Advances in Identifiability of Nonlinear Probabilistic Models

Ilyes Khemakhem

A dissertation submitted in partial fulfillment
of the requirements for the degree of

Doctor of Philosophy
of
University College London.

Gatsby Computational Neuroscience Unit
University College London

April 2022

I, Ilyes Khemakhem, declare that the work presented in this thesis is my own. Where information has been derived from other sources, I confirm that this has been acknowledged in the work.

Abstract

Identifiability is a highly prized property of statistical models. This thesis investigates this property in nonlinear models encountered in two fields of statistics: representation learning and causal discovery. In representation learning, identifiability leads to learning interpretable and reproducible representations, while in causal discovery, it is necessary for the estimation of correct causal directions.

We begin by leveraging recent advances in nonlinear ICA to show that the latent space of a VAE is identifiable up to a permutation and pointwise nonlinear transformations of its components. A factorized prior distribution over the latent variables conditioned on an auxiliary observed variable, such as a class label or nearly any other observation, is required for our result. We also extend previous identifiability results in nonlinear ICA to the case of noisy or undercomplete observations, and incorporate them into a maximum likelihood framework.

Our second contribution is to develop the Independently Modulated Component Analysis (IMCA) framework, a generalization of nonlinear ICA to non-independent latent variables. We show that we can drop the independence assumption in ICA while maintaining identifiability, resulting in a very flexible and generic framework for principled disentangled representation learning. This finding is predicated on the existence of an auxiliary variable that modulates the joint distribution of the latent variables in a factorizable manner.

As a third contribution, we extend the identifiability theory to a broad family of conditional energy-based models (EBMs). This novel model generalizes earlier results by removing any distributional assumptions on the representations, which are ubiquitous in the latent variable setting. The conditional EBM can learn identifiable overcomplete representations and has universal approximation

capabilities.

Finally, we investigate a connection between the framework of autoregressive normalizing flow models and causal discovery. Causal models derived from affine autoregressive flows are shown to be identifiable, generalizing the well-known additive noise model. Using normalizing flows, we can compute the exact likelihood of the causal model, which is subsequently used to derive a likelihood ratio measure for causal discovery. They are also invertible, making them perfectly suitable for performing causal inference tasks like interventions and counterfactuals.

Impact statement

Probabilistic models are a class of machine learning algorithms that define probability distributions across a wide range of mathematical objects, from simple scalars to multi-dimensional pictures and text sequences. Recent developments in deep neural networks have significantly improved the capabilities of such models and heightened interest in them. Within this class of models, expressing the observations as a complicated transformation of a lower-dimensional latent variable is becoming increasingly common. Subsequently, the latent code may be employed in tasks like image classification or compressed sensing.

Identifiability, which allows us to learn repeatable and interpretable representations, is a desired characteristic that such models lack. Identifiable representations are often more appropriate for the task at hand, and they have a substantial impact on learning performance and training robustness. Moreover, they can help state-of-the-art models perform even better in challenging applications like blind source separation, causal discovery, transfer learning, and zero-shot learning, to mention a few.

This thesis establishes theoretical identifiability guarantees for large families of deep probabilistic models which hold regardless of the type of data or the practical implementation. We believe this will pave the way for the widespread use of identifiable models in a variety of applications.

Acknowledgements

My deepest and sincerest gratitude goes to my supervisor, Aapo Hyvärinen. Thank you for your presence, advice, and unwavering support over the last four years. You introduced me to the wonderful realm of research, and I learnt a lot from you. You are, without a doubt, the smartest, funniest, and most humble academic I know. I consider myself blessed to have you as my adviser, and I couldn't have asked for greater mentorship. I also wish to thank Arthur Gretton, my secondary supervisor, for his support and expertise. Your insightful feedback helped sharpen my knowledge and have a better grasp on my research.

I gratefully acknowledge the funding received towards my PhD from the Gatsby Charitable Foundation. I'd want to thank the Gatsby Unit for creating such a stimulating working environment and for making afternoon tea such an important part of my experience in the UK. It has been nothing short of incredible. Special thanks to Ana, Barry, and Mike for their help in making my time at Gatsby as enjoyable and stress-free as possible. I would also like to express my deepest appreciation to my committee, Dr. Mark Herbster and Dr. Pascal Vincent, for agreeing to be my thesis examiners.

Throughout my PhD, I had the privilege of working with several excellent scholars and learnt a great deal from them. I'd like to thank Durk in particular for his great insight into variational autoencoders and energy-based models, as well as Luigi and Hermanni for the fruitful conversations regarding nonlinear ICA and identifiability, in addition to the nice memories from our brief stay in Paris. Most importantly, I want to thank Ricardo for believing in my ideas, being an outstanding mentor at Gatsby, and assisting me with the projects I've worked on during my PhD.

I would also like to say a heartfelt thank you to my family, particularly my

parents, without whom I would not be where I am now. You have provided me with every chance to succeed and have been the finest role models for me to follow. I hope I am unlike you in the smallest possible number of ways. Big big thanks to my younger brother for his continuous support, and to my little sister for being the best distraction I could ask for.

A very special thank you to all the lovely people I've met throughout my time in London. To Alex, brother and good friend, for making me feel right at home in London from the very start, and for being there for me through thick and thin. I love our countless debates on the most insignificant subjects. I know that I've made a friend for life, and I'll forever cherish the countless memories we've made together. Merci beaucoup ! To Philip, the third musketeer, and my most unpredictable and mysterious friend. You have greatly contributed in making my time in London, especially in the beginning, such a great experience. Thank you hajj! To Elena, friend and confidant. You are the kindest and most genuine person I've met in London. You've been selfless in listening to my complaining, and always offering the best advice. I won't forget all the delicious breakfasts we've shared, and thank you for teaching me how to shout at broken water taps. Stay awesome! To Pierre, I'm glad we became friends despite the short time we overlapped at Gatsby. Thank you for all the stimulating scientific discussions, the nights out, the good food and the great music. To Billy, for your boundless energy and for putting up with my ridiculous conversations about multi-humped camels. To Anna, for your invaluable guidance and advice, as well as for being so much fun after work. To Lea, Franziska, Jorge, Ricardo, Nate, Lillianne, Simon, Peter, Clémentine, Hugo, Michael, Roman, Will, Liang, Lucas, Rodrigo, Ted, Heishiro, and everyone else who helped make these four years so special. To my flatmate Aziz, for being solely responsible for the Kingshouse's world-class, five-star experience. Thank you for being a wonderful friend and for your invaluable advice and unwavering support. To the London squad: Omar, Mariem, Akrem, Wiem, Walid, Iyed, Soltan, and Senda, thank you for the fantastic food and for all of the memorable hours of fun and competition during our game nights. To Timothé, Oriane, Maxime, and Pierre-Yves, my fellow passengers on London's only Shinkansen, thank you for all the enjoyable trips and for being a wonderful second family through and since Covid's terrible times.

I am also indebted to all the friends I have made along the way and who

have positively affected my life in various ways. Sonia, Rami, Hela, and Jasser, my best friends, thank you for your unwavering support through good and bad times from the beginning of this journey many years ago. Moez, thank you for being the finest and kindest host every time I visit Paris, as well as a wonderful friend. Pierre, thank you for all the photography outings and for having shared your tent with me under the cold sky of La Courtine many winters ago. To the TSN crew: Théo, Julien, and Raymond, thank you for always being willing to try something new. Last but not least, I'd like to thank my lifelong pals Dhiaeddine, Haitham, Mahdi, Haroun, Mohamed, Amine, and Aymen for their constant support despite the distance that separates us.

Contents

Abstract 5

Impact statement 7

Acknowledgements 9

Contents 13

List of Figures 19

List of Tables 21

Chapter 1: Introduction 23

1.1 Mathematical preliminaries 23

1.2 Identifiability in representation learning 27

1.2.1 Disentangled representation learning 28

1.2.2 Independent component analysis 31

1.3 Identifiability in causal discovery 38

1.3.1 Structural equation models 39

1.3.2 Identifiable nonlinear causal models 41

1.3.3 Estimation methods for causal discovery 42

1.4 Contributions and structure of the thesis 45

1.4.1 Structure and contributions in brief 45

1.4.2 Detailed contributions 46

1.4.3 Publications 53

1.5 Notation and terminology 53

Appendices to Chapter 1 56

1.A Statistical independence 56

1.B Exponential family . . . 56
1.B.1 Conditional exponential family . . . 58
1.B.2 Exponential family and independence . . . 59

Chapter 2: Variational autoencoders and nonlinear ICA 61
2.1 Introduction . . . 62
2.2 Unidentifiability of deep latent variable models . . . 64
2.3 An identifiable model based on conditionally factorial priors . . . 66
2.3.1 Definition of proposed model . . . 66
2.3.2 Estimation by VAE . . . 67
2.3.3 Identifiability and consistency results . . . 69
2.3.4 Interpretation as nonlinear ICA . . . 69
2.3.5 Relation to previous work on disentanglement . . . 70
2.4 Identifiability theory . . . 71
2.4.1 Identifiability up to equivalence class . . . 71
2.4.2 Strongly exponential family . . . 73
2.4.3 General results . . . 74
2.4.4 Characterization of the linear indeterminacy . . . 76
2.4.5 Consistency of estimation . . . 77
2.5 Experiments . . . 78
2.5.1 Mean correlation coefficient as a measure of identifiability 78
2.5.2 Simulations on nonlinear ICA data . . . 79
2.5.3 Application to causal discovery . . . 84
2.6 Conclusion . . . 87
Appendices to Chapter 2 . . . 89
2.A Properties of the strongly exponential family . . . 89
2.B Proofs . . . 91
2.B.1 Identifiability proofs . . . 91
2.B.2 Identifiability under alternative assumptions . . . 99
2.B.3 Consistency proof . . . 100
2.C Unidentifiability of generative models with unconditional prior 101
2.C.1 Factorial priors . . . 101
2.C.2 General priors . . . 102
2.D Identifiability up to equivalence class: examples . . . 103
2.E Link between maximum likelihood and total correlation . . . 105

2.F Experimental protocol and additional experiments 106
2.F.1 Details of implementation for VAE experiments 106
2.F.2 Hippocampal fMRI data 107
2.F.3 Additional experiments 107

Chapter 3: Independently modulated component analysis 113
3.1 Introduction 114
3.2 Independently modulated component analysis 115
3.2.1 Definition of the generative model 115
3.2.2 Identifiability 116
3.2.3 Theoretical analysis 117
3.3 Estimation of IMCA 121
3.4 Conclusion 123
Appendices to Chapter 3 124
3.A Identifiability proofs 124
3.B Estimation proofs 126

Chapter 4: Identifiable conditional energy-based models 129
4.1 Introduction 130
4.2 Identifiable conditional energy-based deep models 132
4.2.1 Model definition 132
4.2.2 Identifiability 133
4.2.3 Universal approximation capability 137
4.3 An identifiable neural network architecture 138
4.4 Applications 139
4.4.1 Estimation of identifiable latent variable models 139
4.4.2 Transfer learning 142
4.5 Experiments 143
4.5.1 Identifiability of representations on image datasets . . . 143
4.5.2 IMCA and nonlinear ICA simulations 147
4.6 Conclusion 148
Appendices to Chapter 4 150
4.A Experimental protocol 150
4.A.1 Architectures and hyperparameters 150
4.A.2 The MCC metric 151

4.A.3 Further experiments . . . 153
4.B Estimation algorithms . . . 165
4.B.1 Conditional denoising score matching . . . 166
4.B.2 Conditional flow contrastive estimation . . . 169
4.C Identifiability of the conditional energy-based model . . . 171
4.C.1 Weak identifiability . . . 171
4.C.2 Strong identifiability . . . 172
4.C.3 Universal approximation capability . . . 176
4.D An identifiable architecture . . . 180
4.E Latent variable estimation in generative models . . . 189
4.E.1 Assumptions . . . 190
4.E.2 Proofs . . . 191

Chapter 5: Causal autoregressive flows 199
5.1 Introduction . . . 200
5.2 Preliminaries . . . 202
5.2.1 Structural equation models . . . 202
5.2.2 Autoregressive normalizing flows . . . 203
5.3 Causal autoregressive flow model . . . 204
5.3.1 From autoregressive flow models to SEMs . . . 204
5.3.2 Model definition and identifiability . . . 206
5.3.3 Choosing causal direction using likelihood ratio . . . 208
5.3.4 Extension to multivariate data . . . 209
5.4 Causal inference using autoregressive flows . . . 210
5.4.1 Interventions . . . 210
5.4.2 Counterfactuals . . . 211
5.5 Experiments . . . 212
5.5.1 Causal discovery . . . 212
5.5.2 Interventions . . . 218
5.5.3 Counterfactuals . . . 219
5.6 Related methods . . . 220
5.7 Conclusion . . . 222
Appendices to Chapter 5 . . . 223
5.A Proofs and additional results . . . 223
5.A.1 Identifiability of the affine causal model . . . 223

5.A.2 Affine autoregressive flows are transitive 229
5.A.3 Affine flows are not universal density approximators . . 232
5.A.4 Universality of the causal function 233
5.B Experimental protocol 234
5.B.1 Architectures and hyperparameters 234
5.B.2 Preprocessing of EEG data 235
5.B.3 Preprocessing of functional MRI data 235

Conclusion **237**
Thesis summary 237
Perspectives and future work 239

Bibliography **241**

List of Figures

Fig. 2.1 Identifiability on a 2D example 81
Fig. 2.2 Performance of iVAE compared to VAE variants 82
Fig. 2.3 Performance of iVAE compared to TCL 83
Fig. 2.4 Dimensionality reduction, hyperparameter selection and discrete ICA 85
Fig. 2.5 Application of iVAE to causal discovery on synthetic data . . 86
Fig. 2.6 Application of iVAE to causal discovery on fMRI data 88
Fig. 2.7 Various source distributions used in nonlinear ICA 106
Fig. 2.8 Recovered latents for iVAE and VAE variants 108
Fig. 2.9 Recovered latents in higher dimension 109
Fig. 2.10 Latent space trained on MNIST 110
Fig. 2.11 Latent space traversal on MNIST 110
Fig. 2.12 Samples from the learned model on MNIST 111
Fig. 2.13 Latent space trained on FashionMNIST 111
Fig. 2.14 Latent space traversal on FashionMNIST 112
Fig. 2.15 Samples from the learned model on FashionMNIST 112

Fig. 4.1 Quantifying the identifiability of learned representations on image datasets 145
Fig. 4.2 Transfer learning on image datasets 145
Fig. 4.3 Simulations on synthetitc nonlinear ICA and IMCA data . . 147
Fig. 4.4 Quantifying identifiability on MNIST and FashionMNIST . . 155
Fig. 4.5 Quantifying identifiability on CIFAR10 and CIFAR100 . . . 156
Fig. 4.6 Quantifying identifiability with Unets 158
Fig. 4.7 Transfer learning samples 158
Fig. 4.8 Transfer learning on image datasets (*cont.*) 159
Fig. 4.9 Transfer learning on image datasets (*cont.*) 160

Fig. 4.10 Transfer learning on image datasets (*cont.*) 161

Fig. 5.1 Causal discovery benchmark on synthetic data 213
Fig. 5.2 Impact of prior mismatch on performace 215
Fig. 5.3 Impact of neural network architecture on performace 216
Fig. 5.4 Arrow of time benchmark on EEG data 217
Fig. 5.5 Interventions on synthetic data 218
Fig. 5.6 Counterfactual predictions on synthetic data 220

List of Tables

Tab. 4.1 Application to transfer learning on image datasets 146
Tab. 4.2 Application to semi-supervised learning on image datasets . 146
Tab. 4.3 Architecture detail 152
Tab. 4.4 Architectures used in the experiments 153
Tab. 4.5 Transfer learning on MNIST 157
Tab. 4.6 Transfer learning on FashionMNIST 157
Tab. 4.7 Transfer learning on CIFAR10 161
Tab. 4.8 Transfer learning on CIFAR100 162
Tab. 4.9 Semi-supervised learning on MNIST 163
Tab. 4.10 Semi-supervised learning on FashionMNIST 163
Tab. 4.11 Semi-supervised learning on CIFAR10 164
Tab. 4.12 Architectures used in nonlinear ICA and IMCA simulations 166

Tab. 5.1 Cause Effect Pairs benchmark 217
Tab. 5.2 Interventions on fMRI data 219

Chapter 1

Introduction

Identifiability is a property of probabilistic models that is required for exact inference of the model parameters: a model is identifiable when two distinct parameters result in two different probability distributions. This thesis addresses the problem of identifiability in deep probabilistic models. More specifically, we begin by discussing its absence in popular representation learning approaches and then propose conditions under which this can be rectified. The identifiability of a new causal model derived from normalizing flows is next investigated, and we show how autoregressive flows are particularly well suited for a variety of causal inference tasks.

This thesis' methodology is based in large part on a set of mathematical tools and probabilistic models that we describe in Sections 1.1 to 1.3 of this introductory chapter. These sections are not intended to be a comprehensive overview of the subjects presented; instead, they are meant to give the essential context in order to motivate and develop the work discussed in Chapters 2 to 5. These tools serve as building blocks for the models that are developed in the remaining chapters. Section 1.4 gives a detailed overview of the contributions of each chapter as well as the structure of the thesis. Finally, the notations used in this thesis are introduced in Section 1.5.

1.1 Mathematical preliminaries

Probabilistic models. At its core, machine learning takes a probabilistic approach to data modelling. The observed data is assumed to be generated by a random process that follows a set of probabilistic assumptions. The "true"

unknown probability distribution P_0 underlying the data is approximated by a probabilistic model, which is defined as a pair $(\mathcal{X}, \mathcal{P})$ of a set of possible observations $\mathcal{X}$ and a set of probability distributions $\mathcal{P} = \{P_{\boldsymbol{\theta}} : \boldsymbol{\theta} \in \Theta\}$ with parameter $\boldsymbol{\theta}$. The model is said to be *well specified* if there exists a $\boldsymbol{\theta} \in \Theta$ such that $P_{\boldsymbol{\theta}} = P_0$. Although it is not often the case that the model contains the true distribution, the goal remains to find the best approximation $P_{\boldsymbol{\theta}*} \approx P_0$ with optimal parameter $\boldsymbol{\theta}^*$ from the data itself. Probabilistic models can be used to make predictions, generate new samples, make inference about missing values, extract useful information from the data, among many other applications.

As an example, consider that we flipped a possibly biased coin N times. A coin toss only has two outcomes: heads with probability $\theta \in [0, 1]$, or tails with probability $1 - \theta$. This event can be modelled by a probabilistic model consisting of Bernoulli distributions: $\mathcal{P} = \{\text{Ber}_\theta : \theta \in [0, 1]\}$ over the binary set $\mathcal{X} = \{0, 1\}$. The coin flips $\mathcal{D} = \{x^{(1)}, \dots, x^{(N)}\}$ are *independent* and *identically distributed (I.I.D)*: independent because the outcome of one coin toss bears no influence on the rest of the outcomes, and identically distributed because they originate from the same coin, and thus are all Ber_θ-distributed.

One of the most commonly used probabilistic models is the *exponential family* (Koopman, 1936; Kupperman, 1958; Andersen, 1970). It is a set of probability distributions whose probability density or mass function can be written as

$$p_{\boldsymbol{\theta}}(\mathbf{x}) = Q(\mathbf{x}) \exp\left[\mathbf{T}(\mathbf{x})^\top \boldsymbol{\lambda}(\boldsymbol{\theta}) - \Gamma(\boldsymbol{\theta})\right], \tag{1.1}$$

where $Q(\mathbf{x})$, $\mathbf{T}(\mathbf{x})$, $\boldsymbol{\lambda}(\boldsymbol{\theta})$ and $\Gamma(\boldsymbol{\theta})$ are known functions. The exponential family includes the Bernoulli, Gaussian and Laplace distributions amongst many other. A more detailed overview of the exponential family and some of its useful properties can be found in Appendix 1.B.

Likelihood function. Given a dataset $\mathcal{D}$ of i.i.d observations, we can define the *likelihood function* $\mathcal{L}(\boldsymbol{\theta})$ as the probability of observing the data given the parameter $\boldsymbol{\theta}$:

$$\mathcal{L}(\boldsymbol{\theta}) = P_{\boldsymbol{\theta}}(\mathbf{x}^{(1)}, \dots, \mathbf{x}^{(N)}) = \prod_{i=1}^{N} P_{\boldsymbol{\theta}}(\mathbf{x}^{(i)}). \tag{1.2}$$

The second equality in the definition above is a result of the independence[1] of the random variables $\mathbf{x}^{(1)}, \dots, \mathbf{x}^{(N)}$. The likelihood function $\mathcal{L}(\boldsymbol{\theta})$ can be

[1] See Appendix 1.A for a brief overview on statistical independence.

used as a means to estimate the optimal parameter $\boldsymbol{\theta}^*$. Intuitively, it is less likely to observe the dataset $\mathcal{D}$ under alternative parameters $\boldsymbol{\theta}$, since the data was generated from $P_{\boldsymbol{\theta}^*}$. Thus, the likelihood function is maximal for $\boldsymbol{\theta} = \boldsymbol{\theta}^*$. This estimation technique is known as *maximum likelihood estimation* (MLE) (Wilks, 1938).

As an illustration, let us go back to the previous example. Suppose that $N_h = 600$ out of $N = 1000$ coin flips were heads. We want to estimate the probability of flipping heads on the next toss using MLE. Using equation (1.2) and the definition of the Bernoulli distribution, the likelihood for the model is

$$\mathcal{L}(\theta) = \mathrm{Ber}_\theta(\mathcal{D}) = \theta^{N_h}(1-\theta)^{N-N_h}.$$

To find the value θ^* that maximizes $\mathcal{L}$, we can differentiate the above expression with respect to its argument θ, and solve for zero. For the coin toss, the likelihood is maximal for $\theta^* = {}^{N_h}/_N = 0.6$, at which it takes the extremely small value $\mathcal{L}(0.6) = 0.6^{600}0.4^{400} \approx 5.184 \times 10^{-293}$.

We often use the logarithm of the likelihood (or *log-likelihood*) $\ell(\boldsymbol{\theta}) \coloneqq \log \mathcal{L}(\boldsymbol{\theta})$ for maximum likelihood estimation. This does not change the optimization landscape since the logarithm is a strictly increasing function, and has the benefit of transforming the product in equation (1.2) into a sum over an I.I.D sample. Incidentally, the logarithm also addresses the problem of extremely small values, and makes differentiating easier for many common probabilistic models, especially when using exponential family distributions defined in equation (1.1). The maximum likelihood estimator is *consistent*. This means that given an infinite amount of data, we can estimate the optimal parameter $\boldsymbol{\theta}^*$ with arbitrary precision.

