model $\mathcal{P}_{\boldsymbol{x}|z}$ is **identifiable** when*

$$\textit{For all } (\tilde{p}_{\boldsymbol{x}|z}, \tilde{p}_z), (\bar{p}_{\boldsymbol{x}|z}, \bar{p}_z) \in \mathcal{P}_{\boldsymbol{x}|z} \times \Delta_k,\ \tilde{p}(\boldsymbol{x}) = \bar{p}(\boldsymbol{x}) \implies (\tilde{p}_{\boldsymbol{x}|z}, \tilde{p}_z) \sim_{\text{clus}} (\bar{p}_{\boldsymbol{x}|z}, \bar{p}_z)\,,$$

where "$\sim_{\text{clus}}$" means that there exists a permutation $\pi : [k] \to [k]$ such that

$$\bar{p}(\boldsymbol{x} \mid z) = \tilde{p}(\boldsymbol{x} \mid \pi(z)) \textit{ and } \bar{p}(z) = \tilde{p}(\pi(z))\,.$$

The converse statement is always true: if two models are $\sim_{\text{clus}}$-equivalent, then their respective marginal over $\boldsymbol{x}$ must be equal since

$$\bar{p}(\boldsymbol{x}) = \sum_{z=1}^{k} \bar{p}_{\boldsymbol{x}|z}(\boldsymbol{x} \mid z)\bar{p}_z(z) = \sum_{z=1}^{k} \tilde{p}_{\boldsymbol{x}|z}(\boldsymbol{x} \mid \pi(z))\tilde{p}_z(\pi(z)) = \tilde{p}(\boldsymbol{x})\,,$$

where the last equality holds because the permutation simply permutes the terms of the sum.

We further compute the posterior of the fitted model of (8.29):

$$\hat{p}(z \mid \boldsymbol{x}) := \frac{\hat{p}_{\boldsymbol{x}|z}(\boldsymbol{x} \mid z)\hat{p}_z(z)}{\hat{p}(\boldsymbol{x})}\,. \tag{8.30}$$

Again assuming the model $\mathcal{P}_{\boldsymbol{x}|z}$ is identifiable and sufficiently expressive to contain the ground-truth $p(\boldsymbol{x} \mid z)$, we can easily show that $\hat{p}(\boldsymbol{x}) = p(\boldsymbol{x})$ and that $\hat{p}(z \mid \boldsymbol{x}) = p(\pi(z) \mid \boldsymbol{x})$, i.e. the marginal and posterior of the learned and ground-truth models match (up to permutation).

The second stage of the clustering-based rule consists in estimating $p(y \mid z)$ using the labelled samples in D_{lab}:

$$\hat{p}_{y|z} \in \arg\max_{\hat{p}_{y|z} \in \mathcal{P}_{y|z}} \frac{1}{n} \sum_{i=1}^{n} \log \sum_{z=1}^{k} \hat{p}_{y|z}(y^i \mid z)\hat{p}(z \mid \boldsymbol{x}^i)\,, \tag{8.31}$$

where $\mathcal{P}_{y|z}$ is some hypothesis class for the conditional $\hat{p}_{y|z}$. Finally, the clustering-based decision rule is given by

$$\delta_{\text{clus}}(O) := \hat{p}_{y|\boldsymbol{x}} \tag{8.32}$$

$$\text{where } \hat{p}_{y|\boldsymbol{x}}(y \mid \boldsymbol{x}) := \sum_{z=1}^{k} \hat{p}_{y|z}(y \mid z)\hat{p}(z \mid \boldsymbol{x})\,. \tag{8.33}$$

The 3 steps to generalizations. We now show how the three steps of Figure 8.1 applies to this setting to show that the clustering-based approach δ_{clus} has a better bias-variance trade-off than δ_{disc}, assuming the clustering is identifiable. Note that Castelli and Cover [1995] made a similar point assuming identifiability of $\mathcal{P}_{\boldsymbol{x}|z}$ and provided a rigorous sample-complexity analysis of the approach.

- *Step 1: Choose assumptions.* This is the *clustering assumption*. We assumed that the relationship between $\boldsymbol{x}$ and y is mediated by a discrete variable z. More precisely, we assume that $p(y \mid \boldsymbol{x}) = \sum_z p(y \mid z)p(z \mid \boldsymbol{x})$.
- *Step 2: Prove identifiability.* We choose a clustering model $\mathcal{P}_{\boldsymbol{x}|z}$ which contains the ground-truth and is identifiable. There are many identifiability results for clustering in the literature. For example, Teicher [1963] showed that the Gaussian mixture model (where $p(\boldsymbol{x} \mid z)$ is Gaussian for all z) is identifiable. The result was further generalized to any exponential family by Barndorff-Nielsen [1965]. Other models include mixtures of product measures (when $p(\boldsymbol{x} \mid z) = \prod_{i=1}^{d_x} p(\boldsymbol{x}_i \mid z)$) [Teicher, 1967] and symmetric components [Hunter

et al., 2007]. Identifiability can also be obtained for the nonparametric case by assuming separation condition [Aragam et al., 2018] or by assuming a Markov chain over the cluster index z [Gassiat et al., 2016]. The reader is referred to Aragam et al. [2018] for a more comprehensive review.

- *Step 3: Prove generalization.* I now argue that the clustering-based rule δ_{clus} has a better bias-variance trade-off than δ_{disc}.

  The clustering-based rule δ_{clus} has lower estimation error (variance). The variance of δ_{clus} should be lower than that of δ_{disc} because the hypothesis class $\mathcal{P}_{y|\boldsymbol{x}}$, used for δ_{disc}, is typically much more "complex" than $\mathcal{P}_{y|z}$, used for δ_{clus}. Intuitively, this is the case because $\boldsymbol{x}$ lives typically in a high-dimensional Euclidean space while z only takes finitely many values and thus the space functions mapping $\boldsymbol{x}$ to a distribution over y is more complex than the space of functions mapping z to a distribution over y. I illustrate this point in Figure 8.2.

  The clustering-based rule δ_{clus} has zero approximation error (no bias). More precisely, if $\mathcal{P}_{\boldsymbol{x}|z}$ is identifiable up to permutation and $\mathcal{P}_{y|z}$ contains the ground-truth $p(y \mid \pi(z))$ for all permutations π, then δ_{clus} is unbiased because

$$\hat{p}_{y|\boldsymbol{x}}(y \mid \boldsymbol{x}) = \sum_{z=1}^{k} \hat{p}_{y|z}(y \mid z)\hat{p}(z \mid \boldsymbol{x}) \tag{8.34}$$

$$= \sum_{z=1}^{k} \hat{p}_{y|z}(y \mid z)p(\pi(z) \mid \boldsymbol{x}) \tag{8.35}$$

$$= \sum_{z=1}^{k} p_{y|z}(y \mid \pi(z))p(\pi(z) \mid \boldsymbol{x}) = p(y \mid \boldsymbol{x})\,, \tag{8.36}$$

  where the second equality holds because $\mathcal{P}_{\boldsymbol{x}|z}$ is identifiable and the third equality holds when taking $\hat{p}_{y|z}(y \mid z) := p_{y|z}(y \mid \pi(z))$, which we are allowed to do because we assumed $p_{y|z}(y \mid \pi(z)) \in \mathcal{P}_{y|z}$.

What can go wrong without identifiability? We answer this question with an example showing that the identifiability of $\mathcal{P}_{\boldsymbol{x}|z}$ is absolutely crucial to ensure that the rule δ_{clus} has zero approximation error (no bias).

Example 8.1 (Unidentifiable clustering can bias δ_{clus})**.** *Assume that $\boldsymbol{x} \in \mathbb{R}^2$, $y, z \in \{0, 1\}$. Furthermore, assume that the ground-truth components $p(\boldsymbol{x} \mid z = 0)$ and $p(\boldsymbol{x} \mid z = 1)$ are given by the blue and red clusters of data represented in Figure 8.3. Furthermore, we assume that $y = z$ in the data-generating process, which of course implies that $p(y = 1 \mid \boldsymbol{x}) = \mathbb{1}(\boldsymbol{x}_1 \geq 0)$.*

Now, if the hypothesis class of components $\mathcal{P}_{\boldsymbol{x}|z}$ is too expressive (and thus unidentifiable), it is possible that the first clustering stage of δ_{clus} yields components $\hat{p}(\boldsymbol{x} \mid z = 0)$ and $\hat{p}(\boldsymbol{x} \mid z = 1)$ that differ from the ground-truth ones. Figure 8.3 shows such a situation, where the estimated components $\hat{p}(\boldsymbol{x} \mid z = 0)$ and $\hat{p}(\boldsymbol{x} \mid z = 1)$ are depicted in green and orange, respectively. Note

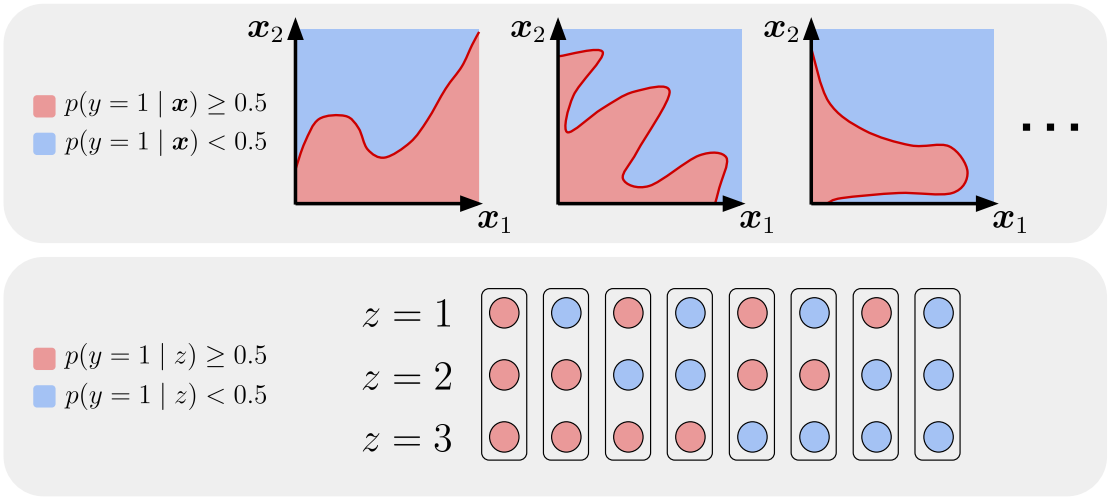


Figure 8.2. Illustrating the fact that $\mathcal{P}_{y|\boldsymbol{x}}$, used in δ_{disc}, is typically much more complex than $\mathcal{P}_{y|z}$, used in δ_{clus}. In this example, we assume $\boldsymbol{x} \in \mathbb{R}^2$, $z \in \{1, 2, 3\}$ and $y \in \{0, 1\}$. We show decision boundaries for both $p(y \mid \boldsymbol{x})$ and $p(y \mid z)$, assuming a threshold of 0.5. Note that we can depict all possible decision boundaries for $p(y \mid z)$ (there are only 2^3 possibilities) while this is impossible for $p(y \mid \boldsymbol{x})$. This observation suggests that δ_{clus} will require less samples to perform well.

that, although the clustering is wrong, the marginal distribution over $\boldsymbol{x}$ is correctly modelled, i.e. $\hat{p}(\boldsymbol{x}) = p(\boldsymbol{x})$. In this case, the posterior of the model is given by $\hat{p}(z = 1 \mid \boldsymbol{x}) = \mathbb{1}(\boldsymbol{x}_2 \geq 0)$. We can thus compute

$$\hat{p}(y = 1 \mid \boldsymbol{x}) = \hat{p}(y = 1 \mid z = 0)\mathbb{1}(\boldsymbol{x}_2 < 0) + \hat{p}(y = 1 \mid z = 1)\mathbb{1}(\boldsymbol{x}_2 \geq 0)\,.$$

It is clear from the above equation that no choice of $\hat{p}(y = 1 \mid z)$ will allow $\hat{p}(y = 1 \mid \boldsymbol{x})$ to exactly match $p(y = 1 \mid \boldsymbol{x}) = \mathbb{1}(\boldsymbol{x}_1 \geq 0)$ for all $\boldsymbol{x}$. Thus, δ_{clus} has a nonzero approximation error, i.e. it is biased.