Latent variable models. Directly modelling the distribution of an observed variable $\mathbf{x}$ often leads to simple, "single-stage" probabilistic models, which often have closed-form expressions for learning and inference. However, this comes at the expense of expressiveness. Latent variable models are a class of probabilistic models that offers an alternative approach to modelling complex distributions. They posit the existence of a unobserved, *latent variable* $\mathbf{z}$, from which the observation $\mathbf{x}$ is generated through a deterministic or stochastic transformation.

Typically, the joint probability density or mass function of a latent variable model has the following structure:

$$p_{\boldsymbol{\theta}}(\mathbf{x}, \mathbf{z}) = p_{\boldsymbol{\theta}}(\mathbf{x}|\mathbf{z})p_{\boldsymbol{\theta}}(\mathbf{z}).$$

This approach allows us to combine simple distributions into richer, more flexible forms. For example, we can model each of $p_{\boldsymbol{\theta}}(\mathbf{z})$ and $p_{\boldsymbol{\theta}}(\mathbf{x}|\mathbf{z})$ as an exponential family distribution (1.1). Yet, the resulting observed probability density or mass function

$$p_{\boldsymbol{\theta}}(\mathbf{x}) = \int p_{\boldsymbol{\theta}}(\mathbf{x}|\mathbf{z})p_{\boldsymbol{\theta}}(\mathbf{z})\mathrm{d}\mathbf{z}$$

can still model a rich class of data distributions.

Identifiability. A probabilistic model $\mathcal{P}$ is said to be *identifiable* if the mapping $\boldsymbol{\theta} \in \Theta \mapsto (\mathbf{x} \in \mathcal{X} \mapsto P_{\boldsymbol{\theta}}(\mathbf{x}))$ is injective:

$$(P_{\boldsymbol{\theta}_1}(\mathbf{x}) = P_{\boldsymbol{\theta}_2}(\mathbf{x}), \forall \mathbf{x} \in \mathcal{X}) \implies \boldsymbol{\theta}_1 = \boldsymbol{\theta}_2. \tag{1.3}$$

In other words, if two parameters $\boldsymbol{\theta}_1$ and $\boldsymbol{\theta}_2$ generate the same distribution over the set of observations $\mathcal{X}$, then they are necessarily equal.

Identifiability is an important property of probabilistic models. It enables precise parameter estimation in the limit of infinite data samples. In fact, the consistency of MLE (and other parameter estimation techniques) is premised on the identifiability of the probabilistic model: under the identifiability condition, the likelihood function $\mathcal{L}(\boldsymbol{\theta})$ has a unique maximum at $\boldsymbol{\theta}^*$.

It is important to note that identifiability is a property of the probabilistic model and not of the estimation method. To illustrate this, we consider the previous coin toss example but defined in two different ways. First, the Bernoulli model above directly models the outcome of a coin flip as a Bernoulli-distributed random variable and is identifiable. To see this, let $(\theta_1, \theta_2) \in \Theta^2$ such that $\mathrm{Ber}_{\theta_1}(x) = \mathrm{Ber}_{\theta_2}(x)$ for $x \in \{0, 1\}$. Since $\mathrm{Ber}_{\theta}(1) = \theta$, we conclude that $\theta_1 = \theta_2$ and that the model is identifiable.

Now imagine that we do not have access to a real coin, but rather to a computer with a random number generator that can only generate samples from a Gaussian distribution with mean μ and variance σ^2. Tossing a coin in this case proceeds in two steps: we first draw a sample $z^{(i)} \sim \mathcal{N}(\mu, \sigma^2)$, then

we assign heads to $x^{(i)}$ if $z^{(i)} \geq 0$ and tails otherwise. This is effectively a latent variable model with parameters (μ, σ) and can be expressed as

$$z^{(i)} \sim \mathcal{N}(\mu, \sigma^2),$$
$$x^{(i)} = \mathbb{1}(z^{(i)} \geq 0).$$

This model is identifiable if two different values of the parameter $\boldsymbol{\theta} = (\mu, \sigma)$ result in two different *observed* distributions. The observed random variable x is Bernoulli distributed with parameter $\tilde{\theta}$ depending on μ and σ. This relationship is

$$\tilde{\theta} = \mathbb{P}(z \geq 0) = 1 - \mathbb{P}(z < 0) = 1 - \phi\left(-\frac{\mu}{\sigma}\right),$$

where ϕ is the *cumulative distribution function*[2] (CDF) of the standard (zero mean and unit variance) Gaussian distribution. Crucially, the value of $\tilde{\theta}$ depends on the ratio μ/σ, which takes the same value for infinitely many pairs (μ, σ), which means that this model is not identifiable.

The definition of identifiability in equation (1.3) may be limiting in some fields of machine learning. We will later (Section 2.4.1) introduce the notion of identifiability up to equivalence class, which is more adaptable and better suited for applications like representation learning and causal discovery.

1.2 Identifiability in representation learning

Recent advances in data collection have resulted in very large datasets, including images (LeCun et al., 1998; Deng et al., 2009; Krizhevsky, 2009), 3D shapes (Chang et al., 2015), text (Marcus et al., 1993; Maas et al., 2011), music (Bertin-Mahieux et al., 2011), and graphs and networks (Yanardag and Vishwanathan, 2015; Hu et al., 2020). As the amount and complexity of the data grew, most of the work in machine learning research went towards developing preprocessing pipelines to assist the extraction of meaningful information from large datasets, allowing for efficient learning. With the rise of deep learning, preprocessing shifted from hand-crafted expertise-based feature engineering to utilizing neural networks to implicitly learn useful representations. This is known as *representation learning*, and it has grown to be one of the modern pillars of machine learning. Deep representations are now widely used in many

[2]The CDF of a continuous random variable X is defined as $F_X(x) = \mathbb{P}(X < x)$.

machine learning applications, including speech recognition and processing (Dahl et al., 2011; Seide et al., 2011), natural language processing (Bengio et al., 2003; Devlin et al., 2019), action recognition (Korbar et al., 2018), domain adaptation (Wang and Deng, 2018), and many more.

Learning good representations can have a significant impact on the performance of machine learning algorithms (Bengio et al., 2013). A representation's quality is frequently characterized by its capacity to improve the performance of a downstream task in which the user is actually engaged. This criterion, however, is only meaningful when such a task exists and is clearly defined, which usually amounts to classification or regression on labelled datasets[3]. A better criterion for assessing the quality of a representation should be inherent to the representation itself, rather than reliant on the context or task in which it may be employed. To this end, recent lines of research aim to learn representations that are true to the explanatory factors of variation behind the data. This desideratum is formalized by the notion of identifiability. Fundamentally, an identifiable and sufficiently flexible probabilistic model can only learn one representation in the limit of infinite data: the ground truth generative factors. Identifiability is necessary for learning representations that are semantically meaningful, reproducible, interpretable and better suited for downstream tasks (Schmidhuber et al., 1996; Bengio et al., 2013; Peters et al., 2017).

In this section, we will briefly review recent methods for representation learning that aim to learn the ground truth generative factors. In particular, we will focus on nonlinear Independent Component Analysis (ICA) (Hyvärinen and Morioka, 2016; Hyvärinen and Morioka, 2017; Hyvärinen et al., 2019), as well as the related field of disentangled representation learning (Higgins et al., 2017; Chen et al., 2018; Esmaeili et al., 2019). This thesis builds on recent advances in nonlinear ICA to propose novel methods for learning identifiable representations.

1.2.1 Disentangled representation learning

Many methods like probabilistic component analysis (Tipping and Bishop, 1999) and variational autoencoders (Kingma and Welling, 2014; Rezende et al.,

[3]Because labelling is a costly and time-consuming endeavour, only a small percentage of today's datasets are labelled.

2014) learn a posterior distribution over a lower-dimensional latent variable. It is hoped that such a posterior will correspond to the underlying distribution of statistically independent sources of variation. A new line of research is being developed for the related goal of learning *disentangled representations* (Alemi et al., 2017; Higgins et al., 2017; Burgess et al., 2018; Chen et al., 2018; Kim and Mnih, 2018; Mathieu et al., 2018; Esmaeili et al., 2019).

In brief, a disentangled representation is one in which single latent components are responsive to changes in a single generative factor (Bengio et al., 2013; Burgess et al., 2018). The objective is to isolate the influence of all factors of variation, which translates to learning a representation with independent components. The majority of previous efforts on disentanglement are based on variational autoencoders (VAEs) and employ regularized objectives to encourage the latent representation to align with the independent components of variation. This is accomplished by augmenting a VAE's loss with hyperparameters that favour a factorized latent representation.

Variational autoencoders. We begin with a brief overview of VAEs (Kingma and Welling, 2014; Rezende et al., 2014), which serve as the foundation for most of the recent disentanglement methods (Higgins et al., 2017; Zhao et al., 2017; Achille and Soatto, 2018; Burgess et al., 2018; Chen et al., 2018; Kim and Mnih, 2018; Mathieu et al., 2018; Esmaeili et al., 2019; Gao et al., 2019). VAEs are a latent variable model aimed at reproducing samples from a given dataset. We suppose that the observation $\mathbf{x}$ is generated by a latent variable $\mathbf{z}$ through a random process consisting of a prior $p_{\boldsymbol{\theta}}(\mathbf{z})$ and a likelihood $p_{\boldsymbol{\theta}}(\mathbf{x}|\mathbf{z})$, also known as a *decoder*. Much of this process is concealed, such as the optimal parameter $\boldsymbol{\theta}^*$ and the latent variable values for each observed data point. To perform inference or learning, a *variational posterior* distribution (also known as an *encoder*) $q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x})$ is introduced, which is used to approximate the often intractable posterior $p_{\boldsymbol{\theta}}(\mathbf{z}|\mathbf{x})$. The encoder and the decoder define a probabilistic autoencoder. The VAE objective is traditionally defined in terms of a lower bound $\mathcal{L}$ on the empirical expectation of the log-likelihood over a dataset $\mathcal{D}$:

$$\mathbb{E}_{p_{\mathcal{D}}(\mathbf{x})}\left[\log p_{\boldsymbol{\theta}}(\mathbf{x})\right] = \mathbb{E}_{p_{\mathcal{D}}(\mathbf{x})}[\mathrm{KL}(q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x})\|p_{\boldsymbol{\theta}}(\mathbf{z}|\mathbf{x}))] + \mathcal{L}(\boldsymbol{\theta}, \boldsymbol{\phi}),$$

where

$$\mathcal{L}(\boldsymbol{\theta}, \boldsymbol{\phi}) = \mathbb{E}_{p_{\mathcal{D}}(\mathbf{x})}\left[\mathbb{E}_{q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x})}\left[\log p_{\boldsymbol{\theta}}(\mathbf{x}|\mathbf{z})\right] - \mathrm{KL}(q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x})\|p_{\boldsymbol{\theta}}(\mathbf{z}))\right]. \tag{1.4}$$

The loss $\mathcal{L}$ is a lower bound on the data-log-likelihood called the *evidence lower bound* (ELBO). We can learn flexible models with a VAE by utilizing neural networks to parametrize the various distributions in equation (1.4). For efficiency, we frequently make assumptions about the form of the distributions in the generative process. Most importantly, $p_{\boldsymbol{\theta}}(\mathbf{z})$ is assumed to have a diagonal covariance, typically a standard Gaussian distribution, implying that the latent components are independent. This independence assumption serves as the foundation for VAE-based disentanglement methods.

β-VAE. Higgins et al. (2017) introduced β-VAE, a variant where a positive hyperparameter is added to the original VAE objective (1.4). The variational lower bound becomes

$$\mathcal{L}_{\beta}(\boldsymbol{\theta}, \boldsymbol{\phi}) = \mathbb{E}_{p_{\mathcal{D}}(\mathbf{x})}\left[\mathbb{E}_{q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x})}\left[\log p_{\boldsymbol{\theta}}(\mathbf{x}|\mathbf{z})\right] - \beta\,\mathrm{KL}(q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x})\|p_{\boldsymbol{\theta}}(\mathbf{z}))\right]. \tag{1.5}$$

The idea behind introducing the β parameter is to encourage a more disentangled latent representation. When the parameter β is set to a large value, the posterior distribution is forced to align with the factorized prior distribution.

FactorVAE and β-TC-VAE. The Kullback-Leibler divergence between the prior $p_{\boldsymbol{\theta}}(\mathbf{z})$ and the variational posterior $q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x})$ in equation (1.5) can be further decomposed into

$$\mathrm{KL}(q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x})\|p_{\boldsymbol{\theta}}(\mathbf{z})) = I(\mathbf{x}, \mathbf{z}) + \mathrm{KL}(q_{\boldsymbol{\phi}}(\mathbf{z})\|\prod_i q_{\boldsymbol{\phi}}(z_i)) + \sum_i \mathrm{KL}(q_{\boldsymbol{\phi}}(z_i)\|p_{\boldsymbol{\theta}}(z_i)), \tag{1.6}$$

where $I(\mathbf{x}, \mathbf{z})$ denotes the mutual information between $\mathbf{x}$ and $\mathbf{z}$ and $q_{\boldsymbol{\phi}}(\mathbf{z}) = \int q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x})p_{\mathcal{D}}(\mathbf{x})\mathrm{d}\mathbf{x}$. Therefore, when $\beta > 1$, the β-VAE also penalizes the mutual information $I(\mathbf{x}, \mathbf{z})$, effectively constraining the capacity of the VAE bottleneck, which results in a loss of information about the observations. Chen et al. (2018) and Kim and Mnih (2018) contend that penalizing the mutual information $I(\mathbf{x}, \mathbf{z})$ is undesirable for disentanglement. Instead, they propose to penalize the total correlation term $\mathrm{KL}(q_{\boldsymbol{\phi}}(\mathbf{z})\|\prod_i q_{\boldsymbol{\phi}}(z_i))$ in equation (1.6). Because total correlation is a measure of dependency between random variables, imposing such a penalty forces the model to find statistically independent latent components. The resulting models, FactorVAE (Kim and Mnih, 2018)

and β-TC-VAE (Chen et al., 2018), minimize an alternative lower bound:

$$\mathcal{L}_{TC}(\boldsymbol{\theta}, \boldsymbol{\phi}) = \mathcal{L}(\boldsymbol{\theta}, \boldsymbol{\phi}) - \beta \, \mathrm{KL}(q_{\boldsymbol{\phi}}(\mathbf{z}) \| \prod_i q_{\boldsymbol{\phi}}(z_i)). \tag{1.7}$$

Since the total correlation is intractable, Kim and Mnih (2018) propose an estimate using the density ratio trick (Nguyen et al., 2010; Sugiyama et al., 2012), which entails training a classifier/discriminator to approximate the density ratio that emerges in the KL term. Concurrently, Chen et al. (2018) present a stochastic biased Monte-Carlo estimate for the terms involved in computing the total correlation. When the hyperparameter β in equation (1.7) is positive, it encourages the aggregated posterior $q_\phi(\mathbf{z})$ to have independent marginals without penalizing the mutual information $I(\mathbf{x}, \mathbf{z})$).

Disentangled representations are not necessarily identifiable. Many methods learn disentangled representations by imposing independence on the latent variables and adding regularization terms to the VAE objective in an ad-hoc manner (Kumar et al., 2017; Zhao et al., 2017; Achille and Soatto, 2018; Esmaeili et al., 2019; Gao et al., 2019). Unfortunately, these techniques do not provide any theoretical identifiability guarantees.

In fact, disentangled representations are not identifiable in general. In other words, learning nonlinear models that seek independence results in arbitrary representations that are not always related to the ground truth factors of variation. A large scale empirical study conducted by Locatello et al. (2019) showed that the proposed models for disentanglement exhibit substantial variance depending on hyperparameters and random seeds. Unsupervised learning of identifiable nonlinear representations has long been known to be theoretically impossible (Hyvärinen and Pajunen, 1999) without any *inductive biases*. Mathieu et al. (2018), Rolinek et al. (2018), and Locatello et al. (2019) address the issue of identifiability, or lack thereof, in deep latent variable models, particularly VAEs, demonstrating that isotropic prior distributions always lead to rotational invariance in the ELBO.

1.2.2 Independent component analysis

Within representation learning, identifiability has mostly been studied in the context of *independent component analysis* (ICA) (Comon, 1994). In ICA, the

observations are considered to be a mixture of independent source variables. The goal is to learn an "unmixing" transformation capable of recovering the original sources based on their independence and the observed mixture.

In this section, we briefly introduce the basic concepts and identifiability theory of linear ICA. We then review the recent advances in nonlinear ICA, which are the starting point for most of the work presented in this thesis.

1.2.2.1 Linear ICA

Over the decades, ICA has been extensively studied in the linear setting, where the mixing is considered to be a matrix (Comon, 1994; Bell and Sejnowski, 1995; Hyvärinen, 1999; Cardoso, 2001; Hyvärinen et al., 2001; Pham and Cardoso, 2001; Hyvärinen et al., 2003; Plumbley, 2003; Hyvärinen and Hurri, 2004; Le et al., 2011; Pfister et al., 2019; Podosinnikova et al., 2019), with applications in finance (Back and Weigend, 1997; Oja et al., 2000), study of functional magnetic resonance imaging (fMRI) data (McKeown et al., 1998; McKeown and Sejnowski, 1998; Calhoun et al., 2003), document analysis (Bingham et al., 2002; Podosinnikova et al., 2015), astronomy (Nuzillard and Bijaoui, 2000), analysis of electroencephalography (EEG) data (Makeig et al., 1996; Makeig et al., 1997; Delorme et al., 2007; Milne et al., 2009), and many more. Essentially, if at most one of the source variables is Gaussian, linear ICA is identifiable up to permutation and scaling of the components. In this section, we briefly review the linear ICA model. The following exposition is based on the seminal paper by Comon (1994) as well as the monograph on linear ICA by Hyvärinen et al. (2001).

Consider a latent source variable $\mathbf{s}$ which is transformed through an unknown linear mixing into observations $\mathbf{x}$:

$$\mathbf{x} = \mathbf{A}\mathbf{s}, \tag{1.8}$$

where $\mathbf{A} \in \mathbb{R}^{d \times d}$ is an invertible matrix.

We want to know if we can recover the original but unknown signal $\mathbf{s}$ while making no or only very weak assumptions on its distribution. The difficulty is that both the distribution of $\mathbf{s}$ and the mixing matrix $\mathbf{A}$ are unknown, making it impossible to determine whether a solution to the problem is related to the true generative process. This problem is known as *blind source separation* (BSS).

Comon (1994) proposed the framework of Independent Component Analysis (ICA), which is an estimation technique that provides an affirmative answer to the question, by only making two weak assumptions:

1. The components $s_1, \dots s_d$ of the source variable $\mathbf{s}$ are independent:
$$p(\mathbf{s}) = \prod_i p_i(s_i). \tag{1.9}$$
2. At most one component has a Gaussian distribution.

Under these assumptions, the model (1.8) is identifiable, meaning that the linear mixing, as well as the true source variable, can be estimated. Linear ICA achieves this goal by simply learning an *unmixing* matrix $\mathbf{B}$ such that
$$\mathbf{z} \coloneqq \mathbf{B}\mathbf{x} \tag{1.10}$$
has independent components. Before we state the main identifiability result, we note that the linear ICA problem has two indeterminacies. To see this, let $\mathbf{a}_i$ be the i-th column of $\mathbf{A}$, and write equation (1.8) as:
$$\mathbf{x} = \sum_{i=1}^{d} s_i \mathbf{a}_i.$$
Then it is clear that the ICA problem is invariant to permutation and scaling of the independent components.

The main identifiability result of linear ICA is summarized by the following theorem.

Theorem 1.1 (Comon (1994), Theorem 11 adapted)**.** *Let* $\mathbf{s}$ *be a random vector that satisfies assumptions 1 and 2, i.e. it has independent components, of which at most one is Gaussian. Let* $\mathbf{C}$ *be an orthogonal* $d \times d$ *matrix and* $\mathbf{z} = \mathbf{C}\mathbf{s}$*. Then the components* z_i *of* $\mathbf{z}$ *are independent if and only if* $\mathbf{C} = \mathbf{D}\mathbf{P}$*, with* $\mathbf{D}$ *diagonal and* $\mathbf{P}$ *a permutation matrix.*

Theorem 1.1 shows that outside of the ambiguities mentioned above, the linear ICA problem is identifiable. A reconstruction $\mathbf{z} = \mathbf{B}\mathbf{x} = \mathbf{B}\mathbf{A}\mathbf{s} = \mathbf{C}\mathbf{s}$ has independent components if and only if it is equal to a permutation and scaling of the components of $\mathbf{s}$. In other words, based on independence alone, if $(\mathbf{A}, \mathbf{s})$ is a solution to the ICA problem (1.8), then all the other solutions necessarily have the form $(\mathbf{A}\mathbf{P}^{-1}\mathbf{D}^{-1}, \mathbf{D}\mathbf{P}\mathbf{s})$ where $\mathbf{P}$ is a permutation matrix and $\mathbf{D}$ is a diagonal matrix.

1.2.2.2 Nonlinear ICA

A straightforward generalization of ICA to the nonlinear setting would assume that a set of independent random vectors are mixed into identically distributed observations through an arbitrary but usually smooth transformation. The matrix $\mathbf{A}$ in the linear ICA model in equations (1.8) and (1.9) is replaced by an invertible mixing function $\mathbf{f} : \mathbb{R}^d \to \mathbb{R}^d$:

$$\begin{aligned} \mathbf{x} &= \mathbf{f}(\mathbf{s}), \\ p(\mathbf{s}) &= \prod_i p_i(s_i). \end{aligned} \tag{1.11}$$

The goal of nonlinear ICA is to learn an unmixing function $\mathbf{g}$ that generalizes the unmixing matrix $\mathbf{B}$ in equation (1.10) such that

$$\mathbf{z} \coloneqq \mathbf{g}(\mathbf{x}) \tag{1.12}$$

has independent components. In the linear setting, solving the blind source separation (BSS) problem of recovering the original signal $\mathbf{s}$ is equivalent to finding independent components through ICA. However, a fundamental problem with nonlinear ICA is that solutions to equation (1.12) such that $\mathbf{z}$ has independent components exist, and they are highly non-unique. As a result, they are not necessarily meaningfully related to the solution to *nonlinear* BSS.

In reality, in the nonlinear case, identifiability is a far more difficult aim to achieve. If s_1 and s_2 are independent random variables, then so are $h_1(s_1)$ and $h_2(s_2)$ for any functions h_1 and h_2. Component-wise transformations $\mathbf{h}(\mathbf{s}) \coloneqq (h_1(s_1), \ldots, h_d(s_d))$ as well as permutation of the components are thus unavoidable indeterminacies of nonlinear BSS.

The real problem is that the unidentifiability of nonlinear ICA extends beyond these two ambiguities: two independent components s_i and s_j can be mixed nonlinearly while remaining statistically independent. Nonlinear transformations introduce numerous degrees of freedom, rendering the problem ill-defined. Hyvärinen and Pajunen (1999) demonstrated that it is always possible to construct a representation $\mathbf{z} = \mathbf{g}(\mathbf{x})$ with independent components that is nonetheless a nonlinear mixture of the underlying independent generative factors. This construction can be traced back to Darmois (1953), which showed that for any two independent variables ξ_1, ξ_2, we can construct infinitely many random variables $y_1 = f_1(\xi_1, \xi_2)$ and $y_2 = f_2(\xi_1, \xi_2)$ that are also independent. This unidentifiability result is summarized by the following theorem.

Theorem 1.2 (Hyvärinen and Pajunen (1999), Theorem 1 adapted)**.** *Let* $\mathbf{x}$ *be a random vector of any distribution. Then there exists a transformation* $\mathbf{g} : \mathbb{R}^d \to [0,1]^d$ *such that* $\mathbf{z} = \mathbf{g}(\mathbf{x})$ *has a uniform distribution. In particular, the components* $z_i \in \mathbf{z}$ *are independent.*

The function $\mathbf{g}$ in Theorem 1.2 is constructed through an iterative procedure analogous to Gram-Schmidt orthogonalization, by recursively applying the conditional cumulative distribution function (CDF) of $\mathbf{x}$:

$$z_i = g_i(\mathbf{x}_{1:i}) \coloneqq \int_{-\infty}^{x_i} p(\tilde{x}_i|\mathbf{x}_{1:i-1})\mathrm{d}\tilde{x}_i. \tag{1.13}$$

This theorem indicates that nonlinear ICA is unidentifiable, which can be seen in few different ways. To begin, it is clear from equation (1.13) that $z_1 = g_1(x_1)$ is a monotonic transformation of the observed variable x_1. By simply permuting the elements of $\mathbf{x}$ before applying $\mathbf{g}$, we conclude that any of the x_i can be taken as one of the independent components, which is unrealistic. Another corollary of the construction (1.13) is that the transformation $\mathbf{g} \coloneqq (g_1, \ldots, g_d)$ as well as its inverse have a triangular Jacobian matrix. If the true mixing $\mathbf{f}$ in equation (1.11) has a full Jacobian, then $\mathbf{z}$ is necessarily a nonlinear mixing of the original source $\mathbf{s}$. Indeed, if this was not the case, we would have

$$\mathbf{f} = \mathbf{P} \circ \mathbf{h} \circ \mathbf{g}^{-1}, \tag{1.14}$$

where $\mathbf{P}$ and $\mathbf{h}$ formalize the permutation and component-wise scaling ambiguities of nonlinear BSS. The right-hand side of equation (1.14) has a triangular Jacobian (up to permutation) whereas the left-hand side has a full Jacobian, which is not possible.

To make nonlinear ICA models identifiable, additional information is needed, which can be in the form of structural constraints on the mixing (Hecht-Nielsen, 1995; Pajunen et al., 1996; Taleb and Jutten, 1999; Lappalainen and Honkela, 2000; Eriksson and Koivunen, 2002) or by restricting the distribution of the latent variables (Harmeling et al., 2003; Hosseini and Jutten, 2003; Jutten et al., 2004; Sprekeler et al., 2014).

In recent years, there has been a renaissance in identifiability results that follow this second approach of constraining the distribution of the independent components. Sprekeler et al. (2014) assume that the independent components are autocorrelated time series; Hyvärinen and Morioka (2016) assume that

they are non-stationary time series; Hyvärinen and Morioka (2017) assume that they have general non-Gaussian temporal dependencies; Hyvärinen et al. (2019) assume that we have access to an auxiliary variable that modulates the distribution of the independent components; and Gresele et al. (2020) assume that we have multiple views of the same independent components. These models achieved significant progress towards providing identifiability guarantees by integrating side information into the generative model. We briefly review the works of Hyvärinen and Morioka (2016) and Hyvärinen et al. (2019) as they are of particular interest for the models discussed in this thesis.

Nonlinear ICA by Time Contrastive Learning. Time Contrastive Learning (TCL) introduced by Hyvärinen and Morioka (2016) is a method for nonlinear ICA based on the assumption that while the sources are independent, they are also *non-stationary* time series. This implies that they can be divided into known non-overlapping segments, such that their distributions vary across segments. The non-stationarity is supposed to be slow compared to the sampling rate, allowing us to consider the distributions inside each segment to be constant over time, and resulting in a *piece-wise stationary* distribution across segments. Formally, given a segment index $\tau \in \mathcal{T}$, where $\mathcal{T}$ is a finite set of indices, the distribution of each latent component z_i within that segment is modelled as an exponential family:

$$\log p_\tau(z_i) = \log q_{i,0}(z_i) + \sum_{j=1}^{k} \lambda_{i,j}(\tau) q_{i,j}(z_i) - \log Z_i(\lambda_{i,1}(\tau), \ldots, \lambda_{i,k}(\tau)), \quad (1.15)$$

where $q_{i,0}$ is a stationary baseline and $\mathbf{q}_i \coloneqq (q_{i,1}, \ldots, q_{i,k})$ is the sufficient statistic for the exponential family of the source z_i. Note that the parameters $\boldsymbol{\lambda}_i \coloneqq (\lambda_{i,1}, \ldots, \lambda_{i,k})$ depend on the segment index, indicating that the distributions of the sources change across segments. It follows from equation (1.15) that the observations are piece-wise stationary.

TCL recovers the inverse transformation $\mathbf{f}^{-1}$ by *self-supervised learning*, where the goal is to classify original data points against segment indices in a multinomial classification task. To this end, TCL employed a deep network consisting of a feature extractor $\mathbf{h}(\mathbf{x}; \eta)$ with parameters η in the form of a neural network, followed by a final classifying layer (e.g. softmax). The theory of TCL, as stated in Theorem 1 of Hyvärinen and Morioka (2016), is

premised on the fact that in order to optimally classify observations into their corresponding segments, the feature extractor $\mathbf{h}(\mathbf{x};\eta)$ must learn about the changes in the underlying distribution of latent sources. The theory shows that the method can learn the independent components up to pointwise nonlinear transformations and a linear transformation $\mathbf{A}$. It is further proposed that a linear ICA can recover the linear mixing $\mathbf{A}$ if the number of segments grows to infinity and the segment distributions are random in a certain sense, but this latter assumption is unrealistic in applications where the number of segments is small.

Nonlinear ICA using auxiliary variables. A more recent development in nonlinear ICA is given by Hyvärinen et al. (2019). The observations $\mathbf{x}$ are still assumed to be a nonlinear invertible transformation of latent sources $\mathbf{z}$. The key element in this model that leads to identifiability is that each source z_i is dependent on some *observed auxiliary variable* $\mathbf{u}$, but independent of all the other sources conditional on $\mathbf{u}$:

$$\begin{aligned}\mathbf{x} &= \mathbf{f}(\mathbf{z}),\\ p(\mathbf{z}|\mathbf{u}) &= \prod_i p_i(z_i|\mathbf{u}).\end{aligned}$$

This formulation is so general that it subsumes previous models by Hyvärinen and Morioka (2016) and Hyvärinen and Morioka (2017) in the sense of the data model. In fact, in the case of non-stationary sources, the auxiliary variable $\mathbf{u}$ can be the segment label. More generally, the auxiliary variable $\mathbf{u}$ can be a class label, the index of a pixel in an image, the description of an image, the sound of a video, amongst others. The estimation technique is considerably different from TCL in that it learns the unmixing function using a self-supervised binary discrimination task based on randomization. More specifically, new data is constructed from the observations $\mathbf{x}$ and $\mathbf{u}$ to obtain two datasets

$$\begin{aligned}\tilde{\mathbf{x}} &= (\mathbf{x}, \mathbf{u}),\\ \tilde{\mathbf{x}}^* &= (\mathbf{x}, \mathbf{u}^*),\end{aligned}$$

where $\mathbf{u}^*$ is drawn randomly from the distribution of $\mathbf{u}$ and independent of $\mathbf{x}$. Then, nonlinear logistic regression is performed using a regression function of the form $r(\mathbf{x}, \mathbf{u}) = \sum_{i=1}^{d} \psi_i(h_i(\mathbf{x}), \mathbf{u})$ to discriminate between actual samples $\tilde{\mathbf{x}}$ and shuffled samples $\tilde{\mathbf{x}}^*$.

According to the generative model, the observed and the auxiliary variables in the non-shuffled dataset $\tilde{\mathbf{x}}$ are linked through a shared latent variable, whereas this link is broken in the shuffled dataset $\tilde{\mathbf{x}}^*$. Thus, the regression function makes use of a feature extractor denoted $\mathbf{h}(\mathbf{x})$ like in TCL, the purpose of which is to extract the latent features that allow distinguishing between the two datasets.

The identifiability theory of Hyvärinen et al. (2019) stipulates that the model is identifiable up to component-wise invertible transformations, provided that the latent distribution $p_i(z_i|\mathbf{u})$ satisfies some regularity constraints. In the particular case of the exponential family distribution

$$\log p(z_i|\mathbf{u}) = \log q_{i,0}(z_i) + \sum_{j=1}^{k} \lambda_{i,j}(\mathbf{u}) q_{i,j}(z_i) - \log Z_i(\lambda_{i,1}(\mathbf{u}), \ldots, \lambda_{i,k}(\mathbf{u})), \quad (1.16)$$

the regularity constraints are much simpler, but the identifiability guarantees are weakened:

- If $k > 1$ in equation (1.16), then the same identifiability result holds.
- If $k = 1$, then it is only possible to recover the latent variables up to an unknown linear transformation $\mathbf{A}$, that can be solved by linear ICA, like in TCL.

1.3 Identifiability in causal discovery

Causal models are a crucial part in modern scientific endeavour (Spirtes et al., 2000; Pearl, 2009a; Van der Laan and Rose, 2018; Glymour et al., 2019). Many of the questions that drive scientific progress are not associational in nature, but are rather about identifying causal relations and the laws that govern them. Causal understanding is important to many applications like development of medical treatments (Imai and Van Dyk, 2004), medical imaging (Castro et al., 2020), advertising (Bottou et al., 2013), econometrics (Heckman, 2008), genetics (Murphy and Mian, 1999) psychology (Foster, 2010; Grosz et al., 2020), sociology (Gangl, 2010), policy making (Kreif and DiazOrdaz, 2019), machine learning (Schölkopf, 2019; Goyal et al., 2020; Teshima et al., 2020; Wu and Fukumizu, 2020; Yang et al., 2020) and many others.

Randomized controlled trials are the gold standard for determining causal relationships: by altering and manipulating some aspects of a system, we can examine how the rest of its features react. These trials allow researchers to acquire a causal understanding of the underlying mechanisms as well as an estimate of the magnitude of the causal relationships. They may also be used to predict the effect of explicit manipulations (*interventions*) to a system or to answer questions about what would happen if we did things differently to the naturally occurring process (*counterfactuals*). Unfortunately, such studies are often prohibitively expensive to carry out, or they may pose certain ethical concerns (Spirtes and Zhang, 2016). As a result, it is critical to create mathematical tools and procedures for performing *causal discovery* (uncovering causal relations) and *causal inference* (intervening on certain aspects of the data and answering counterfactual inquiries) from raw *observational data*. Such data is collected from trials that the researcher has no control over because of the aforementioned ethical or cost constraints.

1.3.1 Structural equation models

The framework of structural equation models[4] (SEMs) (Bollen, 1989) is a mathematical tool that can be used to encapsulate causal knowledge, as well as answer interventional and counterfactual queries (Pearl, 2009b). Fundamentally, SEMs define a generative model that describes the interactions of a set of observations $\mathbf{X} = (X_1, \dots, X_d)$ with a set of mutually independent disturbances or noise variables $\mathbf{N} = (N_1, \dots, N_d)$. They consist of a collection of equations of the form

$$X_j = f_j(\mathbf{PA}_j, N_j), \quad j = 1, \dots, d, \tag{1.17}$$

where $\mathbf{PA}_j \subseteq \{X_1, \dots, X_d\} \setminus X_j$ are called the parents of the variable X_j. An SEM is often associated with a directed acyclic graph (DAG) $\mathcal{G}$ called the *causal graph*. Each node of $\mathcal{G}$ corresponds to an observed variable X_j, and the edges are drawn from each parent to its direct effects. SEMs are very powerful: not only do they describe the set of all distributions, they can be used to perform interventions and answer counterfactual queries by changing the noise distribution or the causal mechanism in one or more of the equations (1.17).

[4]Structural Equation Models (SEM) are sometimes referred to as Structural Causal Models (SCM) or Functional Causal Models (FCM).

SEM are only useful for causal discovery if they define an *identifiable* causal model. A causal model is said to be identifiable if we can distinguish between cause and effect. In fact, the SEM (1.17) is a probabilistic model over the variables $(\mathbf{X}, \mathbf{N})$, parametrized by the causal functions $(f_1, \ldots, f_d)$ and the noise distribution. Under the identifiability assumption, two SEMs that define the same observational distribution over $\mathbf{X}$ have the same parameters. In particular, they define the same causal ordering over the variables $(X_1, \ldots, X_d)$. As an illustration, consider the bivariate case, where we only have two observations X and Y. If we assume that X is the cause of Y, then the causal model is identifiable if a function that expresses X in terms of the hypothetical cause Y and an independent error term cannot be found within the limitations of the SEM. Unfortunately, without restrictions, the causal direction of the general SEM (1.17) is not identifiable (Zhang et al., 2015a). The source of this unidentifiability is the same as in nonlinear ICA: general nonlinear mappings provide far too much flexibility, which implies that we may always represent any variable as the cause of another. In fact, the proof of unidentifiability of general SEMs is based on the unidentifiability theory of nonlinear ICA (Hyvärinen and Pajunen, 1999).

In order to accomplish identifiability, causal discovery algorithms generally adopt one of two techniques. The first approach is to impose constraints on the functions f_j that define the SEM (1.17), such as linear models with non-Gaussian noise (Shimizu et al., 2006; Shimizu et al., 2011; Lacerda et al., 2012; Hyvärinen and Smith, 2013; Zheng et al., 2018), nonlinear additive noise models (ANM) (Hoyer et al., 2009; Peters et al., 2014; Bloebaum et al., 2018), or post-nonlinear models (PNL) (Zhang and Hyvärinen, 2009). The second approach is to consider unconstrained SEMs while imposing constraints on the distribution of the disturbances. Such approaches frequently impose non-stationarity limitations on the distribution of the latent disturbances (Peters et al., 2016; Monti et al., 2019), or make assumptions about the existence of exogenous factors (Zhang et al., 2017). Generally, causal discovery research can be split into two main axes: finding constraints under which an SEM is identifiable, and proposing methods for the estimation of said SEM.

The remainder of this section will focus on *nonlinear* causal models, as they are the most relevant to this thesis. We will begin by giving a brief overview of the most notable identifiable nonlinear causal models, then discuss the several

estimation methods that were proposed in the literature.

1.3.2 Identifiable nonlinear causal models

In this subsection, we will review three of the most notable identifiable nonlinear causal models. The additive noise model (ANM) (Hoyer et al., 2009) and the post-nonlinear model (PNL) (Zhang and Hyvärinen, 2009) restrict the form of the causal functions in the SEM to achieve identifiability, whereas the non-stationary nonlinear SEM model (NonSENS) (Peters et al., 2016; Monti et al., 2019) achieves it by positing the non-stationarity of the latent disturbances.

Additive noise model (ANM). Hoyer et al. (2009) introduced the additive noise model, in which the SEM has the form

$$X_j = f_j(\mathbf{PA}_j) + N_j,$$

and the noise variables $\mathbf{N}$ are both mutually independent and independent of $\mathbf{X}$. Their theoretical identifiability results focuses on the case of two variables X_1 and X_2. It stipulates that if $X_1 \to X_2$, then we can't write $X_1 = g(X_2) + \tilde{N}$ for some function g and noise $\tilde{N} \perp\!\!\!\perp X_2$ that is independent of X_2. Essentially, the SEM is asymmetrical with respect to X_1 and X_2 and can only describe the natural cause-effect relationship. In other words, it is identifiable. Peters et al. (2014) generalized the identifiability result to the case of more than two variables.

Post-nonlinear model (PNL). Zhang and Hyvärinen (2009) introduced the post-nonlinear model, which generalizes ANM by adding a subsequent invertible mapping g_j:

$$X_j = g_j(f_j(\mathbf{PA}_j) + N_j).$$

The noise variables $\mathbf{N}$ are still assumed to be mutually independent and independent of the causes. The authors show that the bivariate PNL model is identifiable in most cases and enumerate five special situations in which the model is not identifiable. This identifiability theory generalizes that of ANM, which is a special case when g_j is the identity mapping. Note that we can reduce the PNL model to an ANM if we transform the effect through

the inverse of the mapping g_j, the transformed variable $g_j^{-1}(X_j)$ being totally correlated with the original effect X_j.

Non-stationary nonlinear SEM model (NonSENS). The two causal models discussed above attained identifiability by restricting the functional form of the causal mechanisms in the SEM. An alternative approach that leads to identifiability is to introduce constraints on the data distribution. Peters et al. (2016) and Monti et al. (2019) assume that the data is collected across a range of experimental settings $e \in \mathcal{E}$, which is effectively a form of non-stationarity:

$$X_j^e = f_j(\mathbf{PA}_j^e, N_j^e), \quad \forall e \in \mathcal{E}.$$

The causal mechanisms f_j are assumed to be the same for all the experimental settings.

Peters et al. (2016) focused on the case of linear models for which they proved identifiability and robustness to model misspecification. Monti et al. (2019) introduced the name NonSENS, and considered general nonlinear relationships between cause, noise and effect. Under the assumption of non-stationarity, they show that the causal model is identifiable even in such a general case by leveraging recent results in the theory of nonlinear ICA (Hyvärinen and Morioka, 2016; Hyvärinen et al., 2019).

1.3.3 Estimation methods for causal discovery

The works discussed in the previous section, as well as others (Hyvärinen and Smith, 2013; Peters et al., 2014; Bloebaum et al., 2018), also proposed estimation methods for causal discovery. This section provides a quick overview of these three models and the accompanying estimating methods, discussing their strengths and weaknesses.

Regression with subsequent independence tests (RESIT). Hoyer et al. (2009) proposed a constraint-based estimation method for *bivariate* additive noise models. A generalization to multivariate models as well as the name RESIT was proposed by Peters et al. (2014). This approach requires least-squares regressions in both directions $X_2 \to X_1$ and $X_1 \to X_2$, essentially learning an estimate of the independent disturbances N_1 and N_2. RESIT then

proceeds with a series of independence tests between the residuals and the causes. If $N_1 \perp\!\!\!\perp X_2$, then we can deduce from the identifiability property that the causal direction is $X_2 \to X_1$, and similarly if $N_1 \perp\!\!\!\perp X_1$. If both independence tests fail or succeed with the same confidence, we cannot conclude on the causal direction.

Choosing a suitable regression model for RESIT is not trivial. On the one hand, a simple model might not capture the causal relation in the right direction well. On the other hand, a complex model is prone to overfitting in the wrong direction, resulting in a more independent residual.

Furthermore, Hoyer et al. (2009) only proposed a method for causal discovery, with no discussion of the ability to answer interventional and counterfactual queries from RESIT. Peters et al. (2014) also do not claim the ability to predict counterfactual statements.

Finally, RESIT is a *constraint-based* method that is especially dependent on its underlying assumptions, in particular on the assumption of *faithfulness* which will typically not be satisfied in the presence of latent *confounders*[5]. This means that if the additive noise assumption fails, RESIT will also fail, regardless of regression class. Peters et al. (2014) proposed an an alternative *score-based* estimation method that does not require independence tests.

Regression-error based causal inference (RECI). Bloebaum et al. (2018) proposed an alternative method for estimating the causal direction in bivariate additive noise models. It is also based on least-squares regressions in both directions $X_2 \to X_1$ and $X_1 \to X_2$. However, unlike RESIT, RECI compares the magnitudes of the *residuals* to each other to deduce the causal direction. They rely on the fact that the expected variance of the effect given the cause is lower than that of the cause given the effect. The direction with the smaller residual magnitude corresponds to the true causal ordering.

Choosing a suitable regression model for RECI is also not trivial. As stated by Bloebaum et al. (2018), a very flexible regression class can reduce the performance of RECI because it is prone to overfitting. A simple regression function, on the other hand, will not be able to explain all the variance in the data, resulting in residuals that do not reflect the true causal direction. Importantly,

[5]A confounder is a hidden variable that affects both dependent and independent variables, resulting in spurious associations in the causal graph.

if the additive noise assumption fails, RECI, regardless of regression class, will fail.

Contrary to what the name of the method might suggest, and similarly to Hoyer et al. (2009) and Peters et al. (2014), the authors focused their efforts on developing a causal discovery method and did not discuss causal inference (interventions and counterfactuals).

Likelihood ratio measure of causal direction. Given the vulnerability of constraint-based methods, one can look to explore score-based approaches. Such methods do not depend on independence tests but instead seek to perform model comparison based on adequately defined score functions. For example, Hyvärinen and Smith (2013) considered the *likelihood ratio* as a score function and used it to measure the causal direction between a pair of variables (X_1, X_2). Briefly, one would need to compute the likelihood under two candidate models: $X_1 \to X_2$ and $X_2 \to X_1$. Then, the log-likelihood ratio is defined as the difference in log-likelihoods under the two models:

$$R = L_{1\to 2} - L_{2\to 1}. \tag{1.18}$$

We conclude that the correct causal direction is $X_1 \to X_2$ if R is positive, and $X_2 \to X_1$ instead.