Remark 8.1. *If we were to modify δ_{clus} so as to train jointly both the clustering model $\hat{p}(\boldsymbol{x} \mid z)$ on $p(\boldsymbol{x})$ and the predictor $\hat{p}(y \mid z)$ on D_{lab}, a solution like the one above would be unlikely since it would yield poor performance on D_{lab}, which would mean the training loss is not minimal.*

Connections to representation learning. The conditionals $\hat{p}(z \mid \boldsymbol{x})$ and $\hat{p}(y \mid z)$ are analogous to the representation $\hat{\boldsymbol{f}}(\boldsymbol{x})$ and the predictor $\hat{\boldsymbol{W}}$ from the previous section, respectively. The motivations are also similar, the representation ($\hat{p}(z \mid \boldsymbol{x})$ or $\hat{\boldsymbol{f}}(\boldsymbol{x})$) is identified using a large dataset while the remaining predictor ($\hat{p}(y \mid z)$ or $\hat{\boldsymbol{W}}$) is learned using a smaller dataset. In both cases, the predictor is much less complex that the whole conditional $p(y \mid \boldsymbol{x})$, which yields benefits in terms of estimation error (lower variance) when learned on smaller datasets.

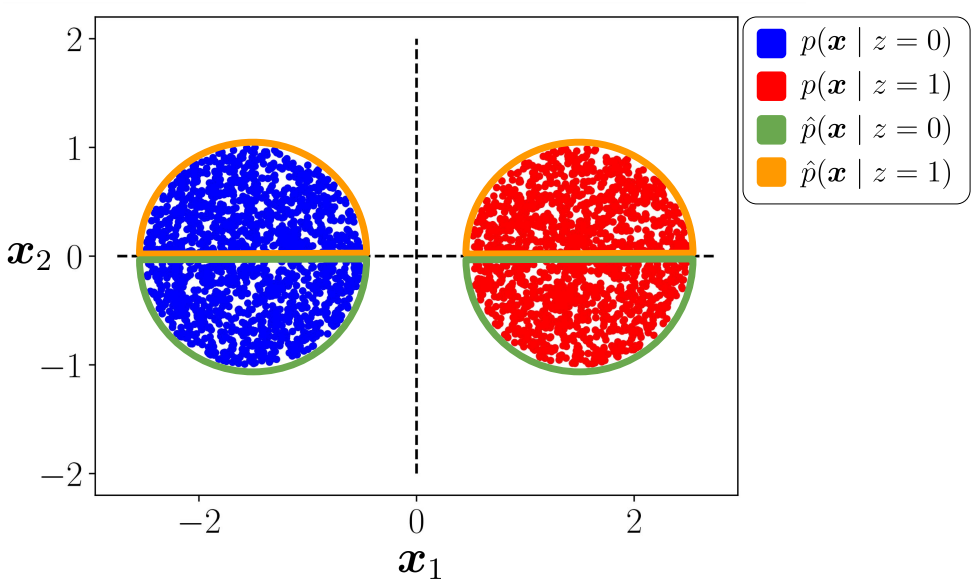


Figure 8.3. Illustration for Example 8.1 showing how an unidentifiable clustering model can bias δ_{clus}. The red and blue clusters correspond to the ground-truth components $p(\boldsymbol{x} \mid z = 0)$ and $p(\boldsymbol{x} \mid z = 1)$, while the region delineated in orange and green correspond to the learned components $\hat{p}(\boldsymbol{x} \mid z = 0)$ and $\hat{p}(\boldsymbol{x} \mid z = 1)$. The latter clustering is valid in the sense that it yields the correct distribution over observations, i.e. $\hat{p}(\boldsymbol{x}) = p(\boldsymbol{x})$, however, it yields a biased δ_{clus}, as explained in Example 8.1.

Chapter 9

Conclusions, Discussions, and Perspectives

This thesis has focused on causal discovery (Chapters 3 & 4), identifiable representation learning (Chapters 5, 6 & 7) and has argued that identifiability analyses are important tools to study the interpretability of models as well as their generalization properties (Chapter 8). In this chapter, we summarize why studying identifiability is important for machine learning (Section 9.1) and discuss future research directions (Section 9.2).

9.1. Why study identifiability?

Scientific understanding. Causal discovery is about automating the scientific process of discovering novel causal relationships. An identifiability guarantee improves trust in the validity of the relationships the model has uncover, conditionally on the assumptions being reasonable and/or testable. For example, biologists are interested in understanding how various genes interact with each other [Lopez et al., 2022]. Latent variable models can also be useful to create a lower dimensional representations of gene perturbations [Lopez et al., 2023]. This is related to the realist interpretation of identifiability discussed in Section 8.1.

Learning interpretable models. A common criticism of deep learning models is that they are black-boxes, i.e. understanding their decision process is difficult. Identifiability can be seen as a necessary condition to make sure a model is interpretable since, without it, multiple reruns of the same algorithm with different initializations will yield different models with different interpretations. This is related to both the "independent-learners interpretation" and the "interpretability interpretation" of identifiability in Section 8.1.

Understanding the behavior of existing algorithms. In Chapter 7 on additive decoders, we provided an identifiability analysis with the goal of shedding light on object-centric representation learning approaches. Why are these methods performing segmentation without any segmentation labels? Although additive decoders are a simplification of the decoders actually used in practice, I

believe a similar style of analysis might provide an answer for more expressive models, probably with more advanced mathematics. The work of Von Kügelgen et al. [2021] is another example where an identifiability analysis is used to gain insight into known algorithms, which in that case are self-supervised learning methods.

Generalization/extrapolation. Chapter 8 showed four different problem settings where identifiability plays a key role in obtaining performance guarantees on some downstream tasks, namely causal discovery (Chapters 3 & 4), sparse multitask learning (Chapter 6), additive decoders for extrapolation (Chapter 7), and semi-supervised learning via clustering (Section 8.2.4). A three-step framework was shown to be common to all four settings: (i) Choose assumptions suitable for the problem at hand postulating that some unknown structure is present in the data, (ii) show that this structure can be recovered from data via an identifiability analysis, and (iii) leverage the learned structure to provide generalization guarantees on the downstream task. Importantly, the last step relies on correctly recovering the structure in the data, which is where identifiability is useful.

9.2. Future research directions

I now briefly describe future research directions either to refine our identifiability analysis or to use identifiability to answer open questions in machine learning.

Finite-sample identifiability analysis. Identifiability, by definition, says nothing about the finite data case. Can we prove some kind of consistency for models that are identifiable only up to some equivalence class? The difficulty stems from the fact that the estimated parameter is in fact an equivalence class of parameters and thus one can only hope to obtain convergence to an equivalence class. How can we formalize convergence in such a setting? Datta and Chakrabarty [2023] showed consistency of probabilistic principal component analysis, which is identifiable only up to some equivalence relation. The key is to consider convergence in some Euclidean quotient space induced by the equivalence relation. Can this sort of approach be extended to more flexible nonlinear latent variable models such as the ones considered in this thesis? What can be said about sample complexity? One could also leverage the fact that deep learning models often operate in the interpolation regime, i.e. the loss is equal to zero for every single data point of the training set. If two models interpolate the data, can we say that their parameters/representations are related via a simple function? If so, does the finite dataset have to satisfy some kind of sufficient variability condition analogous to those of Chapters 5, 6 & 7?

Identifiability analysis with model mispecification. What happens when the fitted model is mispecified in the sense that it cannot express the ground-truth distribution of observations exactly? This question is orthogonal to the question of finite-sample analysis, but as important. Referring to the realist interpretation of Section 8.1, identifiability guarantees assume that the learned model

is expressive enough to represent the data-generating process. What happens when this is not the case? Even defining what we mean by identifiability and discovering structure is difficult in this setting. One would still have to postulate an hypothesis class for the ground-truth model together with a ground-truth representation. For instance, what happens if one performs linear ICA, which assumes the latent factors are statistically independent, on data where the factors are mildly correlated? Proving meaningful theoretical guarantees in such a settings appears to be challenging. Could we still say something meaningful even when the data-generating process has no ground-truth representation? One possible angle could be to sidestep identifiability altogether and analyse generalization/extrapolation directly, since this is often the ultimate goal. Nevertheless, one would still need to assume some structure is present in the data and that somehow the learning algorithm leverages it in order to improve performance on downstream tasks. Instead of going for model mispecification, another direction would simply be to progressively move towards more and more expressive hypothesis class, in the hope that this additional capacity will be enough to avoid model mispecification altogether.

Explaining puzzling observations in deep learning. The literature on deep learning is filled with surprising observations begging for explanations. Examples include the surprising generalization abilities of neural networks [Zhang et al., 2017], the double descent phenomenon [Belkin et al., 2019], linear mode connectivity of neural networks [Garipov et al., 2018, Entezari et al., 2022, Ainsworth et al., 2023], the phenomenon known as grokking [Power et al., 2022], the emergence of interpretable and generalizable algorithms in some neural networks (mechanistic interpretability) [Olah et al., 2020, Elhage et al., 2022] and the surprising creativity of modern generative models [Ramesh et al., 2022]. I believe identifiability can bring insights into some of these puzzles.