We should highlight that this likelihood ratio measure of causal direction was initially created for LiNGAM, a linear model based on non-Gaussianity (Shimizu et al., 2006). Hyvärinen and Smith (2013) suggested an extension of likelihood ratios to ANM, as well as a heuristic approximation that roughly amounts to RECI. Monti et al. (2019) also proposed an extension of likelihood ratios to nonlinear non-stationary causal models.

Other estimation methods. LiNGAM (Shimizu et al., 2006; Shimizu et al., 2011) and NO-TEARS (Zheng et al., 2018) are methods designed for linear causal models. LiNGAM estimates a linear acyclic SEM by solving a linear ICA problem based on the assumption of non-Gaussianity. It is identifiable due to the identifiability of the equivalent ICA model. In contrast, NO-TEARS attempts to estimate a linear Gaussian SEM by solving a non-combinatorial constrained optimization problem. The identifiability theory is thus significantly weaker. The adjacency matrix and its inverse can then be used to perform

interventions and answer counterfactual queries. Monti et al. (2019) proposed an estimation method of the NonSENS model based on solving nonlinear ICA, akin to Shimizu et al. (2006).

1.4 Contributions and structure of the thesis

In this section, we start with a brief summary of the outline and contributions of this thesis, before going into details about the main contributions of Chapters 2 to 5. The list of publications upon which these chapters are based is given at the end of the section.

1.4.1 Structure and contributions in brief

The first three chapters of this thesis are dedicated to the identifiability theory of deep probabilistic models. We provide sufficient conditions under which the representations learned by very broad families of models are unique up to trivial ambiguities. In Chapter 2, we begin by establishing a principled connection between variational autoencoders and an identifiable nonlinear ICA model. Our identifiability theory borrows the idea of auxiliary variables from Hyvärinen et al. (2019) and extends it by incorporating noisy observations and undercomplete representations. Then, in Chapter 3, we present a novel framework called Independently Modulated Component Analysis (IMCA), which generalizes nonlinear ICA to allow for non-independent latent variables while retaining identifiability. This is accomplished by assuming the presence of an auxiliary variable that modulates the latent distribution. In Chapter 4, we develop a large family of conditional energy-based models that incorporates feature extractors to learn identifiable representations. The conditioning variable plays a similar role to the auxiliary variable in Hyvärinen et al. (2019), and the identifiability results apply to overcomplete representations while needing relatively few assumptions.

Finally, in Chapter 5, we work towards developing a new identifiable causal model derived from affine autoregressive normalizing flows (Rezende and Mohamed, 2015; Huang et al., 2018), which are intrinsically connected to SEMs. This affine causal model generalizes the additive noise model (Hoyer et al., 2009) by adding a cause-dependent coefficient to the noise variable in the SEM.

In addition, we show that autoregressive normalizing flows can be used to build a novel approach for causal discovery based on likelihood ratios. The resulting framework can also answer interventional and counterfactual queries thanks to the invertibility of normalizing flows.

1.4.2 Detailed contributions

Contributions of Chapter 2. Inspired by the prevalence of variational autoencoders in disentangled representation learning, this chapter aims to bring the identifiability guarantees of nonlinear ICA into the realm of VAEs. More specifically, we borrow the concept of auxiliary variables from Hyvärinen et al. (2019), and combine it with a VAE to define a generative model where posteriors are non-degenerate:

$$p(\mathbf{z}|\mathbf{u}) = \prod_i p(z_i|\mathbf{u}),$$
$$\mathbf{x} = \mathbf{f}(\mathbf{z}) + \boldsymbol{\varepsilon}.$$

The auxiliary variable $\mathbf{u}$ controls the prior distribution but is independent of $\mathbf{x}$ given the latent variable $\mathbf{z}$. The resulting framework, called iVAE or identifiable VAE, might look similar to semi-supervised learning methods in the VAE context due to the inclusion of the auxiliary variable $\mathbf{u}$. However, the auxiliary variable $\mathbf{u}$ can play a more general role (Hyvärinen et al., 2019). For instance, in time series, it can simply be the time index or history; in audiovisual data, it can be either one of the modalities, where the other is used as an observation. More importantly, and to our knowledge, there is no proof of identifiability in the semi-supervised literature. The following points summarize the contributions of the chapter:

(i) **Show the unidentifiability of nonlinear latent variable models with any prior:** The unidentifiability of nonlinear ICA has been known for decades now (Hyvärinen and Pajunen, 1999). Locatello et al. (2019) rediscovered this result when investigating disentangled representation learning. Both works show that it is impossible to recover the original components after a nonlinear mixing when using a factorized prior. In this chapter, we expand this result and demonstrate that it holds for *any prior*, including non-factorizing ones. This showcases that this problem

is fundamentally ill-defined and that using inductive biases is necessary for identifiability.

(ii) **Provide an estimation method of nonlinear ICA that is based on maximum likelihood estimation (MLE):** Using a VAE to estimate the latent sources offers several benefits over the self-supervised heuristics previously employed in nonlinear ICA (Hyvärinen and Morioka, 2016; Hyvärinen et al., 2019). Maximum likelihood estimation is robust to some failure modes which occur in the context of self-supervised methods, as we demonstrate in a series of experiments where the surrogate classification task in the self-supervised methods fails, resulting in learning the wrong features. Furthermore, the ELBO objective may be used for model selection and validation. Finally, we prove a tight link between maximum likelihood estimation and the maximization of independence of the latent components via total correlation (Watanabe, 1960). Note that, unlike prior work on disentanglement, we do not introduce hyperparameters to penalize the different terms of the ELBO. Instead, we compute the standard evidence lower bound to the log-likelihood $\log p(\mathbf{x}|\mathbf{u})$.

(iii) **Prove the identifiability of a large family with conditional exponential family prior:** Hyvärinen and Morioka (2016) proved identifiability up to a linear transformation of the latent components when the prior distribution is an exponential family. Hyvärinen et al. (2019) proved a stronger identifiability up to component-wise transformation by considering more general prior distributions that fulfil some relatively strict regularity constraints. In this chapter, we choose to focus on the conditional exponential family (1.16) for $p(z_i|\mathbf{u})$, because they lead to simpler assumptions. However, we prove that the model has the stronger identifiability, effectively combining the best of Hyvärinen and Morioka (2016) and Hyvärinen et al. (2019). We further prove that the case $k = 1$, considered only weakly identifiable by Hyvärinen et al. (2019), also benefits from the stronger identifiability guarantees if the sufficient statistics $q_{i,1}$ are monotonic.

(iv) **Generalize the nonlinear ICA model to noisy and undercomplete observations:** The theory of nonlinear ICA assumes that the

observations are a noise-free and invertible transformation of the latent components: $\mathbf{x} = \mathbf{f}(\mathbf{z})$ for a bijective $\mathbf{f}$. This is not very realistic because noise is ubiquitous in practice and the number of factors of variation is often assumed to be much smaller than the dimension of the data. In this chapter, we propose to change the generative model to $\mathbf{x} = \mathbf{f}(\mathbf{z}) + \boldsymbol{\varepsilon}$, where $\boldsymbol{\varepsilon}$ is a noise variable, and $\mathbf{f}$ is only supposed to be injective, and show that the identifiability theory of nonlinear ICA holds for this model. Most importantly, the estimation method developed here is the first that can estimate a model in the presence of noise and with a smaller number of components.

Contributions of Chapter 3. This chapter explores the identifiability of latent variable models in which the latent components are not necessarily independent. We introduce the framework of independently modulated component analysis (IMCA), in which the latent variables have a distribution of the form

$$\log p(\mathbf{z}|\mathbf{u}) = \log q_0(\mathbf{z}) + \sum_i \mathbf{q}_i(z_i)^\top \boldsymbol{\lambda}_i(\mathbf{u}) - Z(\mathbf{u}), \tag{1.19}$$

where q_0 is a base measure that is not necessarily factorized. The critical assumption for identifiability is the use of an auxiliary variable like in Hyvärinen et al. (2019), which independently modulates the distributions of the latent components. The main contributions of this chapter are:

(i) **Prove the identifiability of a model without independent latent components:** Identifiability is often studied within the context of latent variable models with independent components like ICA. Even in linear models, very few works study identifiability outside of this assumption (Hyvärinen and Hurri, 2004; Monti and Hyvärinen, 2018). The IMCA framework introduced above generalizes nonlinear ICA by allowing the latent components to have a dependence through the base measure q_0. We first prove that the identifiability guarantees developed for nonlinear ICA in Chapter 2 also hold for IMCA, which results in a more general framework for principled learning of identifiable representations. Second, we prove that we can further drop the assumption of independent modulation, while maintaining a weaker form of identifiability up to linear transformation.

(ii) **Show that some of the estimation methods developed for nonlinear ICA also work for IMCA:** Nonlinear ICA can be estimated through self-supervised schemes (Hyvärinen and Morioka, 2016; Hyvärinen and Morioka, 2017; Hyvärinen et al., 2019) or maximum likelihood estimation (Chapter 2). In this chapter, we show that both these approaches can be used for the estimation of IMCA.

Contributions of Chapter 4. Representation learning methods, including disentanglement methods and nonlinear ICA, could be divided into two broad classes: *generative approaches* and *feature extraction approaches*. The methods discussed so far are generative: we posit the existence of latent variables that are at the origin of the observed data. The goal is to invert the generative process and recover the original value of these latent factors. In this chapter, we develop an identifiable feature extraction approach for representation learning. To this end, we introduce conditional energy-based models that incorporate feature extractors to learn latent representations. More specifically, denote by $\mathbf{x}$ an observed variable, and $\mathbf{u}$ a conditioning variable. The model family has a density of the form

$$p(\mathbf{x}|\mathbf{u}) \propto \exp\left(-\mathbf{f}(\mathbf{x})^\top \mathbf{g}(\mathbf{u})\right). \tag{1.20}$$

The functions $\mathbf{f}$ and $\mathbf{g}$ are feature extractors used to learn representations from the observations and the conditioning variable, respectively. The conditioning variable $\mathbf{u}$ here plays a similar role to the auxiliary variable in iVAE (Chapter 2). This model, which we call identifiable conditional energy-based deep model, or ICE-BeeM for short, benefits from the tremendous flexibility and generality of energy-based models (EBMs) and drops all assumptions on the learned representations. The following points summarize the contributions of this chapter:

(i) **Prove the identifiability of a family of conditional energy-based models:** Identifiability is defined differently in feature extraction based approaches. Since there is no latent variable that is explicitly modelled, we express identifiability in terms of the similarity between two representations learned by two different models (1.20) from the same dataset. We show in this chapter that our conditional energy-based model has two types of identifiability. The first is weak identifiability akin to that of

Hyvärinen and Morioka (2016), where two representations are equal up to a linear transformation: $\mathbf{f}(\mathbf{x}) = \mathbf{A}\tilde{\mathbf{f}}(\mathbf{x}) + \mathbf{b}$. In applications where the representations are used in a downstream classification task, this weak identifiability form is often sufficient. The second is a stronger identifiability akin to that of Hyvärinen et al. (2019), where two representations are equal up a permutation and a scaling: $f_i(\mathbf{x}) = a\tilde{f}_{i'}(\mathbf{x}) + b$. The identifiability results developed in this chapter make fewer assumptions than earlier work on nonlinear ICA, all the while being stronger.

(ii) **Extend the identifiability results to overcomplete representations:** In Chapter 2, we developed a framework for nonlinear ICA which is capable of learning identifiable undercomplete representations. In this chapter, we extend the weak identifiability up to linear scaling to overcomplete representations, where $\dim(\mathbf{f}(\mathbf{x})) \geq \dim(\mathbf{x})$. This is the first such identifiability result in the nonlinear setting. Moreover, the energy-based model (1.20) can be shown to have universal approximation capability if the dimension of the feature extractor is not constrained. This reinforces the importance of extending identifiability to overcomplete representations.

(iii) **Prove the identifiability of a neural network architecture:** The recent theory of identifiability focused on providing functional conditions for identifiability in the abstraction of the network architecture. While this makes these results more general, such works are a bit removed from the reality of neural network training. In this chapter, we translate the functional identifiability assumptions into a set of constraints on the architecture of a multilayer perceptron (MLP). This is the first step towards bridging the gap between theory and practice.

(iv) **Propose a novel method for the estimation of identifiable latent variable models:** The identifiability of the conditional energy-based model makes it a prime candidate for the estimation of identifiable latent variable models. Combined with its flexibility, it gives ICE-BeeM an edge over previous methods like iVAE (Chapter 2) and TCL (Hyvärinen and Morioka, 2016) in learning the original components in nonlinear ICA and IMCA (Chapter 3). More specifically, under the IMCA model (1.19), the

likelihood can be written as $\log p(\mathbf{x}|\mathbf{u}) = \log p(\mathbf{z}|\mathbf{u}) + h(\mathbf{x})$, where h is a function that groups all the terms that only depends on $\mathbf{x}$. Assume that ICE-BeeM has learned the log-probability density function exactly. Then, by equating the IMCA likelihood (1.19) to equation (1.20), we have, purely heuristically, that $f_i(\mathbf{x}) = q_i(z_i)$, which means that the feature extractor recovers the latent variables up to component-wise nonlinear functions. This is made rigorous in the chapter.

(v) **Large scale experimental validation on image datasets:** The chapter validates the theoretical identifiability findings by comparing multiple representations learned from image datasets for different random initialisations. The benefits of identifiability are then explored in two applications: transfer learning and semi-supervised learning, in which ICE-BeeM comes up on top of the baselines. We conclude this chapter by showing that our method is competitive against state-of-the-art methods in learning the components of nonlinear ICA and IMCA models.

Contributions of Chapter 5. In this chapter, we prove the identifiability of a new causal model that generalizes additive noise models. The new model is inspired by a family of variational inference models called normalizing flows. Normalizing flows define the density of observations $\mathbf{X}$ as a series of invertible transformations of a latent variable $\mathbf{N}$ with a simpler distribution. Autoregressive normalizing flows (Dinh et al., 2014; Dinh et al., 2016; Huang et al., 2018) use transformations of the form

$$X_j = \tau_j(N_j; \mathbf{PA}_j), \tag{1.21}$$

where $\mathbf{PA}_j$ are the variables that precede X_j in the autoregressive ordering. Equation (1.21) is very similar to an SEM (1.17), suggesting that autoregressive normalizing flows have an intrinsic causal ordering. This makes the resulting framework, which we call causal autoregressive flows (CAREFL) an excellent candidate to perform causal discovery and inference, as summarized by the following contributions:

(i) **Prove the identifiability of a new affine causal model:** Autoregressive normalizing flows are known for their flexibility and expressivity.

To derive an identifiable causal model from an autoregressive flow, we restrict the transformer τ_j in equation (1.21) to be an affine function of the noise, with intercept and bias parametrized by the causes:

$$X_j = f_j(\mathbf{PA}_j) + g_j(\mathbf{PA}_j)N_j, \tag{1.22}$$

where g_j is a positive function, and f_j is arbitrary. In this chapter, we prove an identifiability result of this model in the bivariate case. In brief, assume that observations (X_1, X_2) follow the model $X_2 = f_2(X_1) + g_2(X_1)N_2$. If N_2 and X_1 statistically independent and at least one of them is Gaussian, and f_2 is invertible and nonlinear, then the inverse model $X_1 = f_1(X_2) + g_1(X_2)N_1$ for a Gaussian N_1 cannot hold. In our model, in stark contrast to the PNL model, it is not possible to apply a fixed (as in not a function of the cause) transformation to the effect to revert to an additive noise model. This chapter thus presents a novel identifiability result in the context of non-additive noise models, which complements that of the ANM and PNL models.

(ii) **Develop a new nonlinear measure of causal direction based on normalizing flows:** One of the strengths of normalizing flows is that they allow for easy evaluation of the likelihood. In this chapter, we leverage this property to propose a measure of causal direction based on likelihood ratios, extending the work of Hyvärinen and Smith (2013) to nonlinear SEMs. We approach bivariate causal discovery as a model selection problem and compare two candidate models: $X_1 \to X_2$ against $X_1 \leftarrow X_2$. The likelihood ratio is defined analogously to equation (1.18) as $R = \log L_{1\to 2} - \log L_{2\to 1}$. One of the main challenges encountered when computing the likelihood, and more generally the ratio R is that often times we have to evaluate a log-determinant of a Jacobian, which is not trivial. This is why the likelihood ratio measure has been primarily used in the context of linear SEMs. Autoregressive flows are designed specifically to make the evaluation of Jacobian terms easy and are thus well suited to obtain a nonlinear measure of causal direction. Finally, unlike in RESIT (Hoyer et al., 2009; Peters et al., 2014) and RECI (Bloebaum et al., 2018), CAREFL doesn't suffer from having very flexible classes of functions in equation (1.22), which gives it an edge over these methods, even for the estimation of ANM.

(iii) **Show that normalizing flows are well suited for interventional and counterfactual queries:** Another valuable property of normalizing flows is that they can be easily inverted. In this chapter, we leverage this property to evaluate interventional and counterfactual statements using autoregressive flows. These often require inverting the SEM to update the distribution of the disturbances according to a counterfactual observation.

1.4.3 Publications

The four main chapters of this thesis are based on the following publications. Source code for all proposed methods is publicly available.

1. Chapter 2: Variational autoencoders and nonlinear ICA

   I. Khemakhem, D.P. Kingma, R.P. Monti, A. Hyvärinen. "Variational Autoencoders and Nonlinear ICA: A Unifying Framework". In AISTATS, 2020.

   Code: https://github.com/ilkhem/ivae

2. Chapter 3: Independently modulated component analysis and Chapter 4: Identifiable conditional energy-based models

   I. Khemakhem, R.P. Monti, D.P. Kingma, A. Hyvärinen. "ICE-BeeM: Identifiable Conditional Energy-Based Deep Models". In NeurIPS, 2020. (Spotlight presentation)

   Code: https://github.com/ilkhem/icebeem

3. Chapter 5: Causal autoregressive flows

   I. Khemakhem, R.P. Monti, R. Leech, A. Hyvärinen. "Causal Autoregressive Flows". In AISTATS, 2021.

   Code: https://github.com/ilkhem/carefl

1.5 Notation and terminology

All through the thesis, we follow a general set of principles for the mathematical notations:

- We use a capital letter in boldface, for example $\mathbf{A}$, to denote matrices.
- Vectors are denoted by lower-case letters in boldface, for example $\mathbf{v}$. This also applies to vector-valued functions. A notable exception to this rule is the sufficient statistic $\mathbf{T}$ of an exponential family (or any function that plays a similar role), which is a vector.
- Indexing the components of a vector is done with a subscript notation.
- Indexing the elements of a dataset is done with a superscript notation between brackets.

The notation and terminology employed throughout this thesis is as follows:

$\mathbb{R}^{m\times n}$: Set of real-valued $m \times n$ matrices

$\det \mathbf{A}$: Determinant of matrix $\mathbf{A}$

$\operatorname{vol} \mathbf{A}$: Volume of matrix $\mathbf{A}$ — $\operatorname{vol} \mathbf{A} \stackrel{\text{def}}{=} \sqrt{\det(\mathbf{A}^\top \mathbf{A})}$

$\dim(\mathbf{v})$: Dimension of vector $\mathbf{v}$

$\operatorname{rank}(\mathbf{M})$: Rank of matrix $\mathbf{M}$

$\operatorname{diag} \mathbf{v}$: Square matrix whose diagonal entries are the elements of $\mathbf{v}$

$\operatorname{span}(S)$: Linear span or linear hull of a set of vectors S

$\mathbf{J}_\mathbf{f}$: Jacobian matrix of a vector-valued function $\mathbf{f}$

∇g : Gradient of a scalar function g

$\langle \cdot, \cdot \rangle$: Dot product

$\{\ldots\}$: Set of objects

$[a, b]$: Real interval

$[\![a, b]\!]$: Integer set — $[\![a, b]\!] \stackrel{\text{def}}{=} [a, b] \cap \mathbb{Z}$

$\mathbb{E}[\,\cdot\,]$: Expectation of an event

$\mathbb{P}(A)$: Probability of an event A

$\mathbf{x} \sim p$: Sample $\mathbf{x}$ from probability distribution with density p

μ_{Leb} : Lebesgue measure

$|\cdot|$: Absolute value

$\|\cdot\|$: Norm

$\mathcal{C}(\mathcal{X})$: Set of continuous functions on $\mathcal{X}$

$\mathfrak{S}_n$: Set of permutations of $[\![1, n]\!]$

$\mathrm{KL}(\cdot \,\|\, \cdot)$: Kullback-Leibler divergence

$F[\cdot]$: Fourier transform

Appendices to Chapter 1

1.A Statistical independence

Statistical independence is a key concept in the theory of independent component analysis and one of the primary assumptions we will make for the identifiable models provided in this thesis. In this section, we briefly review the definition of statistical independence, as well as a useful corollary of independence.

Definition 1.3 (Statistical independence)**.** *Let $X_1, \dots, X_n$ be n random variables on $\mathcal{X}$, where $n \geq 2$. These random variables are said to be mutually independent (or simply independent) if for any functions $h_1, \dots, h_n$ on $\mathcal{X}$ such that $\mathbb{E}[|h_i(X_i)|] < +\infty$, the following holds:*

$$\mathbb{E}\left[\prod_i h_i(X_i)\right] = \prod_i \mathbb{E}\left[h_i(X_i)\right]. \tag{1.23}$$

If all the X_i are continuous random variables, then condition (1.23) is equivalent to:

$$p_{\mathbf{X}}(x_1, \dots, x_n) = \prod_i p_{X_i}(x_i).$$

Proposition 1.4. *Let X_1 and X_2 be two independent random variables, such that $\mathbb{E}\left[|X_1|^2\right] < +\infty$ and $\mathbb{E}\left[|X_2|^2\right] < +\infty$. Then their covariance is equal to zero:* $\operatorname{cov}(X_1, X_2) = 0$.

Proof. This is an immediate corollary from equation (1.23) when h_1 and h_2 are the identity function, since $\operatorname{cov}(X_1, X_2) = \mathbb{E}\left[X_1 X_2\right] - \mathbb{E}\left[X_1\right]\mathbb{E}\left[X_2\right]$. □

1.B Exponential family

In this section, we briefly review the exponential family, which is a popular parametric set of probability distributions.

Let $\mathcal{P}_\Theta = \{p_{\boldsymbol{\theta}}, \boldsymbol{\theta} \in \Theta\}$ be a probabilistic model, where Θ is the parameter space.