- The **linear mode connectivity of neural networks** refers to the observation that, when two neural networks are trained via stochastic gradient descent, the parameters lying on the line $\lambda\boldsymbol{\theta}_1 + (1-\lambda)\boldsymbol{\theta}_0$ joining the parameters of both models $\boldsymbol{\theta}_0$ and $\boldsymbol{\theta}_1$ have as low a loss as $\boldsymbol{\theta}_0$ and $\boldsymbol{\theta}_1$, when adjusting for the permutation invariance of neural networks. Could this observation be explained by the fact that the equivalence class of parameters fitting the data exactly is convex (modulo the permutation indeterminacy)? A better understanding of parameter identifiability in neural networks could answer this question.
- The field of **mechanistic interpretability** aims at discovering interpretable algorithms encoded in the weights of trained neural networks. The precise reason why these algorithms "emerge" is still open (see Morwani et al. [2024] for first steps answering this question). I hypothesize that, in many cases, identifiability can bring insights into this question: Suppose a task is specified by a ground-truth neural network $\boldsymbol{f}_{\boldsymbol{\theta}}$ which implements some specific algorithms in its weights $\boldsymbol{\theta}$ to associate each problem instance $\boldsymbol{x} \in \mathcal{X}$ to a correct output $y = \boldsymbol{f}_{\boldsymbol{\theta}}(\boldsymbol{x})$. Assume that the same neural network architecture with parameter $\hat{\boldsymbol{\theta}}$ is fitted

to each instance so that $\boldsymbol{f}_{\boldsymbol{\theta}}(\boldsymbol{x}) = \boldsymbol{f}_{\hat{\boldsymbol{\theta}}}(\boldsymbol{x})$ for all $\boldsymbol{x} \in \mathcal{X}$. Can we prove that $\hat{\boldsymbol{\theta}}$ implements the same interpretable algorithm as $\boldsymbol{\theta}$, up to irrelevant indeterminacies? This also relates to generalization and extrapolation: If $\boldsymbol{f}_{\boldsymbol{\theta}}(\boldsymbol{x}) = \boldsymbol{f}_{\hat{\boldsymbol{\theta}}}(\boldsymbol{x})$ but only for all $\boldsymbol{x}$ in some subset $\mathcal{X}^{\text{train}} \subseteq \mathcal{X}$, can we show that $\mathcal{X}^{\text{train}}$ is enough to discover the right algorithm that generalizes to $\mathcal{X}$? Can we find meaningful conditions on $\boldsymbol{f}_{\boldsymbol{\theta}}$ and $\mathcal{X}^{\text{train}}$ such that $\boldsymbol{f}_{\boldsymbol{\theta}}(\boldsymbol{x}) = \boldsymbol{f}_{\hat{\boldsymbol{\theta}}}(\boldsymbol{x})$ for all $\boldsymbol{x} \in \mathcal{X}^{\text{train}}$ implies $\boldsymbol{f}_{\boldsymbol{\theta}}(\boldsymbol{x}) = \boldsymbol{f}_{\hat{\boldsymbol{\theta}}}(\boldsymbol{x})$ for all $\boldsymbol{x} \in \mathcal{X}$?

- The **creativity of modern generative models** such as DALLE-2 is mind-boggling. It appears that these model can indeed be creative in that they can recombine known concepts in novel ways, although it is difficult to know for sure due to the immensity of the datasets these models are trained on. But why is this happening? I conjecture that standard generalization theory in machine learning is not enough to account for this kind of out-of-support generation. Chapter 7 demonstrated that identifiability analyses can shed light on extrapolation ability of additive decoders, which are very limited of course. Could this type of analysis be applied to modern generative models?

Bibliography

M. Aharon, M. Elad, and A. Bruckstein. On the uniqueness of overcomplete dictionaries, and a practical way to retrieve them. *Linear Algebra and its Applications*, 2006.

K. Ahuja, K. Shanmugam, K. R. Varshney, and A. Dhurandhar. Invariant risk minimization games. In *International Conference on Machine Learning*, 2020.

K. Ahuja, J. Hartford, and Y. Bengio. Properties from mechanisms: an equivariance perspective on identifiable representation learning. In *International Conference on Learning Representations*, 2022a.

K. Ahuja, J. Hartford, and Y. Bengio. Weakly supervised representation learning with sparse perturbations. *arXiv preprint arXiv:2206.01101*, 2022b.

K. Ahuja, D. Mahajan, V. Syrgkanis, and I. Mitliagkas. Towards efficient representation identification in supervised learning. In *Conference on Causal Learning and Reasoning*, 2022c.

K. Ahuja, D. Mahajan, Y. Wang, and Y. Bengio. Interventional causal representation learning. In *International Conference on Machine Learning*, 2023.

S. Ainsworth, J. Hayase, and S. Srinivasa. Git re-basin: Merging models modulo permutation symmetries. In *International Conference on Learning Representations*, 2023.

J. Alayrac, P. Bojanowski, N. Agrawal, J. Sivic, I. Laptev, and S. Lacoste-Julien. Learning from narrated instruction videos. *TPAMI*, 2018.

Y. Annadani, J. Rothfuss, A. Lacoste, N. Scherrer, A. Goyal, Y. Bengio, and S. Bauer. Variational causal networks: Approximate bayesian inference over causal structures. *arXiv preprint arXiv:2106.07635*, 2021.

B. Aragam, C. Dan, P. Ravikumar, and E. Xing. Identifiability of nonparametric mixture models and bayes optimal clustering. *Annals of Statistics*, 2018.

A. Argyriou, T. Evgeniou, and M. Pontil. Convex multi-task feature learning. *Machine learning*, 2008.

M. Arjovsky, L. Bottou, I. Gulrajani, and D. Lopez-Paz. Invariant risk minimization. *arXiv preprint arXiv:1907.02893*, 2020.

J. L. Ba, J. R. Kiros, and G. E. Hinton. Layer normalization. *arXiv preprint arXiv:1607.06450*, 2016.

A.-L. Barabási. Scale-free networks: a decade and beyond. *Science*, 2009.

A.-L. Barabási and R. Albert. Emergence of scaling in random networks. *Science*, 1999.

O. Barndorff-Nielsen. Identifiability of mixtures of exponential families. *Journal of Mathematical Analysis and Applications*, 1965.

M. Belkin. Fit without fear: remarkable mathematical phenomena of deep learning through the prism of interpolation. *Acta Numerica*, 2021.

M. Belkin, D. Hsu, S. Ma, and S. Mandal. Reconciling modern machine learning practice and the bias-variance trade-off, 2019.

Y. Bengio. Gradient-based optimization of hyperparameters. *Neural computation*, 2000.

Y. Bengio. The consciousness prior. *arXiv preprint arXiv:1709.08568*, 2019.

Y. Bengio, A. Courville, and P. Vincent. Representation learning: A review and new perspectives. *IEEE transactions on pattern analysis and machine intelligence*, 2013.

Y. Bengio, T. Deleu, N. Rahaman, N. R. Ke, S. Lachapelle, O. Bilaniuk, A. Goyal, and C. Pal. A meta-transfer objective for learning to disentangle causal mechanisms. In *International Conference on Learning Representations*, 2020.

Y. Bengio, S. Lahlou, T. Deleu, E. J. Hu, M. Tiwari, and E. Bengio. Gflownet foundations. *Journal of Machine Learning Research*, 2023.

M. Bereket and T. Karaletsos. Modelling cellular perturbations with the sparse additive mechanism shift variational autoencoder. In *Advances in Neural Information Processing Systems*, 2023.

J. O. Berger. *Statistical Decision Theory and Bayesian Analysis*. Springer, 1985.

J. Bergstra and Y. Bengio. Random search for hyper-parameter optimization. *Journal of Machine Learning Research*, 2012.

L. Bertinetto, J. F. Henriques, P. H. Torr, and A. Vedaldi. Meta-learning with differentiable closed-form solvers. 2019.

Q. Bertrand, Q. Klopfenstein, M. Blondel, S. Vaiter, A. Gramfort, and J. Salmon. Implicit differentiation of lasso-type models for hyperparameter optimization. In *International Conference on Machine Learning*, 2020.

Q. Bertrand, Q. Klopfenstein, M. Massias, M. Blondel, S. Vaiter, A. Gramfort, and J. Salmon. Implicit differentiation for fast hyperparameter selection in non-smooth convex learning. *Journal of Machine Learning Research*, 2022.

D. Bertsekas. *Nonlinear Programming*. Athena Scientific, 1999.

M. Besserve, R. Sun, D. Janzing, and B. Schölkopf. A theory of independent mechanisms for extrapolation in generative models. In *AAAI Conference on Artificial Intelligence*, 2021.

P. J. Bickel, Y. Ritov, and A. B. Tsybakov. Simultaneous analysis of lasso and Dantzig selector. *The Annals of statistics*, 2009.

P. Billingsley. *Probability and Measure*. Wiley Series in Probability and Statistics. Wiley, 1995.

S. Bing, U. Ninad, J. Wahl, and J. Runge. Identifying linearly-mixed causal representations from multi-node interventions. *arXiv preprint arXiv:2311.02695*, 2023.

M. Blondel, Q. Berthet, M. Cuturi, R. Frostig, S. Hoyer, F. Llinares-López, F. Pedregosa, and J.-P. Vert. Efficient and modular implicit differentiation. *Journal of Machine Learning Research*, 2022.

J. Bolte, T. Le, E., Pauwels, and T. Silveti-Falls. Nonsmooth implicit differentiation for machine-learning and optimization. *Advances in Neural Information Processing Systems*, 2021.

J. Bolte, E. Pauwels, and S. Vaiter. Automatic differentiation of nonsmooth iterative algorithms. *Advances in Neural Information Processing Systems*, 2022.

L. Bottou. Large-scale machine learning with stochastic gradient descent. In *International Conference on Computational Statistics*. 2010.

D. Bouchacourt, R. Tomioka, and S. Nowozin. Multi-level variational autoencoder: Learning disentangled representations from grouped observations. *AAAI Conference on Artificial Intelligence*, 2018.

S. P. Boyd, , and L. Vandenberghe. *Convex optimization*. Cambridge university press, 2004.

J. Bradbury, R. Frostig, P. Hawkins, M. J. Johnson, C. Leary, D. Maclaurin, G. Necula, A. Paszke, J. VanderPlas, S. Wanderman-Milne, and Q. Zhang. JAX: composable transformations of Python+NumPy programs, 2018. URL http://github.com/google/jax.

J. Brady, R. S. Zimmermann, Y. Sharma, B. Schölkopf, J. von Kügelgen, and W. Brendel. Provably learning object-centric representations. In *International Conference on Machine Learning*, 2023.

J. Brehmer, P. de Haan, P. Lippe, and T. Cohen. Weakly supervised causal representation learning. In *Advances in Neural Information Processing Systems*, 2022.

L. Breiman. Statistical modeling: the two cultures. 2001.

P. Brouillard, S. Lachapelle, A. Lacoste, S. Lacoste-Julien, and A. Drouin. Differentiable causal discovery from interventional data. In *Advances in Neural Information Processing Systems*, 2020.

T. Brown, B. Mann, N. Ryder, M. Subbiah, J. D. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell, S. Agarwal, A. Herbert-Voss, G. Krueger, T. Henighan, R. Child, A. Ramesh, D. Ziegler, J. Wu, C. Winter, C. Hesse, M. Chen, E. Sigler, M. Litwin, S. Gray, B. Chess, J. Clark, C. Berner, S. McCandlish, A. Radford, I. Sutskever, and D. Amodei. Language models are few-shot learners. In *Advances in Neural Information Processing Systems*, 2020.