Definition 1.5 (Sufficient statistic)**.** *A function* $\mathbf{T}$ *is called a sufficient statistic for a model* $\mathcal{P}_\Theta$ *if and only if*

$$\forall \boldsymbol{\theta} \in \Theta, \quad p_{\boldsymbol{\theta}}(\mathbf{x}) = h(\mathbf{x}) g(\mathbf{T}(\mathbf{x}); \boldsymbol{\theta})$$

for some scalar function h *and* g*.*

In other words, the statistic $\mathbf{T}$ contains all the information needed for the maximum likelihood estimation of $\boldsymbol{\theta}$ from a dataset of observations.

Definition 1.6 (Exponential family)**.** *Let* $\mathbf{x}$ *be a random variable on* $\mathcal{X} \subset \mathbb{R}^d$*. The exponential family is a parametric set of distributions whose probability density function can be written as*

$$p_{\boldsymbol{\theta}}(\mathbf{x}) = \frac{Q(\mathbf{x})}{Z(\boldsymbol{\lambda})} e^{\langle \mathbf{T}(\mathbf{x}), \boldsymbol{\lambda}(\boldsymbol{\theta}) \rangle} = Q(\mathbf{x}) e^{\langle \mathbf{T}(\mathbf{x}), \boldsymbol{\lambda}(\boldsymbol{\theta}) \rangle - \Gamma(\boldsymbol{\lambda})}, \tag{1.24}$$

where

- $\mathbf{T} : \mathbb{R}^d \to \mathbb{R}^k$ *is called the sufficient statistic,*
- $\boldsymbol{\theta} \in \Theta$ *is the parameter,*
- $\boldsymbol{\lambda} \in \mathbb{R}^k$ *is the natural parameter,*
- $Q : \mathbb{R}^d \to \mathbb{R}$ *is the base measure,*
- $Z(\boldsymbol{\lambda})$ *is the normalization constant*
- $\Gamma(\boldsymbol{\lambda}) = \log Z(\boldsymbol{\lambda})$ *is the log-partition function.*

To alleviate notations, we will drop the reference to $\boldsymbol{\theta}$ when it is clear from context which parameter we refer to. The form of equation (1.24) is not unique. We can add terms to $\mathbf{T}(\mathbf{x})$ and $\boldsymbol{\lambda}(\boldsymbol{\theta})$ that are constant in $\boldsymbol{\theta}$, which can then be offset in $Q(\mathbf{x})$ outside of the exponential. In the remainder of this thesis, the dimension $k \in \mathbb{N} \setminus \{0\}$ of the sufficient statistic and natural parameter is always considered to be *minimal*, meaning that we can't rewrite the density $p_{\boldsymbol{\theta}}$ to have the form (1.24) with a smaller $k' < k$. We call k the size of p.

A very natural corollary follows from this assumption:

Lemma 1.7. *Consider an exponential family distribution with* $k \geq 2$ *components. If there exists* $\alpha \in \mathbb{R}^k$ *such that* $T_k(\mathbf{x}) = \sum_{i=1}^{k-1} \alpha_i T_i(\mathbf{x}) + \alpha_k$*, then*

$\boldsymbol{\alpha} = 0$. *In particular, the components of the sufficient statistic* $\mathbf{T}$ *are linearly independent.*

Proof. Suppose the components $(T_1, \dots, T_k)$ of the sufficient statistic are not linearly independent. Then $\exists \boldsymbol{\alpha} \in \mathbb{R}^k \setminus \{0\}$ such that $\forall \mathbf{x} \in \mathbb{R}^d$, $\sum_{i=1}^k \alpha_i T_i(\mathbf{x}) = 0$. Suppose $\alpha_k \neq 0$ (up to rearrangement of the indices), then we can write T_k as a function of the remaining $T_i, i < k$, contradicting the minimality of k. □

The exponential family is comprehensive and includes many of the most common distributions used in statistics, including the Gaussian, gamma, Wishart, exponential and Bernoulli, to name a few.

Example 1.8 (Gaussian distribution)**.** The univariate Gaussian distribution with parameters $\boldsymbol{\theta} = (\mu, \sigma^2)$ is part of the exponential family. Its density has the form

$$p_{\boldsymbol{\theta}}(x) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{(x-\mu)^2}{2\sigma^2}\right),$$

which can be written in the form (1.24) with $\mathbf{T}(x) = (x, x^2)$, $Q(x) = 1/2\pi$ and $\boldsymbol{\lambda} = \left(\frac{\mu}{\sigma^2}, -\frac{1}{2\sigma^2}\right)$. □

Example 1.9 (Laplace distribution)**.** The Laplace distribution with mean μ and scale b has a density of the form

$$p_{\boldsymbol{\theta}}(x) = \frac{1}{2b} \exp\left(\frac{-|x-\mu|}{b}\right).$$

Despite the presence of an exponential in its density, the Laplace distribution with unknown mean and scale is not an exponential family distribution. This is because, unlike the squaring in the density of a Gaussian variable, the absolute value cannot be expanded into a dot-product of terms that either depend on x or $\boldsymbol{\theta} = (\mu, b)$. However, when the mean μ is fixed, the Laplace distribution becomes part of the exponential family with sufficient statistic $T(x) = |x - \mu|$ and natural parameter $\lambda = -\frac{1}{b}$. □

1.B.1 Conditional exponential family

Given two random variables $\mathbf{x}$ and $\mathbf{y}$ on $\mathcal{X} \subset \mathbb{R}^d$ and $\mathcal{Y} \subset \mathbb{R}^m$ respectively, we propose to extend Definition 1.6 to a family of conditional densities $p_{\boldsymbol{\theta}}(\mathbf{x}|\mathbf{y})$. We modify equation (1.24) by making the natural parameter $\boldsymbol{\lambda}$ a function of the conditioning variable $\mathbf{y}$.

Definition 1.10 (Conditional exponential family)**.** *Let* $(\mathbf{x}, \mathbf{y})$ *be a random variable on* $\mathcal{X} \times \mathcal{Y} \subset \mathbb{R}^d \times \mathbb{R}^m$*. The conditional exponential family of* $\mathbf{x}$ *given* $\mathbf{y}$ *has a density of the form*

$$p_{\boldsymbol{\theta}}(\mathbf{x}|\mathbf{y}) = Q(\mathbf{x})e^{\langle \mathbf{T}(\mathbf{x}), \boldsymbol{\lambda}(\mathbf{y})\rangle - \Gamma(\mathbf{y})}, \tag{1.25}$$

where the natural parameter $\boldsymbol{\lambda} : \mathcal{Y} \to \mathbb{R}^k$ *is a function of the conditioning variable* $\mathbf{y}$*, and* $\Gamma(\mathbf{y}) \coloneqq \Gamma(\boldsymbol{\lambda}(\mathbf{y}))$*.*

1.B.2 Exponential family and independence

Let $\mathbf{x}$ be a d-dimensional random vector that belongs to the exponential family. When its components are independent — i.e. $x_i \perp\!\!\!\perp x_j, \forall i \neq j$ — the density (1.24) takes a factorial form:

$$\begin{aligned} p_{\boldsymbol{\theta}}(\mathbf{x}) &= \prod_{i=1}^{d} p_i(x_i), \\ p_i(x_i) &= Q_i(x_i)e^{\langle \mathbf{T}_i(x_i), \boldsymbol{\lambda}_i\rangle - \Gamma_i(\boldsymbol{\lambda}_i)}, \end{aligned}$$

where each of the p_i is an exponential family of size k_i. The joint density $p_{\boldsymbol{\theta}}$ is also an exponential family of size k such that:

$$\begin{aligned} Q(\mathbf{x}) &= \prod_{i=1}^{d} Q_i(x_i), \\ \mathbf{T}(\mathbf{x}) &= (\mathbf{T}_1(x_1), \dots, \mathbf{T}_d(x_d)), \\ \boldsymbol{\lambda} &= (\boldsymbol{\lambda}_1, \dots, \boldsymbol{\lambda}_d), \\ k &= \sum_{i=1}^{d} k_i. \end{aligned}$$

If the components of $\mathbf{x}$ are only independent given a conditioning variable $\mathbf{y}$, then the same decomposition holds for the conditional exponential family (1.25):

$$\begin{aligned} p(\mathbf{x}|\mathbf{y}) &= \prod_{i=1}^{d} p_i(x_i|\mathbf{y}), \\ p_i(x_i|\mathbf{y}) &= Q_i(x_i)e^{\langle \mathbf{T}_i(x_i), \boldsymbol{\lambda}_i(\mathbf{y})\rangle - \Gamma_i(\mathbf{y})}. \end{aligned}$$

Chapter 2

Variational autoencoders and nonlinear ICA

The framework of variational autoencoders allows us to efficiently learn deep latent variable models, such that the model's marginal distribution over observed variables fits the data. Often, we are interested in going a step further and want to approximate the true joint distribution over observed and latent variables, including the true prior and posterior distributions over latent variables. This is known to be generally impossible due to the unidentifiability of the model. We address this issue by showing that for a broad family of deep latent variable models, identifying the true joint distribution over observed and latent variables is actually possible up to very simple transformations, thus achieving a principled and powerful form of disentanglement. Our result requires a factorized prior distribution over the latent variables that is conditioned on an additionally observed variable, such as a class label or almost any other observation. We build on recent developments in nonlinear ICA, which we extend to the case with noisy or undercomplete observations integrated into a maximum likelihood framework. The result also trivially contains identifiable flow-based generative models as a particular case.

This chapter is based on Khemakhem et al. (2020a).

2.1 Introduction

The framework of variational autoencoders (VAEs, Kingma and Welling, 2014; Rezende et al., 2014) and its extensions (e.g. Burda et al., 2015; Kingma et al., 2016; Tucker et al., 2018; Maaløe et al., 2019) offers a scalable set of techniques for learning deep latent variable models and corresponding inference models. The theory behind VAEs tells us how they can be optimised towards an objective function that corresponds to a lower bound of the marginal likelihood of the data, also called the evidence lower bound (ELBO). With VAEs, we can, in principle, learn flexible models of data such that, after optimisation, the model's implicit marginal distribution over the observed variables approximates their true (but unknown) distribution. We can also efficiently synthesise pseudo-data from the model.

However, we are often interested in going further and want to learn the true joint distribution over observed and latent variables. This is generally a challenging task since, by definition, we only ever observe the observed variables, never the latent variables; therefore, we cannot directly estimate their joint distribution. However, if we could somehow achieve this task and learn the true joint distribution, this would imply that we have also learned to approximate the true prior and posterior distributions over latent variables. Learning about these distributions can be very interesting for various purposes, for example, learning about the latent structure behind the data or infer the latent variables from which the data originated. Such inference is potentially useful for various downstream tasks.

Learning the true joint distribution is only possible when the model is *identifiable*, as we will explain. The original VAE theory does not tell us when this is the case; it only tells us how to optimise the model's parameters such that its (marginal) distribution over the observed variables matches the data. The original theory does not tell us if or when we learn the correct joint distribution over observed and latent variables.

Almost no literature exists on achieving this goal. A pocket of the VAE literature works towards the related goal of *disentanglement*, but offers no proofs or theoretic guarantees of identifiability of the model or its variables. The most prominent of such models are β-VAEs and their extensions (Higgins et al., 2017; Burgess et al., 2018; Chen et al., 2018; Higgins et al., 2018;

Kim and Mnih, 2018; Esmaeili et al., 2019), in which the authors introduce adjustable hyperparameters in the VAE objective to encourage disentanglement. Other work attempts to find maximally independent components through the GAN framework (Brakel and Bengio, 2017). However, models in these earlier works are unidentifiable due to non-conditional latent priors, as has been seen empirically (Locatello et al., 2019), and as we will show formally below.

Recent work in nonlinear Independent Component Analysis (ICA) theory (Hyvärinen and Morioka, 2016; Hyvärinen and Morioka, 2017; Hyvärinen et al., 2019) provided the first identifiability results for deep latent variable models. Nonlinear ICA provides a rigorous framework for recovering independent latent variables that were transformed by some invertible nonlinear transformation into the data. Some special but not very restrictive conditions are necessary since it is known that when the function from latent to observed variables is nonlinear, the general problem is ill-posed, and one cannot recover the independent latent variables (Hyvärinen and Pajunen, 1999). However, existing nonlinear ICA methods do not learn to model the data distribution (pdf), nor do they allow us to synthesise pseudo-data.

In this chapter, we show that the joint distribution over observed and latent variables in VAEs is identifiable and learnable under relatively mild conditions, thus bridging the gap between VAEs and nonlinear ICA. To this end, we establish a principled connection between VAEs and an identifiable nonlinear ICA model, providing a unified view of two complementary methods in unsupervised representation learning. This integration is achieved by using a latent prior that has a factorised distribution that is conditioned on additionally observed variables, such as a class label, time index, or almost any other further observation. Our theoretical results trivially apply to any consistent parameter estimation method for deep latent variable models, not just the VAE framework. We found the VAE a logical choice since it allows for efficient latent variable inference and scales to large datasets and models.

Finally, we put our theoretical results to the test in experiments. Perhaps most notably, we find that on a synthetic dataset with a known ground-truth model, our method with an identifiable VAE indeed learns to closely approximate the true joint distribution over observed and latent variables, in contrast with a baseline non-identifiable model. It also improves the performance of nonlinear ICA-based causal discovery methods.

2.2 Unidentifiability of deep latent variable models

Consider an observed data variable (random vector) $\mathbf{x} \in \mathbb{R}^d$, and a latent random vector $\mathbf{z} \in \mathbb{R}^n$. A common deep latent variable model has the following structure:

$$p_{\boldsymbol{\theta}}(\mathbf{x}, \mathbf{z}) = p_{\boldsymbol{\theta}}(\mathbf{x}|\mathbf{z})p_{\boldsymbol{\theta}}(\mathbf{z}), \tag{2.1}$$

where $\boldsymbol{\theta} \in \Theta$ is a vector of parameters, $p_{\boldsymbol{\theta}}(\mathbf{z})$ is called a *prior* distribution over the latent variables. The distribution $p_{\boldsymbol{\theta}}(\mathbf{x}|\mathbf{z})$, often parametrized with a neural network called the *decoder*, tells us how the distribution on $\mathbf{x}$ depends on the values of $\mathbf{z}$. The model then gives rise to the observed distribution of the data as:

$$p_{\boldsymbol{\theta}}(\mathbf{x}) = \int p_{\boldsymbol{\theta}}(\mathbf{x}, \mathbf{z}) \mathrm{d}\mathbf{z}. \tag{2.2}$$

Assuming $p_{\boldsymbol{\theta}}(\mathbf{x}|\mathbf{z})$ is modelled by a deep neural network, this can model a rich class of data distributions $p_{\boldsymbol{\theta}}(\mathbf{x})$.

We assume that we observe data which is generated from an underlying joint distribution $p_{\boldsymbol{\theta}^*}(\mathbf{x}, \mathbf{z}) = p_{\boldsymbol{\theta}^*}(\mathbf{x}|\mathbf{z})p_{\boldsymbol{\theta}^*}(\mathbf{z})$ where $\boldsymbol{\theta}^*$ are its true but unknown parameters. We then collect a dataset of observations of $\mathbf{x}$:

$$\mathcal{D} = \{\mathbf{x}^{(1)}, \ldots, \mathbf{x}^{(N)}\} \text{ where } \begin{aligned} \mathbf{z}^{*(i)} &\sim p_{\boldsymbol{\theta}^*}(\mathbf{z}), \\ \mathbf{x}^{(i)} &\sim p_{\boldsymbol{\theta}^*}(\mathbf{x}|\mathbf{z}^{*(i)}). \end{aligned}$$

Note that the original values $\mathbf{z}^{*(i)}$ of the latent variables $\mathbf{z}$ are by definition not observed and unknown. The ICA literature, including this chapter, uses the term *sources* to refer to $\mathbf{z}^{*(i)}$. Also note that we could just as well have written: $\mathbf{x}^{(i)} \sim p_{\boldsymbol{\theta}^*}(\mathbf{x})$.

The VAE framework (Kingma and Welling, 2014; Rezende et al., 2014) allows us to efficiently optimize the parameters $\boldsymbol{\theta}$ of such models towards the (approximate) maximum marginal likelihood objective, such that after optimization:

$$p_{\boldsymbol{\theta}}(\mathbf{x}) \approx p_{\boldsymbol{\theta}^*}(\mathbf{x}). \tag{2.3}$$

In other words, after optimization, we have then estimated the marginal density of $\mathbf{x}$.

Remark 2.1 (Parameter Space vs Function Space)**.** In this chapter, we use slightly non-standard notation and nomenclature: we use $\boldsymbol{\theta} \in \Theta$ to refer to the model parameters in *function space*. In contrast, let $\mathbf{w} \in \mathcal{W}$ refer to the space of original neural network parameters (weights and biases) in which we usually perform gradient ascent.

The VAE model actually learns a full generative model $p_{\boldsymbol{\theta}}(\mathbf{x}, \mathbf{z})$ and an inference model $q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x})$ that approximates its posterior $p_{\boldsymbol{\theta}}(\mathbf{z}|\mathbf{x})$. The problem is that we generally have no guarantees about what these learned distributions actually are: all we know is that the marginal distribution over $\mathbf{x}$ is meaningful (2.3). The rest of the learned distributions are, generally, quite meaningless.

What we are looking for is models for which the following implication holds for all $(\mathbf{x}, \mathbf{z})$:

$$\forall(\boldsymbol{\theta}, \boldsymbol{\theta}') : \quad p_{\boldsymbol{\theta}}(\mathbf{x}) = p_{\boldsymbol{\theta}'}(\mathbf{x}) \quad \Longrightarrow \quad \boldsymbol{\theta} = \boldsymbol{\theta}'. \tag{2.4}$$

That is: if any two different choices of model parameter $\boldsymbol{\theta}$ and $\boldsymbol{\theta}'$ lead to the same marginal density $p_{\boldsymbol{\theta}}(\mathbf{x})$, then this would imply that they are equal and thus have matching joint distributions $p_{\boldsymbol{\theta}}(\mathbf{x}, \mathbf{z})$. This means that if we learn a parameter $\boldsymbol{\theta}$ that fits the data perfectly: $p_{\boldsymbol{\theta}}(\mathbf{x}) = p_{\boldsymbol{\theta}^*}(\mathbf{x})$, which is the ideal case of equation (2.3)), then its joint density also matches perfectly: $p_{\boldsymbol{\theta}}(\mathbf{x}, \mathbf{z}) = p_{\boldsymbol{\theta}^*}(\mathbf{x}, \mathbf{z})$. If the joint density matches, this also means that we found the correct prior $p_{\boldsymbol{\theta}}(\mathbf{z}) = p_{\boldsymbol{\theta}^*}(\mathbf{z})$ and correct posteriors $p_{\boldsymbol{\theta}}(\mathbf{z}|\mathbf{x}) = p_{\boldsymbol{\theta}^*}(\mathbf{z}|\mathbf{x})$. In case of VAEs, we can then also use the inference model $q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x})$ to efficiently perform inference over the sources $\mathbf{z}^*$ from which the data originates.

The general problem here is a lack of *identifiability guarantees* of the deep latent variable model. We illustrate this by showing that any model with unconditional latent distribution $p_{\boldsymbol{\theta}}(\mathbf{z})$ is unidentifiable, i.e. that equation (2.4) does not hold. In this case, we can always find transformations of $\mathbf{z}$ that change its value but not its distribution. For a spherical Gaussian distribution $p_{\boldsymbol{\theta}}(\mathbf{z})$, for example, applying a rotation keeps its distribution the same. We can then incorporate this transformation as the first operation in $p_{\boldsymbol{\theta}}(\mathbf{x}|\mathbf{z})$. This will not change $p_{\boldsymbol{\theta}}(\mathbf{x})$, but it will change $p_{\boldsymbol{\theta}}(\mathbf{z}|\mathbf{x})$, since now the values of $\mathbf{x}$ come from different values of $\mathbf{z}$. This is an example of a broad class of commonly used models that are non-identifiable. We show rigorously in Appendix 2.C that, in fact, models with *any* form of unconditional prior $p_{\boldsymbol{\theta}}(\mathbf{z})$ are unidentifiable.

2.3 An identifiable model based on conditionally factorial priors

This section defines a broad family of deep latent variable, identifiable models and shows how to estimate the model and its posterior through the VAE framework. We call this family of models, together with its estimation method, identifiable VAE, or iVAE for short.

2.3.1 Definition of proposed model

The primary assumption leading to identifiability is a conditionally factorized prior distribution over the latent variables $p_{\boldsymbol{\theta}}(\mathbf{z}|\mathbf{u})$, where $\mathbf{u}$ is an additionally observed variable (Hyvärinen et al., 2019). The variable $\mathbf{u}$ could be, for example, the time index in a time series (Hyvärinen and Morioka, 2016), previous data points in a time series, a (possibly noisy) class label, or another concurrently observed variable.

Formally, let $\mathbf{x} \in \mathbb{R}^d$, and $\mathbf{u} \in \mathbb{R}^m$ be two observed random variables, and $\mathbf{z} \in \mathbb{R}^n$ (lower-dimensional, $n \leq d$) a latent variable. Let $\boldsymbol{\theta} = (\mathbf{f}, \mathbf{T}, \boldsymbol{\lambda})$ be the parameters of the following conditional generative model:

$$p_{\boldsymbol{\theta}}(\mathbf{x}, \mathbf{z}|\mathbf{u}) = p_{\mathbf{f}}(\mathbf{x}|\mathbf{z}) p_{\mathbf{T},\boldsymbol{\lambda}}(\mathbf{z}|\mathbf{u}), \tag{2.5}$$

where we first define

$$p_{\mathbf{f}}(\mathbf{x}|\mathbf{z}) = p_{\boldsymbol{\varepsilon}}(\mathbf{x} - \mathbf{f}(\mathbf{z})), \tag{2.6}$$

which means that the value of $\mathbf{x}$ can be decomposed as $\mathbf{x} = \mathbf{f}(\mathbf{z}) + \boldsymbol{\varepsilon}$ where $\boldsymbol{\varepsilon}$ is an independent noise variable with probability density function $p_{\boldsymbol{\varepsilon}}(\boldsymbol{\varepsilon})$, i.e. $\boldsymbol{\varepsilon}$ is independent of $\mathbf{z}$ or $\mathbf{f}$. We assume that the function $\mathbf{f} : \mathbb{R}^n \to \mathbb{R}^d$ is *injective*; but it can be an arbitrarily complicated nonlinear function apart from injectivity. For the sake of analysis, we treat the function $\mathbf{f}$ itself as a model parameter; however, in practice, we can use flexible function approximators such as neural networks.

Remark 2.2 (Discrete observations)**.** We can use a well-known logistic model to replace the additive Gaussian noise in equation (2.6) to model discrete

observations. For example, in the binary case, let:

$$\mathbf{m} = \text{sigmoid}(\mathbf{f}(\mathbf{z})), \tag{2.7}$$

$$\mathbf{x} \sim \text{Bernoulli}(\mathbf{m}), \tag{2.8}$$

where sigmoid() is the element-wise sigmoid nonlinearity. However, the mapping $\mathbf{z} \to \mathbf{x}$ can no longer be injective by the very nature of discrete variables. This is one of the key assumptions in our identifiability theory, which can no longer hold. The discrete observation case requires a bespoke identifiability proof. Nevertheless, we provide experiments in Section 2.5.2 which strongly suggest that identifiability is achievable in such a setting.

We describe the model above with noisy and continuous-valued observations $\mathbf{x} = \mathbf{f}(\mathbf{z}) + \boldsymbol{\varepsilon}$. However, our identifiability results also apply to non-noisy observations $\mathbf{x} = \mathbf{f}(\mathbf{z})$, which are a special case of equation (2.6) where $p_{\boldsymbol{\varepsilon}}(\boldsymbol{\varepsilon})$ is Gaussian with infinitesimal variance. For these reasons, we can use flow-based generative models (Dinh et al., 2014) for $p_{\boldsymbol{\theta}}(\mathbf{x}|\mathbf{z})$, while maintaining identifiability.

The prior on the latent variables $p_{\boldsymbol{\theta}}(\mathbf{z}|\mathbf{u})$ is assumed to be *conditionally* factorial, where each element of $z_i \in \mathbf{z}$ has a univariate exponential family distribution given conditioning variable $\mathbf{u}$. The conditioning on $\mathbf{u}$ is through an arbitrary function $\boldsymbol{\lambda}(\mathbf{u})$ (such as a look-up table or neural network) that outputs the individual exponential family parameters $\lambda_{i,j}$. The probability density function is thus given by:

$$p_{\mathbf{T},\boldsymbol{\lambda}}(\mathbf{z}|\mathbf{u}) = \prod_i \frac{Q_i(z_i)}{Z_i(\mathbf{u})} \exp\left[\sum_{j=1}^{k} T_{i,j}(z_i)\lambda_{i,j}(\mathbf{u})\right], \tag{2.9}$$

where Q_i is the base measure, $Z_i(\mathbf{u})$ is the normalizing constant and $\mathbf{T}_i = (T_{i,1}, \ldots, T_{i,k})$ are the sufficient statistics and $\boldsymbol{\lambda}_i(\mathbf{u}) = (\lambda_{i,1}(\mathbf{u}), \ldots, \lambda_{i,k}(\mathbf{u}))$ the corresponding parameters, crucially depending on $\mathbf{u}$. Finally, k, the dimension of each sufficient statistic, is fixed (not estimated). Note that exponential families have universal approximation capabilities, so this assumption is not very restrictive (Sriperumbudur et al., 2017).

2.3.2 Estimation by VAE

Next, we propose a practical estimation method for the model introduced above. Consider we have a dataset $\mathcal{D} = \left\{\left(\mathbf{x}^{(1)}, \mathbf{u}^{(1)}\right), \ldots, \left(\mathbf{x}^{(N)}, \mathbf{u}^{(N)}\right)\right\}$ of observations

generated according to the generative model defined in equation (2.5). We propose to use a VAE as a means of learning the true generating parameters $\boldsymbol{\theta}^* := (\mathbf{f}^*, \mathbf{T}^*, \boldsymbol{\lambda}^*)$, up to the indeterminacies discussed below.

VAEs are a framework that simultaneously learns a deep latent generative model and a variational approximation $q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x}, \mathbf{u})$ of its true posterior $p_{\boldsymbol{\theta}}(\mathbf{z}|\mathbf{x}, \mathbf{u})$, the latter being often intractable. Denote by

$$p_{\boldsymbol{\theta}}(\mathbf{x}|\mathbf{u}) = \int p_{\boldsymbol{\theta}}(\mathbf{x}, \mathbf{z}|\mathbf{u}) \mathrm{d}\mathbf{z}$$

the conditional marginal distribution of the observations, and with $q_{\mathcal{D}}(\mathbf{x}, \mathbf{u})$ we denote the empirical data distribution given by dataset $\mathcal{D}$. VAEs learn the vector of parameters $(\boldsymbol{\theta}, \boldsymbol{\phi})$ by maximizing $\mathcal{L}(\boldsymbol{\theta}, \boldsymbol{\phi})$, a lower bound on the data log-likelihood defined by:

$$\begin{aligned} &\mathbb{E}_{q_{\mathcal{D}}}\left[\log p_{\boldsymbol{\theta}}(\mathbf{x}|\mathbf{u})\right] \geq \\ &\qquad \mathcal{L}(\boldsymbol{\theta}, \boldsymbol{\phi}) := \mathbb{E}_{q_{\mathcal{D}}}\left[\mathbb{E}_{q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x},\mathbf{u})}\left[\log p_{\boldsymbol{\theta}}(\mathbf{x}, \mathbf{z}|\mathbf{u}) - \log q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x}, \mathbf{u})\right]\right]. \end{aligned} \tag{2.10}$$

We use the reparametrization trick (Kingma and Welling, 2014) to sample from $q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x}, \mathbf{u})$. This trick provides a low-variance stochastic estimator for gradients of the lower bound with respect to $\boldsymbol{\phi}$. The training algorithm is the same as in a regular VAE. Estimates of the latent variables can be obtained by sampling from the variational posterior.

VAEs, like any maximum likelihood estimation method, requires the densities to be normalized. To this end, in practice, we choose the prior $p_{\boldsymbol{\theta}}(\mathbf{z}|\mathbf{u})$ to be a Gaussian location-scale family, which is widely used with VAEs.

Remark 2.3. As mentioned in Section 2.3.1, our model contains normalizing flows as a special case when the variance $\mathrm{Var}(\boldsymbol{\varepsilon}) = 0$ and the mixing function $\mathbf{f}$ is parametrized as an invertible flow (Rezende and Mohamed, 2015). Thus, as an alternative estimation method, we could then optimize the log-likelihood directly:

$$\mathbb{E}_{q_{\mathcal{D}}(\mathbf{x},\mathbf{u})}[\log p_{\boldsymbol{\theta}}(\mathbf{x}|\mathbf{u})] = \mathbb{E}_{q_{\mathcal{D}}(\mathbf{x},\mathbf{u})}\left[\log p_{\boldsymbol{\theta}}(\mathbf{f}^{-1}(\mathbf{z})|\mathbf{u}) + \log|\mathbf{J}_{\mathbf{f}^{-1}}(\mathbf{x})|\right],$$

where $\mathbf{J}_{\mathbf{f}^{-1}}$ is easy to compute. The conclusion on consistency given in Section 2.4.5 still holds in this case. This approach was studied in subsequent work by Sorrenson et al. (2020).

2.3.3 Identifiability and consistency results

As discussed in Section 2.2, identifiability as defined by equation (2.4) is very hard to achieve in deep latent variable models. As a first step towards an identifiable model, we seek to recover the model parameters or the latent variables up to trivial transformations. Here, we state our results informally on this weaker form of identifiability of the model —a rigorous treatment is given in Section 2.4. Consider for simplicity the case of no noise and sufficient statistics of size $k = 1$, and define $T_i \coloneqq T_{i,1}$. Then we can recover $\mathbf{z}$ which are related to the original $\mathbf{z}^*$ as follows:

$$(T_1^*(z_1^*), \ldots, T_n^*(z_n^*)) = \mathbf{A}(T_1(z_1), \ldots, T_n(z_n)) \tag{2.11}$$

for an invertible matrix $\mathbf{A}$. That is, we can recover the original latent variables up to component-wise (pointwise) transformations T_i^* and T_i, which are defined as the sufficient statistics of exponential families, and up to a subsequent linear transformation $\mathbf{A}$. Importantly, the linear transformation $\mathbf{A}$ can often be resolved by excluding families where, roughly speaking, only the location (mean) is changing. Then $\mathbf{A}$ is simply a permutation matrix, and equation (2.11) becomes

$$T_i^*(z_i^*) = T_{i'}(z_{i'}) \tag{2.12}$$

for a permuted index i'. Thus, the only real indeterminacy is often the component-wise transformations of the latent variables, which is a fundamental indeterminacy of nonlinear ICA, and may be inconsequential in many applications.

2.3.4 Interpretation as nonlinear ICA

Now we show how the model above is closely related to previous work on nonlinear ICA. In nonlinear ICA, we assume observations $\mathbf{x} \in \mathbb{R}^d$, which are the result of an unknown (but invertible) transformation $\mathbf{f}$ of latent variables $\mathbf{z} \in \mathbb{R}^d$:

$$\mathbf{x} = \mathbf{f}(\mathbf{z}), \tag{2.13}$$

where $\mathbf{z}$ are assumed to follow a factorized (but typically unknown) distribution $p(\mathbf{z}) = \prod_{i=1}^d p_i(z_i)$. This model is essentially a deep generative model. The difference to the definition above is mainly in the lack of noise and the equality

of the dimensions: The transformation $\mathbf{f}$ is deterministic and invertible. Thus, any posteriors would be degenerate.

The goal is then to recover (identify) $\mathbf{f}^{-1}$, which gives the independent components as $\mathbf{z} = \mathbf{f}^{-1}(\mathbf{x})$, based on a dataset of observations of $\mathbf{x}$ alone. Thus, the goal of nonlinear ICA was always identifiability, which is in general not attained by deep latent variable models, as was discussed in Section 2.2 above.

To obtain identifiability, we either have to restrict $\mathbf{f}$ (for instance, make it linear) or introduce some additional constraints on the distribution of the sources $\mathbf{z}$. Recently, three new nonlinear ICA frameworks (Hyvärinen and Morioka, 2016; Hyvärinen and Morioka, 2017; Hyvärinen et al., 2019) exploring the latter direction were proposed, in which it is possible to recover identifiable sources, up to some trivial transformations.

The framework introduced by Hyvärinen et al. (2019) is particularly close to what we proposed above. However, there are several significant differences. First, here we define a generative model where posteriors are non-degenerate, which allows us to show an explicit connection to VAE. We are thus also able to perform maximum likelihood estimation in terms of evidence lower bound, while previous nonlinear ICA used more heuristic self-supervised schemes. Computing a lower bound on the likelihood is useful, for example, for model selection and validation. In addition, we can prove a tight link between maximum likelihood estimation and maximization of independence of latent variables, as discussed in Appendix 2.E. We also learn both the forward and backward models, which allows for recovering independent latent variables from data and generating new data. The forward model is also likely to help investigate the meaning of the latent variables. At the same time, we can provide stronger identifiability results that apply to more general models than recent theory. In particular, we consider the case where the number of latent variables is smaller than the number of observed variables and is corrupted by noise. Given the popularity of VAEs, our current framework should thus be of interest.

2.3.5 Relation to previous work on disentanglement

The iVAE framework might look similar to semi-supervised learning methods in the VAE context due to the inclusion of the auxiliary variable $\mathbf{u}$. However, the auxiliary variable $\mathbf{u}$ can play a more general role. For instance, in time

series, it can simply be the time index or history; in audiovisual data, it can be either one of the modalities, where the other is used as an observation. More importantly, there is no proof of identifiability in the semi-supervised literature.

The question of identifiability, or lack thereof, in deep latent variable models, especially VAEs, has been tackled in work related to disentanglement. In Mathieu et al. (2018), Rolinek et al. (2018), and Locatello et al. (2019) the authors show how isotropic priors lead to rotation invariance in the ELBO. We proved here (Section 2.2 and Appendix 2.C) a much more general result: unconditional priors lead to unidentifiable models. Unlike what this chapter sets to do, these works focused on showcasing this problem, or how it can be avoided in practice, and didn't provide alternative models that can be shown to be identifiable. The proof of identifiability presented in this chapter applies to the generative model itself, regardless of the estimation method. This is why the role of the encoder, which has been claimed to have a central role in some of the work cited above was not the focus of the analysis presented here.

2.4 Identifiability theory

Now we state the main technical results of this chapter. The proofs are in Appendix 2.B.

Notations. Let $\mathcal{Z} \subset \mathbb{R}^n$ and $\mathcal{X} \subset \mathbb{R}^d$ be the domain and the image of $\mathbf{f}$ in equation (2.6), respectively, and $\mathcal{U} \subset \mathbb{R}^m$ the support of the distribution of $\mathbf{u}$. We denote by $\mathbf{f}^{-1}$ the inverse defined from $\mathcal{X} \to \mathcal{Z}$. We suppose that $\mathcal{Z}$, $\mathcal{X}$ and $\mathcal{U}$ are open sets. We denote by $\mathbf{T}(\mathbf{z}) \coloneqq (\mathbf{T}_1(z_1), \dots, \mathbf{T}_n(z_n)) = (T_{1,1}(z_1) \dots, T_{n,k}(z_n)) \in \mathbb{R}^{nk}$ the vector of sufficient statistics of (2.9), $\boldsymbol{\lambda}(\mathbf{u}) = (\boldsymbol{\lambda}_1(\mathbf{u}), \dots, \boldsymbol{\lambda}_n(\mathbf{u})) = (\lambda_{1,1}(\mathbf{u}), \dots, \lambda_{n,k}(\mathbf{u})) \in \mathbb{R}^{nk}$ the vector of its parameters. Finally $\Theta = \{\boldsymbol{\theta} \coloneqq (\mathbf{f}, \mathbf{T}, \boldsymbol{\lambda})\}$ is the domain of parameters describing equation (2.5).

2.4.1 Identifiability up to equivalence class

Traditionally, a probabilistic model $\mathcal{P} = \{p_{\boldsymbol{\theta}} : \boldsymbol{\theta} \in \Theta\}$ is said to be identifiable if the mapping $\boldsymbol{\theta} \mapsto p_{\boldsymbol{\theta}}$ is bijective, *i.e.*

$$p_{\boldsymbol{\theta}_1} = p_{\boldsymbol{\theta}_2} \implies \boldsymbol{\theta}_1 = \boldsymbol{\theta}_2. \tag{2.14}$$

This implication is too strong and often impractical for many statistical models. For example, as discussed in Section 1.2.2.1, linear ICA is only identifiable up to scaling and permutation of the independent components. Neural networks are a more contemporary example for which this definition is too restrictive: it is well known that there is a many-to-one mapping from the space of weights and biases parametrizing the network to its function space.

We can relax the definition (2.14) by only requiring the parameters $\boldsymbol{\theta}_1$ and $\boldsymbol{\theta}_2$ to be "equivalent" to each other. An equivalence relation is a binary relation between pairs of elements of a set $\mathcal{X}$ and is defined as follows:

Definition 2.4 (Equivalence relation)**.** *A relation $\sim$ on a set $\mathcal{X}$ is called* an equivalence relation *if and only if it satisfies the following three properties:*

- *Reflexive: $x \sim x, \forall x \subset \mathcal{X}$.*
- *Transitive: $x \sim y$ and $y \sim z$ implies $x \sim z$, $\forall x, y, z \in \mathcal{X}$.*
- *Symmetric: $x \sim y$ implies $y \sim x$, $\forall x, y \in \mathcal{X}$.*

A classical example of equivalence relations is the modulo operation in arithmetic. For each $x \in \mathcal{X}$, the *equivalence class of* x denoted $[x]$ is defined as $[x] = \{y \in \mathcal{X} : y \sim x\}$, i.e. the set of all the elements of $\mathcal{X}$ which are $\sim$-related to x. The set of all equivalence classes induced by $\sim$ forms a partition of $\mathcal{X}$ into disjoint subsets. This partition, denoted $\mathcal{X}/_{\sim}$ is defined as $\mathcal{X}/_{\sim} = \{[x] : x \in \mathcal{X}\}$ is called the quotient set of $\mathcal{X}$ by $\sim$.

We can use equivalence relations to formalize the types of ambiguities for a given statistical model. This leads to the new definition of identifiability up to equivalence class:

Definition 2.5 (Identifiability up to equivalence class)**.** *Let $\mathcal{P} = \{\mathcal{P}_{\boldsymbol{\theta}} : \boldsymbol{\theta} \in \Theta\}$ be a probabilistic model, and let $\sim$ be an equivalence relation on Θ. We say that the model $\mathcal{P}$ is identifiable up to $\sim$ (or $\sim$-identifiable) if*

$$\mathcal{P}_{\boldsymbol{\theta}_1} = \mathcal{P}_{\boldsymbol{\theta}_2} \implies \boldsymbol{\theta}_1 \sim \boldsymbol{\theta}_2. \tag{2.15}$$

The elements of the quotient space $\Theta/_{\sim}$ are called the identifiability classes.

This definition can be met by any probabilistic model is if the equivalence relation is very broad. For example, if we define the equivalence relation on

the space of parameters of a VAE that define the same observed distribution, then by virtue of the unidentifiability (Section 2.2) and the ability of a VAE to approximate data densities very well, the parameter space will be constituted of very few classes of equivalence.

For Definition 2.5 to be non-vacuous, we need to carefully select an equivalence relation that only reflects the indeterminacies of the task at hand. In linear ICA for instance, the mixing matrix is uniquely recovered up to a scaled permutation. The permutation is irrelevant, and the scaling is circumvented by whitening the data. This means that linear ICA is not identifiable in the strictest sense. But if we consider two model parameters to be equivalent if they are equal up to permutation and scaling, linear ICA becomes identifiable up to this equivalence class. This example, as well as others of useful equivalence classes for identifiability, are discussed in more detail in Appendix 2.D.

For the purpose of this chapter, we define two equivalence relations on the set of parameters Θ of the model (2.5).

Definition 2.6. *Let $\sim$ be the equivalence relation on Θ defined as follows:*

$$(\mathbf{f}, \mathbf{T}, \boldsymbol{\lambda}) \sim (\tilde{\mathbf{f}}, \tilde{\mathbf{T}}, \tilde{\boldsymbol{\lambda}}) \iff \exists \mathbf{A}, \mathbf{c} \mid \mathbf{T}(\mathbf{f}^{-1}(\mathbf{x})) = \mathbf{A}\tilde{\mathbf{T}}(\tilde{\mathbf{f}}^{-1}(\mathbf{x})) + \mathbf{c}, \forall \mathbf{x} \in \mathcal{X}, \tag{2.16}$$

where $\mathbf{A}$ is an $nk \times nk$ matrix and $\mathbf{c}$ is a vector

If $\mathbf{A}$ is invertible*, we denote this relation by $\sim_{\mathbf{A}}$. If $\mathbf{A}$ is a* block permutation matrix*, where each block linearly transforms $\mathbf{T}_i$ into $\tilde{\mathbf{T}}_{i'}$, we denote it by $\sim_{\mathbf{P}}$.*

2.4.2 Strongly exponential family

The role of the auxiliary variable $\mathbf{u}$ in equation (2.9) is to control the distribution of the latent variable $\mathbf{z}$ through the natural parameter $\boldsymbol{\lambda} : \mathcal{U} \to \mathbb{R}^{nk}$. This modulation is crucial because it reduces the degrees of freedom of the latent distribution, making identifiability a more attainable goal.

To this end, we introduce the *strongly exponential family*, a subclass of the exponential family, defined as follows.

Definition 2.7 (Strongly exponential distributions)**.** *We say that an exponential family distribution is* strongly exponential *if for any subset $\mathcal{X} \subset \mathbb{R}^d$ the*

following is true:

$$\left(\exists\, \boldsymbol{\lambda} \in \mathbb{R}^k \,|\, \forall \mathbf{x} \in \mathcal{X}, \langle \mathbf{T}(\mathbf{x}), \boldsymbol{\lambda} \rangle = const\right) \implies (\mu_{Leb}(\mathcal{X}) = 0 \text{ or } \boldsymbol{\lambda} = 0)\,, \quad (2.17)$$

where μ_{Leb} is the Lebesgue measure.

In other words, the density of a strongly exponential distribution has almost surely the exponential component in its expression and can only be reduced to the base measure on a set of measure zero. Useful properties of the strongly exponential family are discussed in Appendix 2.A.

Example 2.8. The strongly exponential family is very general and includes most of the usual exponential family distributions like the Gaussian, Laplace, Pareto, Chi-squared, Gamma, Beta, *etc.*

On the other hand, consider an exponential family distribution whose density function is

$$p(x) = \frac{e^{-x^2}}{Z(\boldsymbol{\lambda})} \exp\left(\theta_1 \min(0, x) - \theta_2 \max(0, x)\right).$$

This density sums to 1, and $Z(\boldsymbol{\lambda})$ is well defined. Yet, for $x \in (-\infty, o)$ and for $\boldsymbol{\lambda} = (0, \lambda_2)$ for any value of λ_2, the dot-product $\mathbf{T}(x)^\top \boldsymbol{\lambda} = 0$, which means that p is not strongly exponential. □

2.4.3 General results

The following theorem gives the main identifiability result of this chapter. An alternative formulation is discussed in Appendix 2.B.2.

Theorem 2.9. *Assume that we observe data sampled from a generative model defined according to equations (2.5) to (2.9), with parameters $(\mathbf{f}, \mathbf{T}, \boldsymbol{\lambda})$. Assume the following holds:*

(i) The set $\{\mathbf{x} \in \mathcal{X} | \varphi_\varepsilon(\mathbf{x}) = 0\}$ has measure zero, where φ_ε is the characteristic function of the density p_ε defined in equation (2.6).

(ii) The mixing function $\mathbf{f}$ in equation (2.6) is injective.

(iii) The distributions of the independent latents are all strongly exponential (Definition 2.7).