S. Buchholz, M. Besserve, and B. Schölkopf. Function classes for identifiable nonlinear independent component analysis. In *Advances in Neural Information Processing Systems*, 2022.

S. Buchholz, G. Rajendran, E. Rosenfeld, B. Aragam, B. Schölkopf, and P. K. Ravikumar. Learning linear causal representations from interventions under general nonlinear mixing. In *Advances Neural Information Processing Systems*, 2023.

P. Bühlmann, J. Peters, and J. Ernest. CAM: Causal additive models, high-dimensional order search and penalized regression. *Annals of Statistics*, 2014.

C. Burgess and H. Kim. 3d shapes dataset. https://github.com/deepmind/3dshapes-dataset/, 2018.

C. P. Burgess, L. Matthey, N. Watters, R. Kabra, I. Higgins, M. Botvinick, and A. Lerchner. MONet: Unsupervised scene decomposition and representation. *arXiv preprint arXiv:1901.11390*, 2019.

G. Casella and R. Berger. *Statistical Inference*. Duxbury Resource Center, 2001.

V. Castelli and T. M. Cover. On the exponential value of labeled samples. *Pattern Recogn. Lett.*, 1995.

O. Chapelle, B. Schölkopf, and A. Zien, editors. *Semi-Supervised Learning*. The MIT Press, 2006.

R. T. Q. Chen, X. Li, R. G., and D. Duvenaud. Isolating sources of disentanglement in vaes. In *Advances in Neural Information Processing Systems*, 2018.

S. S. Chen, D. L. Donoho, and M. A. Saunders. Atomic decomposition by basis pursuit. *SIAM Journal on Scientific Computing*, 1998.

T. Chen, S. Kornblith, M. Norouzi, and G. Hinton. A simple framework for contrastive learning of visual representations. In *International Conference on Machine Learning*, 2020.

D. M. Chickering. Optimal structure identification with greedy search. In *Journal of Machine Learning Research*, 2003.

A. Clauset, C. Shalizi, and M. Newman. Power-law distributions in empirical data. *SIAM review*, 2009.

P. Comon. Independent component analysis. *Higher-Order Statistics*, 1992.

P. Comon. Independent component analysis, a new concept? *Signal Processing*, 1994.

K. Crammer and Y. Singer. On the algorithmic implementation of multiclass kernel-based vector machines. *Journal of Machine Learning Research*, 2001.

E. Crawford and J. Pineau. Spatially invariant unsupervised object detection with convolutional neural networks. *AAAI Conference on Artificial Intelligence*, 2019.

J. Cussens. Bayesian network learning with cutting planes. In *Conference on Uncertainty in Artificial Intelligence*, 2011.

G. Darmois. Analyse générale des liaisons stochastiques: etude particulière de l'analyse factorielle linéaire. *Revue de l'Institut International de Statistique*, 1953.

A. Datta and S. Chakrabarty. On the consistency of maximum likelihood estimation of probabilistic principal component analysis. In *Advances in Neural Information Processing Systems*, 2023.

I. Daunhawer, A. Bizeul, E. Palumbo, A. Marx, and J. E. Vogt. Identifiability results for multimodal contrastive learning. In *International Conference on Learning Representations*, 2023.

A. S. d'Avila Garcez and L. Lamb. Neurosymbolic AI: The 3rd wave. *arXiv preprint arXiv:2012.05876*, 2020.

J. P. Day. The uniformity of nature. *American Philosophical Quarterly*, 1975.

T. Deleu, A. Góis, C. C. Emezue, M. Rankawat, S. Lacoste-Julien, S. Bauer, and Y. Bengio. Bayesian structure learning with generative flow networks. In *Conference on Uncertainty in Artificial Intelligence*, 2022.

J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.

A. Dittadi, F. Träuble, F. Locatello, M. Wuthrich, V. Agrawal, O. Winther, S. Bauer, and B. Schölkopf. On the transfer of disentangled representations in realistic settings. In *International Conference on Learning Representations*, 2021.

A. Dittadi, S. S. Papa, M. De Vita, B. Schölkopf, O. Winther, and F. Locatello. Generalization and robustness implications in object-centric learning. In *International Conference on Machine Learning*, 2022.

A. Dixit, O. Parnas, B. Li, J. Chen, C. P. Fulco, L. Jerby-Arnon, N. D. Marjanovic, D. Dionne, T. Burks, R. Raychndhury, T. M. Adamson, B. Norman, E. S. Lander, J. S. Weissman, N. Friedman, and A. Regev. Perturb-seq: dissecting molecular circuits with scalable single-cell rna profiling of pooled genetic screens. *Cell*, 2016.

D. Donoho and C. Grimes. Image manifolds which are isometric to euclidean space. *Journal of Mathematical Imaging and Vision*, 2003a.

D. L. Donoho and C. Grimes. Hessian eigenmaps: Locally linear embedding techniques for high-dimensional data. *National Academy of Sciences*, 2003b.

A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, J. Uszkoreit, and N. Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. In *International Conference on Learning Representations*, 2021a.

A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *International Conference on Learning Representations*, 2021b.

Y. Du and I. Mordatch. Implicit generation and modeling with energy based models. In *Advances in Neural Information Processing Systems*, 2019.

S. Duan, L. Matthey, A. Saraiva, N. Watters, C. Burgess, A. Lerchner, and I. Higgins. Unsupervised model selection for variational disentangled representation learning. In *International Conference on Learning Representations*, 2020.

R. Durrett. *Probability: Theory and examples*. Cambridge University Press, 2011.

C. Eastwood and C. K. Williams. A framework for the quantitative evaluation of disentangled representations. In *International Conference on Learning Representations*, 2018.

D. Eaton and K. Murphy. Exact bayesian structure learning from uncertain interventions. In *International Conference on Artificial Intelligence and Statistics*, 2007.

F. Eberhardt. Causation and intervention. *Unpublished doctoral dissertation, Carnegie Mellon University*, 2007.

F. Eberhardt. Almost Optimal Intervention Sets for Causal Discovery. In *Conference on Uncertainty in Artificial Intelligence*, 2008.

F. Eberhardt and R. Scheines. Interventions and causal inference. *Philosophy of Science*, 2007.

F. Eberhardt, C. Glymour, and R. Scheines. On the Number of Experiments Sufficient and in the Worst Case Necessary to Identify all Causal Relations among N Variables. In *Conference on Uncertainty in Artificial Intelligence*, 2005.

N. Elhage, T. Hume, C. Olsson, N. Schiefer, T. Henighan, S. Kravec, Z. Hatfield-Dodds, R. Lasenby, D. Drain, C. Chen, R. Grosse, S. McCandlish, J. Kaplan, D. Amodei, M. Wattenberg, and C. Olah. Toy models of superposition. *Transformer Circuits Thread*, 2022. https://transformer-circuits.pub/2022/toy_model/index.html.

G. Elidan. Bayesian Network Repository, 2001. https://www.cse.huji.ac.il/ galel/Repository/.

M. Engelcke, A. R. Kosiorek, O. P. Jones, and I. Posner. Genesis: Generative scene inference and sampling with object-centric latent representations. In *International Conference on Learning Representations*, 2020.

R. Entezari, H. Sedghi, O. Saukh, and B. Neyshabur. The role of permutation invariance in linear mode connectivity of neural networks. In *International Conference on Learning Representations*, 2022.

S. M. A. Eslami, N. Heess, T. Weber, Y. Tassa, D. Szepesvari, K. Kavukcuoglu, and G. E. Hinton. Attend, infer, repeat: Fast scene understanding with generative models. In *Advances in Neural Information Processing Systems*, 2016.

C. Fang, Y. Xu, and D. N. Rockmore. Unbiased metric learning: On the utilization of multiple datasets and web images for softening bias. In *IEEE International Conference on Computer Vision*, 2013.

Z. Fang, S. Zhu, J. Zhang, Y. Liu, Z. Chen, and Y. He. On low-rank directed acyclic graphs and causal structure learning. *IEEE Transactions on Neural Networks and Learning Systems*, 2023.

R. Feynman. *The Character of Physical Law*. MIT Press, 1965.

C. Finn, P. Abbeel, and S. Levine. Model-agnostic meta-learning for fast adaptation of deep networks. In *International Conference on Machine Learning*, 2017.

J. A. Fodor and Z. W. Pylyshyn. Connectionism and cognitive architecture: A critical analysis. *Cognition*, 1988.

K. Fukumizu, A. Gretton, X. Sun, and B. Schölkopf. Kernel measures of conditional dependence. In *Advances in Neural Information Processing Systems*, 2008.

M. Fumero, F. Wenzel, L. Zancato, A. Achille, E. Rodolà, S. Soatto, B. Schölkopf, and F. Locatello. Leveraging sparse and shared feature activations for disentangled representation learning. In

Advances in Neural Information Processing Systems, 2023.

J. Gallego-Posada and J. Ramirez. Cooper: a toolkit for lagrangian-based constrained optimization. https://github.com/cooper-org/cooper, 2022.

J. Gallego-Posada, J. Ramirez De Los Rios, and A. Erraqabi. Flexible learning of sparse neural networks via constrained l_0 regularization. In *NeurIPS Workshop LatinX in AI*, 2021.

T. Garipov, P. Izmailov, D. Podoprikhin, D. P. Vetrov, and A. G. Wilson. Loss surfaces, mode connectivity, and fast ensembling of dnns. In *Advances in Neural Information Processing Systems*, 2018.

E. Gassiat, A. Cleynen, and S. Robin. Inference in finite state space non parametric hidden markov models and applications. *Statistics and Computing*, 2016.

A. Gentzel, D. Garant, and D. Jensen. The case for evaluating causal models using interventional measures and empirical data. In *Advances in Neural Information Processing Systems*, 2019.

P. Georgiev, F. Theis, and A. Cichocki. Sparse component analysis and blind source separation of underdetermined mixtures. *IEEE Transactions on Neural Networks*, 2005.

M. Germain, K. Gregor, I. Murray, and H. Larochelle. Made: Masked autoencoder for distribution estimation. In *International Conference on Machine Learning*, 2015.

P. Geurts, D. Ernst, and L. Wehenkel. Extremely randomized trees. *Machine learning*, 2006.

L. Girin, S. Leglaive, X. Bie, J. Diard, T. Hueber, and X. Alameda-Pineda. Dynamical variational autoencoders: A comprehensive review. *arXiv preprint arXiv:2008.12595*, 2020.

X. Glorot and Y. Bengio. Understanding the difficulty of training deep feedforward neural networks. In *International Conference on Artificial Intelligence and Statistics*, 2010a.

X. Glorot and Y. Bengio. Understanding the difficulty of training deep feedforward neural networks. In *International Conference on Artificial Intelligence and Statistics*, 2010b.

X. Glorot, A. Bordes, and Y. Bengio. Deep sparse rectifier neural networks. In *International Conference on Artificial Intelligence and Statistics*, 2011.