(iv) There exist $nk+1$ *distinct points* $\mathbf{u}^0, \ldots, \mathbf{u}^{nk}$ *such that the matrix*

$$\mathbf{L} = (\boldsymbol{\lambda}(\mathbf{u}_1) - \boldsymbol{\lambda}(\mathbf{u}_0), \ldots, \boldsymbol{\lambda}(\mathbf{u}_{nk}) - \boldsymbol{\lambda}(\mathbf{u}_0)) \tag{2.18}$$

of size $nk \times nk$ *is invertible.*

then the parameters $(\mathbf{f}, \mathbf{T}, \boldsymbol{\lambda})$ *are* $\sim_{\mathbf{A}}$*-identifiable.*

This theorem guarantees a basic form of identifiability of the generative model (2.5). In fact, suppose the data was generated according to the set of parameters $(\mathbf{f}, \mathbf{T}, \boldsymbol{\lambda})$, and let $(\tilde{\mathbf{f}}, \tilde{\mathbf{T}}, \tilde{\boldsymbol{\lambda}})$ be the parameters obtained from some learning algorithm (supposed consistent in the limit of infinite data) that perfectly approximates the marginal distribution of the observations. Then the theorem says that necessarily $(\tilde{\mathbf{f}}, \tilde{\mathbf{T}}, \tilde{\boldsymbol{\lambda}}) \sim_{\mathbf{A}} (\mathbf{f}, \mathbf{T}, \boldsymbol{\lambda})$. If there were no noise, this would mean that the learned transformation $\tilde{\mathbf{f}}$ transforms the observations into latents $\tilde{\mathbf{z}} = \tilde{\mathbf{f}}^{-1}(\mathbf{x})$ that are equal to the true generative latents $\mathbf{z} = \mathbf{f}^{-1}(\mathbf{x})$, up to a linear invertible transformation (the matrix $\mathbf{A}$) and pointwise nonlinearities (in the form of $\mathbf{T}$ and $\tilde{\mathbf{T}}$). With noise, we obtain the posteriors of the latents up to an analogous indeterminacy.

Remark 2.10 (Understanding assumption (iv) in Theorem 2.9)**.** Let $\mathbf{u}^0$ be an arbitrary point in its support $\mathcal{U}$, and

$$\mathbf{h}(\mathbf{u}) = \left(\lambda_{1,1}(\mathbf{u}) - \lambda_{1,1}(\mathbf{u}^0), \ldots, \lambda_{n,k}(\mathbf{u}) - \lambda_{n,k}(\mathbf{u}^0)\right) \in \mathbb{R}^{nk}.$$

Saying that there exists nk distinct points $\mathbf{u}^1$ to $\mathbf{u}^{nk}$ (all different from $\mathbf{u}^0$) such that $\mathbf{L}$ is invertible is equivalent to saying that the vectors $(\mathbf{h}(\mathbf{u}^1), \ldots, \mathbf{h}(\mathbf{u}^{nk}))$ are linearly independent in $\mathbb{R}^{nk}$. Let us suppose for a second that for any such choice of points, these vectors are not linearly independent. This means that $\mathbf{h}(\mathcal{U})$ is necessarily included in a subspace of $\mathbb{R}^{nk}$ of dimension at most $nk-1$. Such a subspace has measure zero in $\mathbb{R}^{nk}$. Thus, if $\mathbf{h}(\mathcal{U})$ is not included in a subset of measure zero in $\mathbb{R}^{nk}$, this cannot be true, and there exists a set of points $\mathbf{u}^1$ to $\mathbf{u}^{nk}$ (all different from $\mathbf{u}^0$) such that $\mathbf{L}$ is invertible. This implies that as long as the $\lambda_{i,j}(\mathbf{u})$ are generated randomly and independently, then almost surely, $\mathbf{h}(\mathcal{U})$ will not be included in any such subset with measure zero, and the assumption holds.

We next give a simple example where this assumption always holds. Suppose $n = 2$ and $k = 1$, and that the auxiliary variable is a positive scalar.

Consider sources $z_i \sim \mathcal{N}(0, \lambda_i(u))$ that are distributed according to Gaussian distributions with zero mean and variances modulated as follows:

$$\lambda_1(u) = u,$$
$$\lambda_2(u) = u^2.$$

Because the functions $u \mapsto u$ and $u \mapsto u^2$ are linearly independent (as functions), then for any choice of "pivot" point u_0, for instance $u_0 = 1$, and any choice of distinct nonzero scalars u_1 and u_2, the columns of the matrix $\mathbf{L} \coloneqq (\boldsymbol{\lambda}(u_1) - 1, \boldsymbol{\lambda}(u_2) - 1)$ are linearly independent, and the matrix is invertible.

2.4.4 Characterization of the linear indeterminacy

The equivalence relation $\sim_{\mathbf{A}}$ provides a useful form of identifiability, but it is very desirable to remove the linear indeterminacy $\mathbf{A}$, and reduce the equivalence relation to $\sim_{\mathbf{P}}$ by analogy with linear ICA where such matrix is resolved up to a *permutation* and *signed scaling*. We present in this section sufficient conditions for such reduction and special cases to avoid.

We will start by giving two theorems that provide sufficient conditions. Theorem 2.11 deals with the more general case $k \geq 2$, while Theorem 2.12 deals with the special case $k = 1$.

Theorem 2.11 ($k \geq 2$)**.** *Assume the hypotheses of Theorem 2.9 hold, and that $k \geq 2$. Further assume:*

(i) The sufficient statistics $T_{i,j}$ in equation (2.9) are twice differentiable.

(ii) The mixing function $\mathbf{f}$ has all second order cross derivatives.

then the parameters $(\mathbf{f}, \mathbf{T}, \boldsymbol{\lambda})$ are $\sim_{\mathbf{P}}$-identifiable.

Theorem 2.12 ($k = 1$)**.** *Assume the hypotheses of Theorem 2.9 hold, and that $k = 1$. Further assume:*

(i) The sufficient statistics $T_{i,1}$ are not monotonic (strictly increasing or decreasing).

(ii) All partial derivatives of $\mathbf{f}$ are continuous.

then the parameters $(\mathbf{f}, \mathbf{T}, \boldsymbol{\lambda})$ are $\sim_{\mathbf{P}}$-identifiable.

These two theorems imply that in most cases $\tilde{\mathbf{f}}^{-1} \circ \mathbf{f} : \mathcal{Z} \to \mathcal{Z}$ is a pointwise nonlinearity, (*i.e.* each of its components is a function of only one z_i) which essentially means that the estimated latent variable $\tilde{\mathbf{z}}$ are equal to a permutation and a pointwise nonlinearity of the original latent variables $(z_1, \ldots, z_n)$. To the best of our knowledge, this kind of identifiability is stronger than any previous literature results (Hyvärinen and Morioka, 2016; Hyvärinen and Morioka, 2017; Hyvärinen et al., 2019) and is considered sufficient in many applications, like linear classification in a downstream task or for use in causal discovery.

There are very special cases where a linear indeterminacy cannot be resolved, as summarized by the following proposition.

Proposition 2.13. *Assume that $k = 1$, and that*

(i) $T_{i,1}(z_i) = z_i$ for all i.

(ii) $Q_i(z_i) = 1$ or $Q_i(z_i) = e^{-z_i^2}$ for all i.

Then $\mathbf{A}$ can not be reduced to a permutation matrix.

This proposition stipulates that if the components are Gaussian (or exponential in the case of non-negative components) and *only* the location is changing, we cannot hope to reduce the matrix $\mathbf{A}$ in $\sim_{\mathbf{A}}$ to a permutation. To prove this in the Gaussian case, we simply consider orthogonal transformations of the latent variables, which all give rise to the same observational distribution with a simple adjustment of parameters.

2.4.5 Consistency of estimation

The theory above further implies a consistency result on the VAE. If the variational distribution $q_{\boldsymbol{\phi}}$ is a broad parametric family that includes the true posterior, then we have the following result.

Theorem 2.14. *Assume the following:*

(i) The family of distributions $q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x}, \mathbf{u})$ contains $p_{\mathbf{f},\mathbf{T},\boldsymbol{\lambda}}(\mathbf{z}|\mathbf{x}, \mathbf{u})$.

(ii) We maximize $\mathcal{L}(\boldsymbol{\theta}, \boldsymbol{\phi})$ with respect to both $\boldsymbol{\theta}$ and $\boldsymbol{\phi}$.

then in the limit of infinite data, the VAE learns the true parameters $\boldsymbol{\theta}^* := (\mathbf{f}^*, \mathbf{T}^*, \boldsymbol{\lambda}^*)$ up to the equivalence class defined by $\sim$ in equation (2.16).

2.5 Experiments[1]

2.5.1 Mean correlation coefficient as a measure of identifiability

To measure the quality of a reconstruction $\mathbf{z}$ found by an ICA algorithm, we need to design a measure of identifiability that is invariant to the ambiguities of ICA.

To this end, let $\rho : \mathcal{X} \times \mathcal{Y} \to [0, 1]$ be a measure of dependence (also known as correlation coefficient) between two random variables $\mathbf{x} \in \mathcal{X}$ and $\mathbf{y} \in \mathcal{Y}$. We can use ρ to compare $z_i \in \mathbf{z}$ to a true source component $z_i^* \in \mathbf{z}^*$: if $\mathbf{z}$ is a perfect reconstruction of $\mathbf{z}^*$, then $\rho(z_i, z_i^*)$ should be equal to 1 for all i. In truth, however, there is a permutation and component-wise transformation indeterminacies between $\mathbf{z}$ and $\mathbf{z}^*$. In the case of linear ICA, the component-wise transformation is simply scaling by a scalar.

The invariance to component-wise transformation can be inherited from the measure of dependence ρ. For example. Pearson's correlation coefficient (Pearson and Galton, 1895) defined as

$$\rho_p(\mathbf{x}, \mathbf{y}) = \frac{|\mathrm{cov}(\mathbf{x}, \mathbf{y})|}{\sqrt{\mathrm{cov}(\mathbf{x}, \mathbf{x})\,\mathrm{cov}(\mathbf{y}, \mathbf{y})}}$$

is invariant to linear transformation of $\mathbf{x}$ and $\mathbf{y}$ which we can use to circumvent any scaling ambiguity.

To account for the permutation invariance, we can use the independence property of $\mathbf{z}$ and $\mathbf{z}^*$. In an ideal scenario, the correlation $\rho_p(z_i, z_j^*)$ should be zero for all j except for one. Thus, by computing all pairs of correlation coefficients between the components z_i of $\mathbf{z}$ and the components z_j^* of $\mathbf{z}^*$, we can find the optimal permutation by solving a linear sum assignment problem (Kuhn, 1955; Bertsekas, 1992). We subsequently average over all correlation coefficients to define the mean correlation coefficient (MCC) metric.

As an illustrative example, let $\mathbf{x} \in \mathbb{R}^2$ be a bivariate random variable such that $x_1 \perp\!\!\!\perp x_2$, and let $\mathbf{y} = (2x_2, \frac{x_1}{2})$. If we don't account for any permutations, then the average correlation is equal to $\frac{1}{2}\sum_i \rho_p(x_i, y_i) = 0$ because $x_1 \perp\!\!\!\perp x_2$. In reality though, $\mathbf{y}$ and $\mathbf{x}$ are perfectly correlated, since the value of $\mathbf{x}$ completely

[1] Code to reproduce the experiments is available at https://github.com/ilkhem/ivae.

determines that of $\mathbf{y}$. Thus, we have to find the optimal permutation of the elements of $\mathbf{y}$ in order to maximise the average correlation.

The MCC was used as performance metric in recent work on nonlinear ICA (Hyvärinen and Morioka, 2016; Hyvärinen et al., 2019). In this section, we introduce a new definition that formalises this measure.

Definition 2.15 (Mean correlation coefficient)**.** *Let $\rho : \mathcal{X} \times \mathcal{Y} \to [0, 1]$ be a measure of dependence. We define the mean correlation coefficient function as:*

$$\Psi_\rho(\mathbf{x}, \mathbf{y}) = \max_{\sigma \in \mathfrak{S}_d} \frac{1}{d} \sum_{i=1}^{d} \rho(x_i, y_{\sigma(i)}). \tag{2.19}$$

The MCC measure is invariant by permutation and component-wise transformations. The latter is limited to the class of invariances satisfied by the measure of dependence ρ. For example, Pearson's correlation coefficient ρ_p (Pearson and Galton, 1895) is invariant to linear transformations, which makes Ψ_{ρ_p} well adapted as an identifiability measure for linear ICA. Spearman's rank correlation coefficient ρ_r (Spearman, 1904) is invariant to monotonic transformations. To use the MCC as an identifiability measure for nonlinear ICA, it is best to combine it with a dependence measure ρ that is invariant to any pointwise nonlinearity. One such measure is the randomized dependence coefficient (Lopez-Paz et al., 2013, RDC) which satisfies $\rho_{\text{rdc}}(\mathbf{x}, \mathbf{y}) = \rho_{\text{rdc}}(f(\mathbf{x}), g(\mathbf{y}))$ for bijective functions f, g, and $\rho_{\text{rdc}}(\mathbf{x}, \mathbf{y}) = 1$ if for functions f or g, $\mathbf{y} = f(\mathbf{x})$ or $\mathbf{x} = g(\mathbf{y})$. The MCC $\Psi_{\rho_{\text{rdc}}}$ is a useful measure of identifiability for nonlinear ICA models.

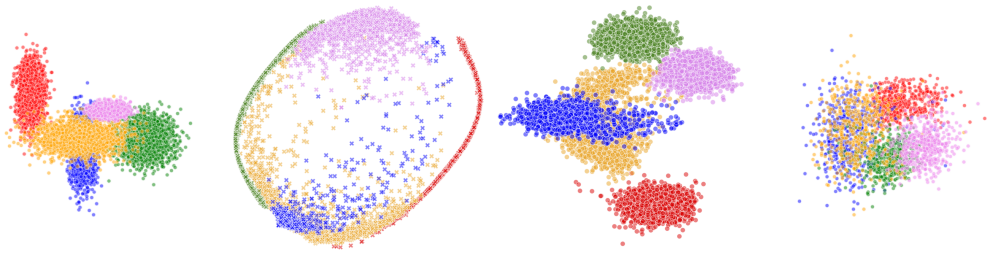
2.5.2 Simulations on nonlinear ICA data

Dataset 1: non-stationary data. We run simulations on data used previously in the nonlinear ICA literature (Hyvärinen and Morioka, 2016; Hyvärinen et al., 2019). We generate synthetic datasets where the sources are non-stationary Gaussian time series: we divide the sources into M segments of L samples each. The conditioning variable $\mathbf{u}$ (subsequently denoted u because it is scalar) is the segment label, and its distribution is uniform on the integer set $[\![1, M]\!]$. Within each segment, the conditional prior distribution is chosen from the family (2.9) for small k. When $k = 2$, we used mean and variance modulated Gaussian distribution. When $k = 1$, we used variance modulated

Gaussian or Laplace (to fall within the hypotheses of Theorem 2.12). The true parameters $\boldsymbol{\lambda}_i$ were randomly and independently generated across the segments and the components from a non degenerate distributions to satisfy assumption (iv) of Theorem 2.9. Following Hyvärinen et al. (2019), we mix the sources using a multi-layer perceptron (MLP) and add small Gaussian noise.

Dataset 2: significant mean modulated data. Here, we generated non-stationary 2D data from a modified dataset as follows: we generate $\mathbf{z}^*|u \sim \mathcal{N}(\boldsymbol{\mu}(u), \mathrm{diag}(\boldsymbol{\sigma}^2(u))$ where u is the segment index, $\mu_1(u) = 0$ for all u and $\mu_2(u) = \alpha\gamma(u)$ where $\alpha \in \mathbb{R}$ and γ is a permutation. Essentially, the mean of the second source, z_2^*, is significantly modulated by the segment index. An example is plotted in Figure [2.7b]. The variance $\boldsymbol{\sigma}^2(u)$ is generated randomly and independently across the segments. We then mix the sources into observations $\mathbf{x}$ such that $x_1 = \mathbf{MLP}(z_1^*, z_2^*)$ and $x_2 = z_2^*$, thus preserving the significant modulation of the mean in x_2. We note that this is just one of many potential mappings from $\mathbf{z}$ to $\mathbf{x}$ which could have been employed to yield significant mean modulation in x_2 across segments. TCL learns to unmix observations, $\mathbf{x}$, by solving a surrogate classification task. Formally, TCL seeks to train a deep network to accurately classify each observation into its corresponding segment. As such, the dataset mentioned above is designed to highlight the following limitation of TCL: due to its reliance on optimising a self-supervised objective, it can fail to recover latent variables when the associated task is too easy. In fact, by choosing a large enough value of the separation parameter α (in our experiments $\alpha = 2$), it is possible to classify samples by looking at the mean of x_2.

Model specification. Our estimates of the latent variables are generated from the variational posterior $q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{u}, \mathbf{x})$, for which we chose the following form: $q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x}, \mathbf{u}) = \mathcal{N}\left(\mathbf{z}|\mathbf{g}(\mathbf{x}, \mathbf{u}; \boldsymbol{\phi}_{\mathbf{g}}), \mathrm{diag}\, \boldsymbol{\sigma}^2(\mathbf{x}, \mathbf{u}; \boldsymbol{\phi}_{\boldsymbol{\sigma}})\right)$, a multivariate Gaussian with a diagonal covariance. The noise distribution $p_{\boldsymbol{\varepsilon}}$ is Gaussian with small variance. The functional parameters of the decoder and the inference model, as well as the conditional prior are chosen to be MLPs. We use an Adam optimizer (Kingma and Ba, 2014) to update the parameters of the network by maximizing $\mathcal{L}(\boldsymbol{\theta}, \boldsymbol{\phi})$ in equation (2.10). The architectural and hyperparameter choices are detailed in Appendix 2.F.1.

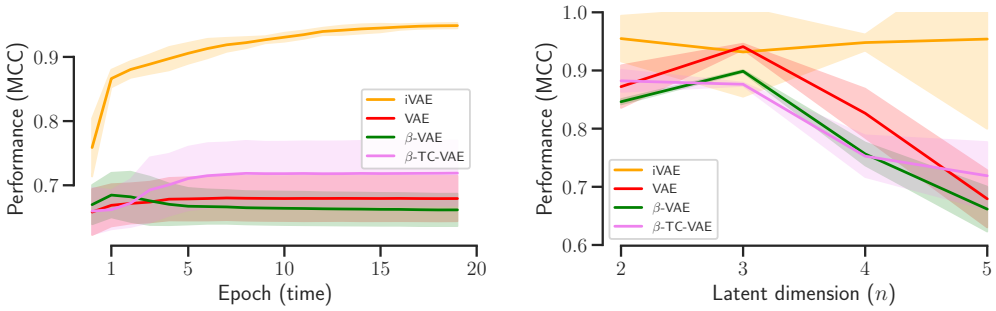


(a) $p_{\boldsymbol{\theta}^*}(\mathbf{z}|\mathbf{u})$ (b) $p_{\boldsymbol{\theta}^*}(\mathbf{x}|\mathbf{u})$ (c) $p_{\boldsymbol{\theta}}(\mathbf{z}|\mathbf{x},\mathbf{u})$ (d) $p_{\text{VAE}}(\mathbf{z}|\mathbf{x})$

Figure 2.1: Visualization of both observation and latent spaces in the case $n = d = 2$ and where the number of segments is $M = 5$ (segments are colour coded). First, data is generated in (a)-(b) as follows: (a) samples from the true distribution of the sources $p_{\boldsymbol{\theta}^*}(\mathbf{z}|\mathbf{u})$: Gaussian with non stationary mean and variance, (b) are observations sampled from $p_{\boldsymbol{\theta}^*}(\mathbf{x}|\mathbf{z})$. Second, after learning both a vanilla VAE and an iVAE models, we plot in (c) the latent variables sampled from the posterior $q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x},\mathbf{u})$ of the iVAE and in (d) the latent variables sampled from the posterior of the vanilla VAE.

2D example. First, we show a visualisation of identifiability of iVAE in a 2D case in Figure [2.1], where we plot the original sources, observed data and the posterior distributions learned by our model, compared to a vanilla VAE. Our method recovers the original sources up to trivial indeterminacies (rotation and sign flip), whereas the VAE fails to correctly separate the latent variables.

Comparison to VAE variants. We compared the performance of iVAE to a vanilla VAE. We used the same network architecture for both models, with the sole exception of the addition of the conditional prior in iVAE . When the data is centred, the VAE prior is Gaussian or Laplace. We also compared the performance to two models from the disentanglement literature, namely a β-VAE (Higgins et al., 2017) and a β-TC-VAE (Chen et al., 2018). The parameter β of the β-VAE and the parameters α, β and γ for β-TC-VAE were chosen by following the instructions of their respective authors. We trained these 4 models on the dataset described above, with $M = 40$, $L = 1000$, $d = 5$ and $n \in [2, 5]$. Figure [2.2a] compares performances obtained from an optimal choice of parameters achieved by iVAE and the three models discussed above when the dimension of the latent space equals the dimension of the data ($n = d = 5$). iVAE achieved an MCC score of above 95%, whereas the other three models fail at finding a good estimation of the true parameters. We



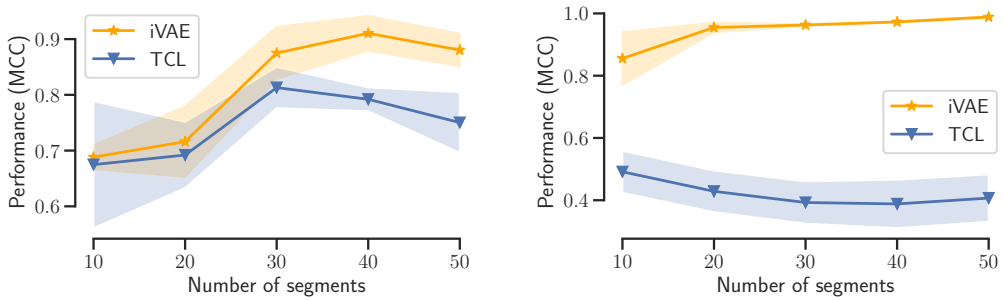
(a) Training dynamics (b) Changing n

Figure 2.2: Performance of iVAE in recovering the true sources, compared to VAE, β-VAE and β-TC-VAE, for $M = 40$, $L = 1000$ and $d = 5$ (and $n = 5$ for (a)).

further investigated the impact of the latent dimension on the performance in Figure [2.2b]. iVAE has much higher correlations than the three other models, especially as the dimension increases. Further visualization are in Appendix 2.F.3.

Comparison to TCL. Next, we compared our method to previous nonlinear ICA methods, namely TCL by Hyvärinen and Morioka (2016), which is based on a self-supervised classification task (see Section 1.2.2.2 for a brief review on TCL). We run simulations on the same dataset as Figure [2.2a], where we varied the number of segments from 10 to 50. Our method slightly outperformed TCL in our experiments. The results are reported in Figure [2.3a]. Note that according to Hyvärinen et al. (2019), TCL performs best among previously proposed methods for this kind of data.