P. W. Glynn. Likelihood ratio gradient estimation for stochastic systems. *Commun. ACM*, 1990.

I. Goodfellow, A. Courville, and Y. Bengio. *Deep learning*. MIT Press, 2016.

O. Goudet, D. Kalainathan, P. Caillou, D. Lopez-Paz, I. Guyon, and M. Sebag. Learning Functional Causal Models with Generative Neural Networks. In *Explainable and Interpretable Models in Computer Vision and Machine Learning*. Springer International Publishing, 2018.

A. Goyal and Y. Bengio. Inductive biases for deep learning of higher-level cognition. *arXiv preprint arXiv:2011.15091*, 2021.

A. Goyal, A. R. Didolkar, N. R. Ke, C. Blundell, P. Beaudoin, N. Heess, M. C. Mozer, and Y. Bengio. Neural production systems. In *Advances in Neural Information Processing Systems*, 2021a.

A. Goyal, A. Lamb, J. Hoffmann, S. Sodhani, S. Levine, Y. Bengio, and B. Schölkopf. Recurrent independent mechanisms. In *International Conference on Learning Representations*, 2021b.

K. Greff, A. Rasmus, M. Berglund, T. Hao, H. Valpola, and J. Schmidhuber. Tagger: Deep unsupervised perceptual grouping. In *Advances in Neural Information Processing Systems*, 2016.

K. Greff, S. van Steenkiste, and J. Schmidhuber. Neural expectation maximization. In *Advances in Neural Information Processing Systems*, 2017.

K. Greff, R. L. Kaufman, R. Kabra, N. Watters, C. Burgess, D. Zoran, L. Matthey, M. Botvinick, and A. Lerchner. Multi-object representation learning with iterative variational inference. In *International Conference on Machine Learning*, 2019.

K. Greff, S. van Steenkiste, and J. Schmidhuber. On the binding problem in artificial neural networks. *arXiv preprint arXiv:2012.05208*, 2020.

L. Gresele. *Learning Identifiable Representations: Independent Influences and Multiple Views*. Phd thesis, Universität Tübingen, 2023.

L. Gresele, P. K. Rubenstein, A. Mehrjou, F. Locatello, and B. Schölkopf. The incomplete rosetta stone problem: Identifiability results for multi-view nonlinear ica. In *Conference on Uncertainty in Artificial Intelligence Conference*, 2020.

L. Gresele, J. V. Kügelgen, V. Stimper, B. Schölkopf, and M. Besserve. Independent mechanism analysis, a new concept? In *Advances in Neural Information Processing Systems*, 2021.

A. Gretton, K. Fukumizu, C. Teo, L. Song, B. Schölkopf, and A. Smola. A kernel statistical test of independence. In *Advances in Neural Information Processing Systems*, 2007.

R. Gribonval and S. Lesage. A survey of sparse component analysis for blind source separation: principles, perspectives, and new challenges. In *European Symposium on Artificial Neural Networks*, 2006.

M. Guruswamy Sethuraman, R. Lopez, R. Mohan, F. Fekri, T. Biancalani, and J.-C. Hütter. NODAGS-Flow: Nonlinear cyclic causal structure learning. In *International Conference on Artificial Intelligence and Statistics*, 2023.

M. U. Gutmann and A. Hyvärinen. Noise-contrastive estimation of unnormalized statistical models, with applications to natural image statistics. *Journal of Machine Learning Research*, 2012.

H. Hälvä and A. Hyvärinen. Hidden markov nonlinear ica: Unsupervised learning from nonstationary time series. In *Conference on Uncertainty in Artificial Intelligence (UAI)*, 2020.

H. Hälvä, S. L. Corff, L. Lehéricy, J. So, Y. Zhu, E. Gassiat, and A. Hyvarinen. Disentangling identifiable features from noisy data with structured nonlinear ICA. In *Advances in Neural Information Processing Systems*, 2021.

S. Harnad. The symbol grounding problem. *Physica D: Nonlinear Phenomena*, 1990.

T. Hastie, R. Tibshirani, and J. Friedman. *The elements of statistical learning: data mining, inference and prediction*. Springer, 2009.

A. Hauser and P. Bühlmann. Characterization and greedy learning of interventional markov equivalence classes of directed acyclic graphs. *Journal of Machine Learning Research*, 2012.

C. Heinze-Deml, M. H. Maathuis, and N. Meinshausen. Causal structure learning. *Annual Review of Statistics and Its Application*, 2018a.

C. Heinze-Deml, J. Peters, and N. Meinshausen. Invariant causal prediction for nonlinear models. *Journal of Causal Inference*, 2018b.

I. Higgins, L. Matthey, A. Pal, C. P. Burgess, X. Glorot, M. Botvinick, S. Mohamed, and A. Lerchner. beta-vae: Learning basic visual concepts with a constrained variational framework. In *International Conference on Learning Representations*, 2017.

A. E. Hoerl and R. W. Kennard. Ridge regression: Biased estimation for nonorthogonal problems. *Technometrics*, 1970.

D. Horan, E. Richardson, and Y. Weiss. When is unsupervised disentanglement possible? In *Advances in Neural Information Processing Systems*, 2021a.

D. Horan, E. Richardson, and Y. Weiss. When is unsupervised disentanglement possible? In *Advances in Neural Information Processing Systems*, 2021b.

T. Hospedales, A. Antoniou, P. Micaelli, and A. Storkey. Meta-learning in neural networks: A survey. *IEEE transactions on pattern analysis and machine intelligence*, 2021.

C.-J. Hsieh, K.-W. Chang, C.-J. Lin, S. S. Keerthi, and S. Sundararajan. A dual coordinate descent method for large-scale linear SVM. In *International Conference on Machine Learning*, 2008.

J. Hu and K. Huang. Global identifiability of ℓ_1-based dictionary learning via matrix volume optimization. In *Advances in Neural Information Processing Systems*, 2023.

B. Huang, K. Zhang, Y. Lin, B. Schölkopf, and C. Glymour. Generalized score functions for causal discovery. In *ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, 2018a.

C.-W. Huang, D. Krueger, A. Lacoste, and A. Courville. Neural autoregressive flows. In *International Conference on Machine Learning*, 2018b.

D. Hume. *An Enquiry Concerning Human Understanding*. 1748.

D. R. Hunter, S. Wang, and T. P. Hettmansperger. Inference for mixtures of symmetric distributions. *The Annals of Statistics*, 2007.

A. Hyttinen, F. Eberhardt, and M. Järvisalo. Constraint-based causal discovery: Conflict resolution with answer set programming. In *Conference on Uncertainty in Artificial Intelligence*, 2014.

A. Hyvarinen and H. Morioka. Unsupervised feature extraction by time-contrastive learning and nonlinear ica. In *Advances in Neural Information Processing Systems*, 2016.

A. Hyvarinen and H. Morioka. Nonlinear ICA of Temporally Dependent Stationary Sources. In *International Conference on Artificial Intelligence and Statistics*, 2017.

A. Hyvärinen and P. Pajunen. Nonlinear independent component analysis: Existence and uniqueness results. *Neural Networks*, 1999.

A. Hyvärinen, J. Karhunen, and E. Oja. *Independent Component Analysis*. Wiley, 2001.

A. Hyvärinen, H. Sasaki, and R. E. Turner. Nonlinear ica using auxiliary variables and generalized contrastive learning. In *International Conference on Artificial Intelligence and Statistics*, 2019.

A. Hyvärinen, I. Khemakhem, and H. Morioka. Nonlinear independent component analysis for principled disentanglement in unsupervised deep learning. *Patterns*, 2023.

S. Ioffe and C. Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *International Conference on Machine Learning*, 2015.

T. Jaakkola, D. Sontag, A. Globerson, and M. Meila. Learning Bayesian Network Structure using LP Relaxations. In *International Conference on Artificial Intelligence and Statistics*, 2010.

A. Jaber, M. Kocaoglu, K. Shanmugam, and E. Bareinboim. Causal discovery from soft interventions with unknown targets: Characterization and learning. In *Advances in Neural Information Processing Systems*, 2020.

E. Jang, S. Gu, and B. Poole. Categorical reparameterization with gumbel-softmax. *International Conference on Machine Learning*, 2017.

Y. Jiang and B. Aragam. Learning nonparametric latent causal graphs with unknown interventions. In *Advances in Neural Information Processing Systems*, 2023.

J. Johnson, B. Hariharan, L. van der Maaten, L. Fei-Fei, C. L. Zitnick, and R. B. Girshick. Clevr: A diagnostic dataset for compositional language and elementary visual reasoning. *IEEE Conference on Computer Vision and Pattern Recognition*, 2016.

C. Jutten and J. Herault. Blind separation of sources, part 1: An adaptive algorithm based on neuromimetic architecture. *Signal Process.*, 1991.

D. Kalainathan and O. Goudet. Causal discovery toolbox: Uncover causal relationships in python. *arXiv preprint arXiv:1903.02278*, 2019.

D. Kalainathan, O. Goudet, I. Guyon, D. Lopez-Paz, and M. Sebag. SAM: Structural agnostic model, causal discovery and penalized adversarial learning. *arXiv preprint arXiv:1803.04929*, 2018.

T. Karaletsos, S. Belongie, and G. Rätsch. Bayesian representation learning with oracle constraints. In *International Conference on Learning Representations*, 2016.

N. R. Ke, O. Bilaniuk, A. Goyal, S. Bauer, H. Larochelle, C. Pal, and Y. Bengio. Learning neural causal models from unknown interventions. *arXiv preprint arXiv:1910.01075*, 2019.

N. R. Ke, A. R. Didolkar, S. Mittal, A. Goyal, G. Lajoie, S. Bauer, D. J. Rezende, M. C. Mozer, Y. Bengio, and C. Pal. Systematic evaluation of causal discovery in visual model based reinforcement learning. *arXiv preprint arXiv:2107.00848*, 2021.

N. R. Ke, S. Chiappa, J. X. Wang, J. Bornschein, A. Goyal, M. Rey, T. Weber, M. Botvinick, M. C. Mozer, and D. J. Rezende. Learning to induce causal structure. In *International Conference on Learning Representations*, 2023.

H. Keurti, H.-R. Pan, M. Besserve, B. F. Grewe, and B. Schölkopf. Homomorphism autoencoder – learning group structured representations from observed transitions. In *International Conference on Machine Learning*, 2023.

I. Khemakhem, D. Kingma, R. Monti, and A. Hyvarinen. Variational autoencoders and nonlinear ica: A unifying framework. In *International Conference on Artificial Intelligence and Statistics*, 2020a.

I. Khemakhem, R. Monti, D. Kingma, and A. Hyvarinen. Ice-beem: Identifiable conditional energy-based deep models based on nonlinear ica. In *Advances in Neural Information Processing Systems*, 2020b.

H. Kim and A. Mnih. Disentangling by factorising. In *International Conference on Machine Learning*, 2018.

D. P. Kingma and J. Ba. Adam: A method for stochastic optimization. In *International Conference on Learning Representations*, 2015.

D. P. Kingma and M. Welling. Auto-encoding variational bayes. In *International Conference on Learning Representations*, 2014.

B. Kivva, G. Rajendran, P. K. Ravikumar, and B. Aragam. Identifiability of deep generative models without auxiliary information. In *Advances in Neural Information Processing Systems*, 2022.

D. A. Klindt, L. Schott, Y. Sharma, I. Ustyuzhaninov, W. Brendel, M. Bethge, and D. M. Paiton. Towards nonlinear disentanglement in natural data with temporal sparse coding. In *International Conference on Learning Representations*, 2021.

M. Kocaoglu, C. Snyder, A. G. Dimakis, and S. Vishwanath. CausalGAN: Learning causal implicit generative models with adversarial training. In *International Conference on Learning Representations*, 2018.

M. Kocaoglu, A. Jaber, K. Shanmugam, and E. Bareinboim. Characterization and learning of causal graphs with latent variables from soft interventions. In *Advances in Neural Information Processing Systems*. 2019.

D. Koller and N. Friedman. *Probabilistic Graphical Models: Principles and Techniques - Adaptive Computation and Machine Learning*. MIT Press, 2009.

K. B. Korb, L. R. Hope, A. E. Nicholson, and K. Axnick. Varieties of causal intervention. In *Pacific Rim International Conference on Artificial Intelligence*, 2004.

K. Kreutz-Delgado, J. F. Murray, B. D. Rao, K. Engan, T.-W. Lee, and T. J. Sejnowski. Dictionary learning algorithms for sparse representation. *Neural computation*, 2003.

A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in Neural Information Processing Systems*, 2012.

D. Krueger, E. Caballero, J.-H. Jacobsen, A. Zhang, J. Binas, R. L. Priol, D. Zhang, and A. Courville. Out-of-distribution generalization via risk extrapolation ({re}x), 2021a.

D. Krueger, E. Caballero, J.-H. Jacobsen, A. Zhang, J. Binas, D. Zhang, R. Le Priol, and A. Courville. Out-of-distribution generalization via risk extrapolation (rex). In *International Conference on Machine Learning*, 2021b.

A. Kumar, P. Sattigeri, and A. Balakrishnan. Variational inference of disentangled latent concepts from unlabeled observations. In *International Conference on Learning Representations*, 2018.

N. Kühl, M. Goutier, L. Baier, C. Wolff, and D. Martin. Human vs. supervised machine learning: Who learns patterns faster? *Cognitive Systems Research*, 2022.

S. Lachapelle and S. Lacoste-Julien. Partial disentanglement via mechanism sparsity. In *UAI Workshop on Causal Representation Learning*, 2022.

S. Lachapelle, P. Brouillard, T. Deleu, and S. Lacoste-Julien. Gradient-based neural DAG learning. In *International Conference on Learning Representations*, 2020.

S. Lachapelle, R. P., Y. Sharma, K. E. Everett, R. Le Priol, A. Lacoste, and S. Lacoste-Julien. Disentanglement via mechanism sparsity regularization: A new principle for nonlinear ICA. In *Conference on Causal Learning and Reasoning*, 2022.

S. Lachapelle, T. Deleu, D. Mahajan, I. Mitliagkas, Y. Bengio, S. Lacoste-Julien, and Q. Bertrand. Synergies between disentanglement and sparsity: Generalization and identifiability in multi-task learning. In *International Conference on Machine Learning*, 2023a.

S. Lachapelle, D. Mahajan, I. Mitliagkas, and S. Lacoste-Julien. Additive decoders for latent variables identification and cartesian-product extrapolation. In *Advances in Neural Information Processing Systems*, 2023b.

S. Lacoste–Julien, F. Huszár, and Z. Ghahramani. Approximate inference for the loss-calibrated bayesian. In *International Conference on Artificial Intelligence and Statistics*, 2011.

B. M. Lake, T. D. Ullman, J. B. Tenenbaum, and S. J. Gershman. Building machines that learn and think like people. *Behavioral and Brain Sciences*, 2017.

S. L. Lauritzen. *Graphical Models*. Oxford University Press, 1996.

R. Le Priol, R. Babanezhad, Y. Bengio, and S. Lacoste-Julien. An analysis of the adaptation speed of causal models. In *International Conference on Artificial Intelligence and Statistics*, 2021.

H.-C. Lee, M. Danieletto, R. Miotto, S. T. Cherng, and J. T. Dudley. *Scaling structural learning with NO-BEARS to infer causal transcriptome networks*. 2020.

K. Lee, S.Maji, A. Ravichandran, and S. Soatto. Meta-learning with differentiable convex optimization. In *IEEE/CVF conference on computer vision and pattern recognition*, 2019.

F. Leeb, G. Lanzillotta, Y. Annadani, M. Besserve, S. Bauer, and B. Schölkopf. Structure by architecture: Disentangled representations without regularization. *arXiv preprint arXiv:2006.07796*, 2021.

T. Leemann, M. Kirchhof, Y. Rong, E. Kasneci, and G. Kasneci. When are post-hoc conceptual explanations identifiable? *Conference on Uncertainty in Artificial Intelligence*, 2023.

A. Lei, B. Schölkopf, and I. Posner. Variational causal dynamics: Discovering modular world models from interventions. *Transactions on Machine Learning Research*, 2023.

D. Li, Y. Yang, Y. Song, and T. M. Hospedales. Deeper, broader and artier domain generalization. In *IEEE International Conference on Computer Vision*, 2017.

H. Li, Q. Xiao, and J. Tian. Supervised whole dag causal discovery. *arXiv preprint arXiv:2006.04697*, 2020.

W. Liang, A. Kekić, J. von Kügelgen, S. Buchholz, M. Besserve, L. Gresele, and B. Schölkopf. Causal component analysis. In *Advances in Neural Information Processing Systems*, 2023.

Z. Lin, Y. Wu, S. V. Peri, W. Sun, G. Singh, F. Deng, J. Jiang, and S. Ahn. Space: Unsupervised object-oriented scene representation via spatial attention and decomposition. In *International Conference on Learning Representations*, 2020.

P. Lippe, S. Magliacane, S. Löwe, Y. M. Asano, T. Cohen, and E. Gavves. CITRIS: Causal identifiability from temporal intervened sequences. *arXiv preprint arXiv:2202.03169*, 2022.

P. Lippe, S. Magliacane, S. Löwe, Y. M. Asano, T. Cohen, and E. Gavves. BISCUIT: Causal representation learning from binary interactions. In *Conference on Uncertainty in Artificial Intelligence*, 2023a.

P. Lippe, S. Magliacane, S. Löwe, Y. M. Asano, T. Cohen, and E. Gavves. iCITRIS: Causal representation learning for instantaneous temporal effects. In *International Conference on Learning Representations*, 2023b.

Y. Liu, Z. Zhang, D. Gong, M. Gong, B. Huang, A. van den Hengel, K. Zhang, and J. Q. Shi. Identifying weight-variant latent causal models. *arXiv preprint arXiv:2208.14153*, 2023.

F. Locatello, S. Bauer, M. Lucic, G. Raetsch, S. Gelly, B. Schölkopf, and O. Bachem. Challenging common assumptions in the unsupervised learning of disentangled representations. In *International Conference on Machine Learning*, 2019.

F. Locatello, B. Poole, G. Raetsch, B. Schölkopf, O. Bachem, and M. Tschannen. Weakly-supervised disentanglement without compromises. In *International Conference on Machine Learning*, 2020a.

F. Locatello, M. Tschannen, S. Bauer, G. Rätsch, B. Schölkopf, and O. Bachem. Disentangling factors of variations using few labels. In *International Conference on Learning Representations*, 2020b.

F. Locatello, D. Weissenborn, T. Unterthiner, A. Mahendran, G. Heigold, J. Uszkoreit, A. Dosovitskiy, and T. Kipf. Object-centric learning with slot attention. In *Advances in Neural Information Processing Systems*, 2020c.

R. Lopez, J.-C. Huetter, J. Pritchard, and A. Regev. Large-scale differentiable causal discovery of factor graphs. In *Advances in Neural Information Processing Systems*, 2022.

R. Lopez, N. Tagasovska, S. Ra, K. Cho, J. K. Pritchard, and A. Regev. Learning causal representations of single cells via sparse mechanism shift modeling. *Conference on Causal Learning and*

Reasoning, 2023.

D. Lopez-Paz, K. Muandet, B. Schölkopf, and I. Tolstikhin. Towards a learning theory of cause-effect inference. In *International Conference on Machine Learning*, 2015.

L. Lorch, J. Rothfuss, B. Schölkopf, and A. Krause. DiBS: Differentiable bayesian structure learning. In *Advances in Neural Information Processing Systems*, 2021.

K. Lounici, M. Pontil, and A. B. Tsybakov. Oracle inequalities and optimal inference under group sparsity. *The Annals of statistics*, 2011a.

K. Lounici, M. Pontil, S. Van De Geer, and A. B. Tsybakov. Oracle inequalities and optimal inference under group sparsity. *The Annals of Statistics*, 2011b.

C. Lu, Y. Wu, J. M. Hernández-Lobato, and B. Schölkopf. Nonlinear invariant risk minimization: A causal approach. *arXiv preprint arXiv:2102.12353*, 2021.

K. Madan, N. R. Ke, A. Goyal, B. Schölkopf, and Y. Bengio. Fast and slow learning of recurrent independent mechanisms. In *International Conference on Learning Representations*, 2021.

C. J. Maddison, A. Mnih, and Y. W. Teh. The concrete distribution: A continuous relaxation of discrete random variables. *International Conference on Machine Learning*, 2017.

S. Magliacane, T. van Ommen, T. Claassen, S. Bongers, P. Versteeg, and J. Mooij. Domain adaptation by using causal inference to predict invariant conditional distributions. In *Advances in Neural Information Processing Systems*, 2018.

J. Mairal, J. Ponce, G. Sapiro, A. Zisserman, and F. Bach. Supervised dictionary learning. *Advances in neural information processing systems*, 2008.

J. Mairal, F. Bach, J. Ponce, and G. Sapiro. Online dictionary learning for sparse coding. In *International Conference on Machine Learning*, 2009.

J. Mairal, F. Bach, and J. Ponce. Task-driven dictionary learning. *IEEE transactions on pattern analysis and machine intelligence*, 2011.

B. Malézieux, T. Moreau, and M. Kowalski. Dictionary and prior learning with unrolled algorithms for unsupervised inverse problems. *International Conference on Learning Representations*, 2022.

A. Mansouri, J. Hartford, K. Ahuja, and Y. Bengio. Object-centric causal representation learning. In *NeurIPS Workshop on Symmetry and Geometry in Neural Representations*, 2022.

G. Marcus, E. Davis, and S. Aaronson. A very preliminary analysis of dall-e 2. *arXiv preprint arXiv:2204.13807*, 2022.

G. F. Marcus. The algebraic mind : integrating connectionism and cognitive science, 2001.

A. Maurer, M. Pontil, and B. Romera-Paredes. Sparse coding for multitask and transfer learning. International Conference on Machine Learning, 2013.

A. Maurer, M. Pontil, and B. Romera-Paredes. The benefit of multitask representation learning. *Journal of Machine Learning Research*, 2016.

T. Mikolov, M. Karafiát, L. Burget, J. Cernocký, and S. Khudanpur. Recurrent neural network based language model. ISCA, 2010.

D. Miladinović, M. W. Gondal, B. Schölkopf, J. M. Buhmann, and S. Bauer. Disentangled state space representations. *arXiv preprint arXiv:1906.03255*, 2019.

M. Mohri, A. Rostamizadeh, and A. Talwalkar. MIT Press, 2018.

M. L. Montero, C. J. Ludwig, R. P. Costa, G. Malhotra, and J. Bowers. The role of disentanglement in generalisation. In *International Conference on Learning Representations*, 2021.

R. P. Monti and A. Hyvärinen. A unified probabilistic model for learning latent factors and their connectivities from high-dimensional data. *Conference on Uncertainty in Artificial Intelligence*, 2018.

J. M. Mooij, S. Magliacane, and T. Claassen. Joint causal inference from multiple contexts. *arXiv preprint arXiv:1611.10351*, 2016.

J. M. Mooij, S. Magliacane, and T. Claassen. Joint causal inference from multiple contexts. *Journal of Machine Learning Research*, 2020.

G. E. Moran, D. Sridhar, Y. Wang, and D. Blei. Identifiable deep generative models via sparse decoding. *Transactions on Machine Learning Research*, 2022.

H. Morioka and A. Hyvarinen. Connectivity-contrastive learning: Combining causal discovery and representation learning for multimodal data. In *International Conference on Artificial Intelligence and Statistics*, 2023.

H. Morioka, H. Hälvä, and A. Hyvärinen. Independent innovation analysis for nonlinear vector autoregressive process. In *International Conference on Artificial Intelligence and Statistics*, 2021.

D. Morwani, B. L. Edelman, C.-A. Oncescu, R. Zhao, and S. M. Kakade. Feature emergence via margin maximization: case studies in algebraic tasks. In *International Conference on Learning Representations*, 2024.

J. Munkres. *Analysis On Manifolds*. Basic Books, 1991.

J. R. Munkres. *Topology*. Prentice Hall, Inc., 2 edition, 2000.

K. P. Murphy. *Probabilistic Machine Learning: Advanced Topics*. MIT Press, 2023.

S. Nair, Y. Zhu, S. Savarese, and L. Fei-Fei. Causal induction from visual observations for goal directed tasks. *arXiv preprint arXiv:1910.01751*, 2019.

A. Nazaret, J. Hong, E. Azizi, and D. Blei. Stable differentiable causal discovery. *arXiv preprint arXiv:2311.10263*, 2023.

I. Ng, Z. Fang, S. Zhu, Z. Chen, and J. Wang. Masked gradient-based causal structure learning. *arXiv preprint arXiv:1910.08527*, 2019.

I. Ng, S. Lachapelle, N. R. Ke, S. Lacoste-Julien, and K. Zhang. On the convergence of continuous constrained optimization for structure learning. In *International Conference on Artificial Intelligence and Statistics*, 2022.

M. Nishikawa-Toomey, T. Deleu, J. Subramanian, Y. Bengio, and L. Charlin. Bayesian learning of causal structure and mechanisms with gflownets and variational bayes. *arXiv preprint arXiv:2211.02763*, 2023.

C. Olah, N. Cammarata, L. Schubert, G. Goh, M. Petrov, and S. Carter. Zoom in: An introduction to circuits. *Distill*, 2020. https://distill.pub/2020/circuits/zoom-in.

A. Oord, Y. Li, and O. Vinyals. Representation learning with contrastive predictive coding. *Advances in Neural Information Processing Systems*, 2018.

F. R. M. Pavan and M. D. Miranda. On the darmois-skitovich theorem and spatial independence in blind source separation. *Journal of Communication and Information Systems*, 2018.

J. Pearl. A constraint propagation approach to probabilistic reasoning. *Conference on Uncertainty in Artificial Intelligence*, 1985.

J. Pearl. *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*. Morgan Kaufmann Publishers Inc., 1988.

J. Pearl. *Causality: Models, Reasoning and Inference*. Cambridge University Press, 2nd edition, 2009a.

J. Pearl. *Causality*. Cambridge university press, 2009b.

J. Pearl. The seven tools of causal inference, with reflections on machine learning. *Commun. ACM*, 2019.

F. Pedregosa. Hyperparameter optimization with approximate gradient. In *International conference on machine learning*, 2016.

W. Peebles, J. Peebles, J.-Y. Zhu, A. A. Efros, and A. Torralba. The hessian penalty: A weak prior for unsupervised disentanglement. In *European Conference on Computer Vision*, 2020.

J. Peters and P. Bühlman. Identifiability of Gaussian structural equation models with equal error variances. *Biometrika*, 2014.

J. Peters and P. Bühlmann. Structural intervention distance (SID) for evaluating causal graphs. *Neural Computation*, 2015.

J. Peters, D. Janzing, and B. Schölkopf. Causal inference on discrete data using additive noise models. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2011.

J. Peters, J. M. Mooij, D. Janzing, and B. Schölkopf. Causal discovery with continuous additive noise models. *Journal of Machine Learning Research*, 2014.

J. Peters, P. Bühlmann, and N. Meinshausen. Causal inference by using invariant prediction: identification and confidence intervals. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 2016.

J. Peters, D. Janzing, and B. Schölkopf. *Elements of Causal Inference - Foundations and Learning Algorithms*. MIT Press, 2017.

N. Pfister, P. Bühlmann, and J. Peters. Invariant causal prediction for sequential data. *Journal of the American Statistical Association*, 2019.

T. A. Poggio, K. Kawaguchi, Q. Liao, B. Miranda, L. Rosasco, X. Boix, J. Hidary, and H. Mhaskar. Theory of deep learning III: explaining the non-overfitting puzzle. *arXiv preprint arXiv:1801.00173*, 2018.

D. Pollard. *A User's Guide to Measure Theoretic Probability*. Cambridge University Press, 2001.

A. Power, Y. Burda, H. Edwards, I. Babuschkin, and V. Misra. Grokking: Generalization beyond overfitting on small algorithmic datasets. *arXiv preprint arXiv:2201.02177*, 2022.

L. Prechelt. Early stopping - but when? In *Neural Networks: Tricks of the Trade, volume 1524 of LNCS, chapter 2*, 1997.

A. Radford, J. W. Kim, C. Hallacy, A. Ramesh, G. Goh, S. Agarwal, G. Sastry, A. Askell, P. Mishkin, J. Clark, et al. Learning transferable visual models from natural language supervision. In *International Conference on Machine Learning*, 2021.

A. Ramesh, P. Dhariwal, A. Nichol, C. Chu, and M. Chen. Hierarchical text-conditional image generation with clip latents. *arXiv preprint arXiv:2204.06125*, 2022.

J. Ramsey, M. Glymour, R. Sanchez-Romero, and C. Glymour. A million variables and more: the fast greedy equivalence search algorithm for learning high-dimensional graphical causal models, with an application to functional magnetic resonance images. *I. J. Data Science and Analytics*, 2017.

P. Reizinger, L. Gresele, J. Brady, J. V. Kügelgen, D. Zietlow, B. Schölkopf, G. Martius, W. Brendel, and M. Besserve. Embrace the gap: VAEs perform independent mechanism analysis. In *Advances in Neural Information Processing Systems*, 2022.

D. J. Rezende and S. Mohamed. Variational inference with normalizing flows. *International Conference on Machine Learning*, 2015.

D. J. Rezende, S. Mohamed, and D. Wierstra. Stochastic backpropagation and approximate inference in deep generative models. In *International Conference on Machine Learning*, 2014.

P. Richtárik and M. Takáč. Iteration complexity of randomized block-coordinate descent methods for minimizing a composite function. *Mathematical Programming*, 2014.

G. Roeder, L. Metz, and D. P. Kingma. On linear identifiability of learned representations. In *International Conference on Machine Learning*, 2021.

R. Rombach, A. Blattmann, D. Lorenz, P. Esser, and B. Ommer. High-resolution image synthesis with latent diffusion models. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022.

S. Ross. *A First Course in Probability*. Pearson Prentice Hall, 2010.

K. Roth, M. Ibrahim, Z. Akata, P. Vincent, and D. Bouchacourt. Disentanglement of correlated factors via hausdorff factorized support. In *International Conference on Learning Representations*,

2023.

K. Sachs, O. Perez, D. Pe'er, D. Lauffenburger, and G. Nolan. Causal protein-signaling networks derived from multiparameter single-cell data. *Science*, 2005.

S. Sagawa, P. W. Koh, T. B. Hashimoto, and P. Liang. Distributionally robust neural networks. In *International Conference on Learning Representations*, 2020.

W. C. Salmon. The uniformity of nature. *Philosophy and Phenomenological Research*, 1953.

A. Schell and H. Oberhauser. Nonlinear independent component analysis for discrete-time and continuous-time signals. *The Annals of Statistics*, 2023.

N. Scherrer, O. Bilaniuk, Y. Annadani, A. Goyal, P. Schwab, B. Schölkopf, M. C. Mozer, Y. Bengio, S. Bauer, and N. R. Ke. Learning neural causal models with active interventions. *arXiv preprint arXiv:2109.02429*, 2022.

B. Schölkopf. Causality for machine learning. *arXiv preprint arXiv:1911.10500*, 2019.

B. Schölkopf, F. Locatello, S. Bauer, N. R. Ke, N. Kalchbrenner, A. Goyal, and Y. Bengio. Toward causal representation learning. *IEEE - Advances in Machine Learning and Deep Neural Networks*, 2021.

S. Shalev-Shwartz and S. Ben-David. *Understanding Machine Learning: From Theory to Algorithms*. Cambridge University Press, 2014.

X. Shen, F. Liu, H. Dong, Q. Lian, Z. Chen, and T. Zhang. Weakly supervised disentangled generative causal representation learning. *Journal of Machine Learning Research*, 2022.

S. Shimizu, P. Hoyer, A. Hyvärinen, and A. Kerminen. A linear non-gaussian acyclic model for causal discovery. *Journal of Machine Learning Research*, 2006.

V. P. Skitivic. On a property of the normal distribution. *Izvestiya Akademii Nauk SSSR. Seriya Matematicheskaya*, 1953.

J. Snell, K. Swersky, and R. Zemel. Prototypical networks for few-shot learning. *Advances in Neural Information Processing Systems*, 2017.

P. Sorrenson, C. Rother, and U. Köthe. Disentanglement by nonlinear ica with general incompressible-flow networks (GIN). In *International Conference on Learning Representations*, 2020.

P. Spirtes, C. N. Glymour, R. Scheines, and D. Heckerman. *Causation, prediction, and search*. 2000.

C. Squires, Y. Wang, and C. Uhler. Permutation-based causal structure learning with unknown intervention targets. *Conference on Uncertainty in Artificial Intelligence*, 2020.

C. Squires, A. Seigal, S. Bhate, and C. Uhler. Linear causal disentanglement via interventions. In *International Conference on Machine Learning*, 2023.

E. V. Strobl, K. Zhang, and S. Visweswaran. Approximate kernel-based conditional independence tests for fast non-parametric causal discovery. *Journal of Causal Inference*, 2019.

C. Szegedy, W. Zaremba, I. Sutskever, J. Bruna, D. Erhan, I. J. Goodfellow, and R. Fergus. Intriguing properties of neural networks. In *International Conference on Learning Representations*, 2014.

A. Taleb and C. Jutten. Source separation in post-nonlinear mixtures. *IEEE Transactions on Signal Processing*, 1999.

H. Teicher. Identifiability of Finite Mixtures. *The Annals of Mathematical Statistics*, 1963.

H. Teicher. Identifiability of mixtures of product measures. *The Annals of Mathematical Statistics*, 1967.

J. B. Tenenbaum, C. Kemp, T. L. Griffiths, and N. D. Goodman. How to grow a mind: Statistics, structure, and abstraction. *Science*, 2011.

V. Thomas, J. Pondard, E. Bengio, M. Sarfati, P. Beaudoin, M.-J. Meurs, J. Pineau, D. Precup, and Y. Bengio. Independently controllable factors. *arXiv preprint arXiv:1708.01289*, 2017.

R. Tibshirani. Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society: Series B (Methodological)*, 58(1):267–288, 1996.

T. Tieleman and G. Hinton. Lecture 6.5—RmsProp: Divide the gradient by a running average of its recent magnitude. COURSERA: Neural Networks for Machine Learning, 2012.

L. Tong, V. Soon, Y. Huang, and R. Liu. AMUSE: a new blind identification algorithm. In *IEEE International Symposium on Circuits and Systems*, 1990.

L. Tong, Y. Inouye, and R.-w. Liu. Waveform-preserving blind estimation of multiple independent sources. *IEEE Transactions on Signal Processing*, 1993.

C. Toth, L. Lorch, C. Knoll, A. Krause, F. Pernkopf, R. Peharz, and J. von Kügelgen. Active bayesian causal inference. In *Advances in Neural Information Processing Systems*, 2022.

F. Träuble, E. Creager, N. Kilbertus, F. Locatello, A. Dittadi, A. Goyal, B. Schölkopf, and S. Bauer. On disentangled representations learned from correlated data. In *International Conference on Machine Learning*, 2021.

S. Triantafillou and I. Tsamardinos. Constraint-based causal discovery from multiple interventions over overlapping variable sets. *Journal of Machine Learning Research*, 2015.

P. Tseng. Convergence of a block coordinate descent method for nondifferentiable minimization. *Journal of optimization theory and applications*, 2001.

T. Van den Bulcke, et al. SynTReN: a generator of synthetic gene expression data for design and analysis of structure learning algorithms. 2006.

S. van Steenkiste, F. Locatello, J. Schmidhuber, and O. Bachem. Are disentangled representations helpful for abstract visual reasoning? In *Advances in Neural Information Processing Systems*, 2019.

B. Varici, E. Acarturk, K. Shanmugam, A. Kumar, and A. Tajer. Score-based causal representation learning from interventions: Nonparametric identifiability. In *Causal Representation Learning Workshop at NeurIPS 2023*, 2023a.

B. Varici, E. Acartürk, K. Shanmugam, and A. Tajer. General identifiability and achievability for causal representation learning. *arXiv preprint arXiv:2310.15450*, 2023b.

H. Venkateswara, J. Eusebio, S. Chakraborty, and S. Panchanathan. Deep hashing network for unsupervised domain adaptation. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2017.

T. Verma and J. Pearl. Equivalence and synthesis of causal models. In *Conference on Uncertainty in Artificial Intelligence*, 1990.

O. Vinyals, C. Blundell, T. Lillicrap, D. Wierstra, et al. Matching networks for one shot learning. *Advances in Neural Information Processing Systems*, 2016.

S. Volodin. CauseOccam : Learning interpretable abstract representations in reinforcement learning environments via model sparsity. Master's thesis, École Polytechnique Fédérale de Lausanne, 2021.

J. Von Kügelgen, Y. Sharma, L. Gresele, W. Brendel, B. Schölkopf, M. Besserve, and F. Locatello. Self-supervised learning with data augmentations provably isolates content from style. In *Advances in Neural Information Processing Systems*, 2021.

J. von Kügelgen, M. Besserve, W. Liang, L. Gresele, A. Kekić, E. Bareinboim, D. Blei, and B. Schölkopf. Nonparametric identifiability of causal representations from unknown interventions. In *Advances in Neural Information Processing Systems*, 2023.

M. J. Wainwright and M. I. Jordan. Graphical models, exponential families, and variational inference. *Found. Trends Mach. Learn.*, 2008.

Y. Wang and M. I. Jordan. Desiderata for representation learning: A causal perspective, 2022.

Y. Wang, L. Solus, K. Yang, and C. Uhler. Permutation-based causal inference algorithms with interventions. In *Advances in Neural Information Processing Systems*, 2017.

Z. Wang, L. Gui, J. Negrea, and V. Veitch. Concept algebra for text-controlled vision models. *arXiv preprint arXiv:2302.03693*, 2023.

L. Wasserman. *All of statistics : a concise course in statistical inference*. Springer, 2010.

T. W. Webb, Z. Dulberg, S. M. Frankland, A. A. Petrov, R. C. O'Reilly, and J. D. Cohen. Learning representations that support extrapolation. In *International Conference on Machine Learning*, 2020.

J. S. Weinstock, M. M. Arce, J. W. Freimer, M. Ota, A. Marson, A. Battle, and J. K. Pritchard. Gene regulatory network inference from crispr perturbations in primary cd4+ t cells elucidates the genomic basis of immune disease. *bioRxiv*, 2023.

T. Wiedemer, P. Mayilvahanan, M. Bethge, and W. Brendel. Compositional generalization from first principles. In *Advances in Neural Information Processing Systems*, 2023.

T. Wiedemer, J. Brady, A. Panfilov, A. Juhos, M. Bethge, and W. Brendel. Provable compositional generalization for object-centric learning. In *International Conference on Learning*

Representations, 2024.

R. J. Williams. Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Mach. Learn.*, 1992.

M. Wortsman, G. Ilharco, J. W. Kim, M. Li, S. Kornblith, R. Roelofs, R. G. Lopes, H. Hajishirzi, A. Farhadi, H. Namkoong, and L. Schmidt. Robust fine-tuning of zero-shot models. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022.

S. Wright and J. Nocedal. Numerical optimization. *Springer Science*, 1999.

M. Wu, Y. Bao, R. Barzilay, and T. Jaakkola. Sample, estimate, aggregate: A recipe for causal discovery foundation models. *arXiv preprint arXiv:2402.01929*, 2024.

Q. Xi and B. Bloem-Reddy. Indeterminacy in generative models: Characterization and strong identifiability. In *International Conference on Artificial Intelligence and Statistics*, 2023.

D. Xu, D. Yao, S. Lachapelle, P. Taslakian, J. von Kügelgen, F. Locatello, and S. Magliacane. A sparsity principle for partially observable causal representation learning. In *Causal Representation Learning Workshop at NeurIPS 2023*, 2023.

K. D. Yang, A. Katcoff, and C. Uhler. Characterizing and learning equivalence classes of causal DAGs under interventions. *International Conference on Machine Learning*, 2018.

M. Yang, F. Liu, Z. Chen, X. Shen, J. Hao, and J. Wang. CausalVAE: Disentangled representation learning via neural structural causal models. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021.

D. Yao, D. Xu, S. Lachapelle, S. Magliacane, P. Taslakian, G. Martius, J. von Kügelgen, and F. Locatello. Multi-view causal representation learning with partial observability. *arXiv preprint arXiv:2311.04056*, 2023.

W. Yao, G. Chen, and K. Zhang. Temporally disentangled representation learning. In *Advances in Neural Information Processing Systems*, 2022a.

W. Yao, Y. Sun, A. Ho, C. Sun, and K. Zhang. Learning temporally causal latent processes from general temporal data. In *International Conference on Learning Representations*, 2022b.

W. Yao, Y. Sun, A. Ho, C. Sun, and K. Zhang. Learning temporally causal latent processes from general temporal data. In *International Conference on Learning Representations*, 2022c.

Y. Yu, J. Chen, T. Gao, and M. Yu. DAG-GNN: DAG structure learning with graph neural networks. In *International Conference on Machine Learning*, 2019a.

Y. Yu, J. Chen, T. Gao, and M. Yu. DAG-GNN: DAG structure learning with graph neural networks. In *International Conference on Machine Learning*, 2019b.

C. Zhang, S. Bengio, M. Hardt, B. Recht, and O. Vinyals. Understanding deep learning requires rethinking generalization. In *International Conference on Learning Representations*, 2017.

H. Zhang, Y.-F. Zhang, W. Liu, A. Weller, B. Schölkopf, and E. Xing. Towards principled disentanglement for domain generalization. In *IEEE/CVF Conference on Computer Vision and*

Pattern Recognition, 2022.

J. Zhang, K. Greenewald, C. Squires, A. Srivastava, K. Shanmugam, and C. Uhler. Identifiability guarantees for causal disentanglement from soft interventions. In *Advances in Neural Information Processing Systems*, 2023.

K. Zhang and A. Hyvärinen. On the identifiability of the post-nonlinear causal model. In *Conference on Uncertainty in Artificial Intelligence*, 2009.

K. Zhang, J. Peters, D. Janzing, and B. Schölkopf. Kernel-based conditional independence test and application in causal discovery. *Conference on Uncertainty in Artificial Intelligence*, 2011.

K. Zhang, Z. Wang, J. Zhang, and B. Schölkopf. On estimation of functional causal models: General results and application to the post-nonlinear causal model. *ACM Trans. Intell. Syst. Technol.*, 2015.

Q. Zhang, S. Filippi, A. Gretton, and D. Sejdinovic. Large-scale kernel methods for independence testing. *Statistics and Computing*, 2018.

X. Zheng, B. Aragam, P. Ravikumar, and E. Xing. DAGs with NO TEARS: Continuous optimization for structure learning. In *Advances in Neural Information Processing Systems*, 2018.

X. Zheng, C. Dan, B. Aragam, P. Ravikumar, and E. Xing. Learning sparse nonparametric dags. In *International Conference on Artificial Intelligence and Statistics*, 2020.

Y. Zheng, I. Ng, and K. Zhang. On the identifiability of nonlinear ICA: Sparsity and beyond. In *Advances in Neural Information Processing Systems*, 2022.

S. Zhu and Z. Chen. Causal discovery with reinforcement learning. *International Conference on Learning Representations*, 2020.

A. M. Zimmer, Y. K. Pan, T. Chandrapalan, R. W. Kwong, and S. F. Perry. Loss-of-function approaches in comparative physiology: is there a future for knockdown experiments in the era of genome editing? *Journal of Experimental Biology*, 2019.