Finally, we wanted to show that our method is robust to some failure modes that occur in self-supervised methods. The theory of TCL is premised on the notion that to accurately classify observations into their relative segments, the model must learn the true log-densities of sources within each segment. While such theory will hold in the limit of infinite data, we considered here a special case where accurate classification did not require learning the log-densities very precisely. This was achieved by generating synthetic data where x_2 alone contained sufficient information to perform classification, by making the mean of x_2 significantly modulated across segments (dataset 2 in Section 2.5.2). In such



(a) Normal (b) Easy to classify

Figure 2.3: (*a*) Performance of iVAE in comparison to TCL in recovering the true sources on normal data (*b*) Performance of iVAE in comparison to TCL in recovering the true sources on easy to classify data.

a setting, TCL is able to obtain high classification accuracy without unmixing observations, resulting in its failure to recover latent variables as reflected in Figure [2.3b]. In contrast, the proposed iVAE, by virtue of optimising a maximum likelihood objective, does not suffer from such degenerate behaviour.

Discrete observations. To further test the capabilities of our method, we tested it on discrete data and compared its identifiability performance to a vanilla VAE. The dimensions of the data and latent variables are $d = 100$ and $n = 10$. The results are shown in Figure [2.4a] and prove that our method is capable of performing discrete ICA.

Dimensionality selection and reduction. The examples in Figure [2.2] already showcased dimensionality reduction. In Figure [2.2b] for example, we have a mismatch between the dimensions of the latent variables and observations. In real-world ICA applications, we usually do not know the dimension of the latent variables beforehand. One way to guess it is to use the ELBO as a proxy to select the dimension. Our method enables this when compared to previous nonlinear ICA methods like TCL (Hyvärinen and Morioka, 2016). This is showcased in Figure [2.4b], where the real dimensions of the simulated data are $d^* = 80$ and $n^* = 15$, and we run multiple experiments where we vary the latent dimensions between 2 and 40. We can see that the ELBO can be a good proxy for dimension selection since it has a "knee" around the right value of

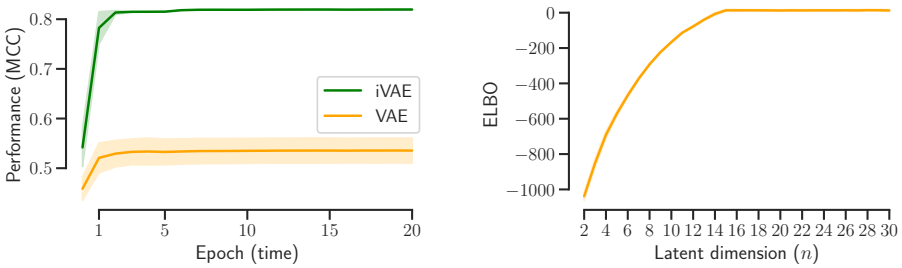
the dimension.

Hyperparameter selection. One important benefit of the proposed method is that it seeks to optimise an objective function derived from the marginal log-likelihood of observations. As such, it follows that we may employ the ELBO to perform hyperparameter selection. To verify this claim, we run experiments for various distinct choices of hyperparameters (for example, the dimension of hidden layers, number of hidden layers in the estimation network, learning rate, nonlinearities) on a synthetic dataset. Results are provided in Figure [2.4c] which serves to empirically demonstrate that the ELBO is indeed a good proxy for how accurately we can recover the true latent variables. In contrast, alternative methods for nonlinear ICA, such as TCL, do not provide principled and reliable proxies which reflect the accuracy of estimated latent sources.

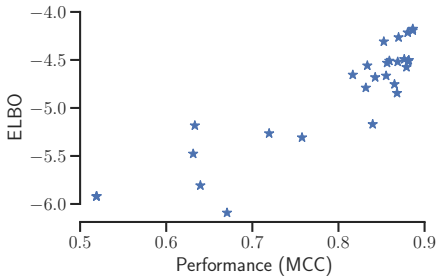
2.5.3 Application to causal discovery

An important application of ICA methods is within the domain of causal discovery (Peters et al., 2017). The use of ICA methods in this domain is premised on the equivalence between a (nonlinear) ICA model and the corresponding structural equation model (SEM). Such a connection was initially exploited in the linear case (Shimizu et al., 2006) and extended to the nonlinear case by Monti et al. (2019) who employed TCL.

Briefly, consider data $\mathbf{x} = (x_1, x_2)$. The goal is to establish if the causal direction is $x_1 \to x_2$, or $x_2 \to x_1$, or conclude that no (acyclic) causal relationship exists. Assuming $x_1 \to x_2$, then the problem can be described by the following SEM: $x_1 = f_1(n_1),\ x_2 = f_2(x_1, n_2)$ where $\mathbf{f} = (f_1, f_2)$ is a (possibly nonlinear) mapping and $\mathbf{n} = (n_1, n_2)$ are latent disturbances that are assumed to be independent. The above SEM can be seen as a nonlinear ICA model where latent disturbances, $\mathbf{n}$, are the sources. As such, we may perform causal discovery by first recovering latent disturbances (using TCL or iVAE) and then running a series of independence tests. Formally, if $x_1 \to x_2$ then, denoting statistical independence by $\perp\!\!\!\perp$, it suffices to verify that $x_1 \perp\!\!\!\perp n_2$ whereas $x_1 \not\perp\!\!\!\perp n_1$, $x_2 \not\perp\!\!\!\perp n_1$ and $x_2 \not\perp\!\!\!\perp n_2$. Such an approach can be extended beyond



(a) Discrete ICA (b) Dimensionality selection



(c) Hyperparameter selection

Figure 2.4: (a) Performance of iVAE and VAE on discrete ICA task. (b) Evolution of the post training ELBO as a function of the latent dimension. The real dimension of the data is $d^* = 80$ and the real dimension of the latent space is $n^* = 15$. We observe an elbow at around 15, thus successfully guessing the real dimension. (c) ELBO as a function of the performance. Each star is an experiment run for a different set of hyperparameters.

two-dimensional observations as described in Monti et al. (2019), and is called Nonlinear SEM Estimation based on Non-Stationarity (NonSENS).

Throughout all causal discovery experiments, we employ HSIC as a general test of statistical independence (Gretton et al., 2005). It is important to note that the aforementioned testing procedure can produce one of three decisions: $x_1 \to x_2$, $x_2 \to x_1$ or a third decision which states that no acyclic causal direction can be determined. The first two outcomes correspond to identifying the causal structure and will occur when we fail to reject the null hypothesis in only one of the four tests. Whereas the third decision (no evidence of acyclic causal structure) will be reported when either there is evidence to reject the null in all four tests, or we fail to reject the null more than once. Typically, this will occur if the nonlinear unmixing has failed to accurately recover the true latent sources.

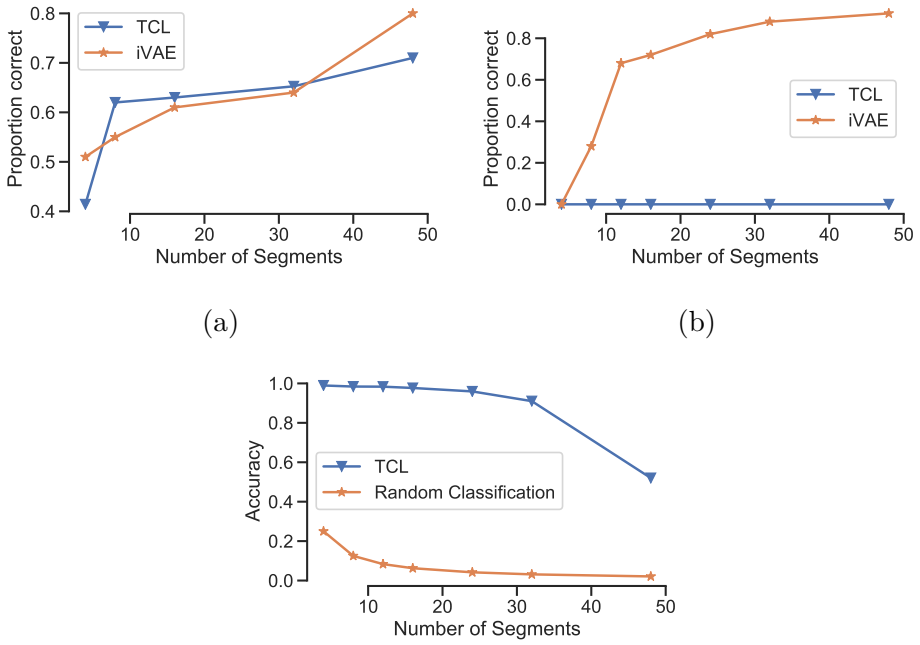


Figure 2.5: (*a*) Performance of nonlinear causal discovery for "normal" data, where iVAE or TCL are employed to recover latent disturbances. (*b*) Similarly, but when underlying sources display significant mean modulation across segments, making them easy to classify. (*c*) Classification accuracy of TCL when applied on data displaying significant mean modulation. We note that the accuracy of TCL is significantly above a random classifier, indicating that the surrogate classification problem employed in TCL training has been effectively optimized.

2.5.3.1 Simulated data

Experiments on "normal" simulated data. The data generation process used by Monti et al. (2019, Section 4) is similar to the one we described in Section 2.5.2, with the difference that the mixing should be in such a way that we get an acyclic causal relationship between the observations. This can be achieved by ensuring weight matrices in the mixing network are all lower-triangular, thereby introducing acyclic causal structure over observations. When comparing iVAE and TCL in this setting, we report the proportion of times the correct causal direction is reported. The results are reported in Figure [2.5a] where we note that both TCL and iVAE perform comparably.

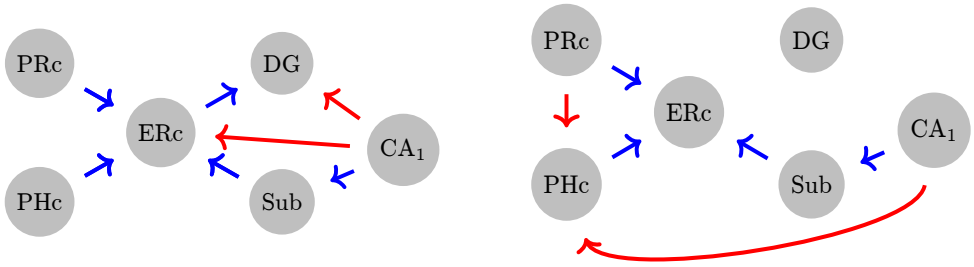
Experiments on significant mean modulated data. As a further experiment, we consider causal discovery in the scenario where one or both of the underlying sources demonstrate a significant mean modulation as shown in Figure [2.7]. In such a setting, the surrogate classification problem, which is solved as part of TCL training, becomes significantly easier to the extent that TCL no longer needs to learn an accurate representation of the log-density of sources within each segment. This is to the detriment of TCL as it implies that it cannot accurately recover latent sources and therefore fails at the task of causal discovery, as seen in Figure [2.5b]. This is a result of the fact that iVAE directly optimises the log-likelihood as opposed to a surrogate classification problem. Moreover, Figure [2.5c] visualises the mean classification accuracy for TCL as a function of the number of segments. We note that TCL consistently obtains classification accuracy that are significantly better than random classification. This provides evidence that the poor performance of TCL in the context of data with significant mean modulations is not a result of sub-optimal optimisation but are instead a negative consequence of TCL's reliance on solving a surrogate classification problem to perform nonlinear unmixing.

2.5.3.2 Real world fMRI data

To further demonstrate the benefits of iVAE as compared to TCL, both algorithms were employed to learn the causal structure from fMRI data collected by Poldrack et al. (2015) (see details in Appendix 2.F.2). The recovered causal graphs are shown in Figure [2.6]. Edges indicate causal relations between regions: blue edges are anatomically feasible whilst red edges are not. There is significant overlap between the estimated causal networks, but in the case of iVAE both anatomically incorrect edges (between CA_1 and ERc, and CA_1 and DG) actually correspond to indirect causal effects. This contrasts with TCL where incorrect edges are incompatible with anatomical structure and cannot be explained as indirect effects.

2.6 Conclusion

Unsupervised learning can have many different goals, such as: (*i*) approximate the data distribution, (*ii*) generate new samples, (*iii*) learn useful features,



(a) iVAE (b) TCL

Figure 2.6: Estimated causal graph on hippocampal fMRI data unmixing of sources is achieved via iVAE (left) or TCL (right). Blue edges are feasible given anatomical connectivity, red edges are not.

and above all (*iv*) learn the original latent code that generated the data (identifiability). Deep latent variable models typically implemented by VAEs are an excellent framework to achieve (*i*), and are thus our first building block. The nonlinear ICA model developed by Hyvärinen and Morioka (2016) and Hyvärinen et al. (2019) is the only existing framework to provably achieve (*iv*). We bring these two pieces together to create our new model termed iVAE . In particular, this is the first rigorous proof of identifiability in the context of VAEs. Our model in fact checks all the four boxes above that are desired in unsupervised learning.

The advantage of the new framework over typical deep latent variable models used with VAEs is that we actually recover the original latents, thus providing principled disentanglement. On the other hand, the advantages of this algorithm for solving nonlinear ICA over previous self-supervised heuristics are several; briefly, we significantly strengthen the identifiability results, we obtain the likelihood and can use MLE, we learn a forward model as well and can generate new data, and we consider the more general cases of noisy data with fewer components.

Appendices to Chapter 2

2.A Properties of the strongly exponential family

In this section, we give helpful properties of the strongly exponential family, which will be useful to prove the identifiability theorems. Some properties apply to univariate densities — as is duly specified.

Lemma 2.16. *Consider a* univariate strongly exponential *family distribution such that its sufficient statistic* $\mathbf{T}$ *is* differentiable *almost surely. Then* $T_i' \neq 0$ *almost everywhere on* $\mathbb{R}$ *for all* $1 \leq i \leq k$.

Proof. Suppose that p is strongly exponential, and let $\mathcal{X} = \cup_i \{x \in \mathbb{R}, T_i'(x) \neq 0\}$. Chose any $\boldsymbol{\lambda} \in \mathbb{R}^k \setminus \{0\}$. Then $\forall x \in \mathcal{X}, \langle \mathbf{T}'(x), \boldsymbol{\lambda} \rangle = 0$. By integrating, we find that $\langle \mathbf{T}(x), \boldsymbol{\lambda} \rangle = \text{const}$. By hypothesis, this means that $\mu_{Leb}(\mathcal{X}) = 0$. □

Lemma 2.17. *Consider a* univariate *strongly exponential distribution of size* $k \geq 2$ *with sufficient statistic* $\mathbf{T}(x) = (T_1(x), \ldots, T_k(x))$. *Further assume that* $\mathbf{T}$ *is* differentiable *almost everywhere.*

Then there exist k *distinct values* x_1 *to* x_k *such that* $(\mathbf{T}'(x_1), \ldots, \mathbf{T}'(x_k))$ *are linearly independent in* $\mathbb{R}^k$.

Proof. Suppose that for any choice of such k points $(x_1, \ldots, x_k)$, the vector family $(\mathbf{T}'(x_1), \ldots, \mathbf{T}'(x_k))$ is never linearly independent. That means that $\mathbf{T}'(\mathbb{R})$ is included in a subspace of $\mathbb{R}^k$ of dimension at most $k-1$.

Let $\boldsymbol{\lambda}$ be a non zero vector that is orthogonal to $\mathbf{T}'(\mathbb{R})$. Then for all $x \in \mathbb{R}$, we have $\langle \mathbf{T}'(x), \boldsymbol{\lambda} \rangle = 0$. By integrating we find that $\langle \mathbf{T}(x), \boldsymbol{\lambda} \rangle = \text{const}$. Since this is true for all $x \in \mathbb{R}$ and for a $\boldsymbol{\lambda} \neq 0$, we conclude that the distribution is not strongly exponential, which contradicts our hypothesis. □

Lemma 2.18. *Consider a strongly exponential distribution of size* $k \geq 2$. *Further assume that* $\mathbf{T}$ *is differentiable almost everywhere. Then there exist* $k+1$ *distinct values* $\mathbf{x}^{(0)}$ *to* $\mathbf{x}^{(k)}$ *such that the matrix*

$$\mathbf{R} = \left(\mathbf{T}(\mathbf{x}^{(1)}) - \mathbf{T}(\mathbf{x}^{(0)}), \ldots, \mathbf{T}(\mathbf{x}^{(k)}) - \mathbf{T}(\mathbf{x}^{(0)})\right) \in \mathbb{R}^{k \times k}$$

is invertible.

Proof. Suppose that for a given $\mathbf{x}^{(0)}$, we can't find points $\mathbf{x}^{(1)}$ to $\mathbf{x}^{(k)}$ such that the matrix $\mathbf{R}$ is not invertible. Then the function $\mathbf{g}(\mathbf{x}) = \mathbf{T}(\mathbf{x}) - \mathbf{T}(\mathbf{x}^{(0)})$ would live in a $k-1$ dimensional subspace of $\mathbb{R}^k$. This means that we can find a nonzero $\boldsymbol{\theta} \in \mathbb{R}^k$ such that $\mathbf{g}(\mathbf{x})^\top \boldsymbol{\theta} = 0$, which implies that $\mathbf{T}(\mathbf{x})^\top \boldsymbol{\theta} = \mathbf{T}(\mathbf{x}^{(0)})^\top \boldsymbol{\theta} = \text{const}$ for any $\mathbf{x} \in \mathcal{X}$. This contradicts the assumption that the distribution is strongly exponential. Therefore we have that $\mathbf{R}$ is invertible. □

Lemma 2.19. *Consider a univariate strongly exponential distribution of size $k \geq 2$ with sufficient statistic $\mathbf{T}$. Further assume that $\mathbf{T}$ is* twice differentiable *almost everywhere. Then*

$$\dim\left(\operatorname{span}\left\{(T_i'(x), T_i''(x))^\top, 1 \leq i \leq k\right\}\right) \geq 2 \tag{2.20}$$

almost everywhere on $\mathbb{R}$.

Proof. Suppose there exists a set $\mathcal{X}$ of measure greater than zero where equation (2.20) doesn't hold. This means that the vectors $[T_i'(x), T_i''(x)]^\top$ are collinear for any i and for all $x \in \mathcal{X}$. In particular, it means that there exists $\alpha \in \mathbb{R}^k \setminus \{0\}$ s.t. $\sum_i \alpha_i T_i'(x) = 0$. By integrating, we get $\langle \mathbf{T}(x), \boldsymbol{\alpha} \rangle = \text{const}$, $\forall x \in \mathcal{X}$. Since $l(\mathcal{X}) > 0$, this contradicts equation (2.17). □

Lemma 2.20. *Consider n univariate strongly exponential distributions of size $k \geq 2$ with respective sufficient statistics $\mathbf{T}_j = (T_{j,1}, \dots T_{j,k})$, $1 \leq j \leq n$. Further assume that the sufficient statistics are* twice differentiable. *Define the vectors $\mathbf{e}^{(j,i)} \in \mathbb{R}^{2n}$, such that $\mathbf{e}^{(j,i)} = \big(0, \dots, 0, T_{j,i}', T_{j,i}'', 0, \dots, 0\big)$, where the nonzero entries are at indices $(2j, 2j+1)$. Finally, let $\mathbf{x} \coloneqq (x_1, \dots, x_n) \in \mathbb{R}^n$.*

Then the matrix

$$\overline{\mathbf{e}}(\mathbf{x}) \coloneqq (\mathbf{e}^{(1,1)}(x_1), \dots, \mathbf{e}^{(1,k)}(x_1), \dots \mathbf{e}^{(n,1)}(x_n), \dots, \mathbf{e}^{(n,k)}(x_n)) \tag{2.21}$$

of size $(2n \times nk)$ has rank $2n$ almost everywhere on $\mathbb{R}^n$.

Proof. It is easy to see that the matrix $\overline{\mathbf{e}}(\mathbf{x})$ has at least rank n, because by varying the index j in $\mathbf{e}^{(j,i)}$ we change the position of the nonzero entries. By changing the index i, we change the component within the same sufficient statistic. Now fix j and consider the submatrix $\left[\mathbf{e}^{(j,1)}(x_j), \dots, \mathbf{e}^{(j,k)}(x_j)\right]$. By using Lemma 2.19, we deduce that this submatrix has rank greater or equal

to 2 because its columns span a subspace of dimensions greater or equal to 2 almost everywhere on $\mathbb{R}$. Thus, we conclude that the rank of $\overline{\mathbf{e}}(\mathbf{x})$ is $2n$ almost everywhere on $\mathbb{R}^n$. □

2.B Proofs

2.B.1 Identifiability proofs

Proposition 2.21. *The binary relations $\sim_{\mathbf{A}}$ and $\sim_{\mathbf{P}}$ defined in Definition 2.6 are equivalence relations on Θ.*

Proof. The following proof applies to both $\sim_{\mathbf{A}}$ and $\sim_{\mathbf{P}}$ which we will simply denote by $\sim$.

It is clear that $\sim$ is reflexive and symmetric. Let $((\mathbf{f}, \mathbf{T}, \boldsymbol{\lambda}), (\tilde{\mathbf{f}}, \tilde{\mathbf{T}}, \tilde{\boldsymbol{\lambda}}), (\overline{\mathbf{f}}, \overline{\mathbf{T}}, \overline{\boldsymbol{\lambda}})) \in \Theta^3$, s.t. $(\mathbf{f}, \mathbf{T}, \boldsymbol{\lambda}) \sim (\tilde{\mathbf{f}}, \tilde{\mathbf{T}}, \tilde{\boldsymbol{\lambda}})$ and $(\mathbf{f}, \mathbf{T}, \boldsymbol{\lambda}) \sim (\overline{\mathbf{f}}, \overline{\mathbf{T}}, \overline{\boldsymbol{\lambda}})$. Then $\exists \mathbf{A}_1, \mathbf{A}_2$ and $\mathbf{c}_1, \mathbf{c}_2$ s.t.

$$\begin{aligned} \mathbf{T}(\mathbf{f}^{-1}(\mathbf{x})) &= \mathbf{A}_1 \tilde{\mathbf{T}}(\tilde{\mathbf{f}}^{-1}(\mathbf{x})) + \mathbf{c}_1 \text{ and} \\ \overline{\mathbf{T}}(\overline{\mathbf{f}}^{-1}(\mathbf{x})) &= \mathbf{A}_2 \mathbf{T}(\mathbf{f}^{-1}(\mathbf{x})) + \mathbf{c}_2 \\ &= \mathbf{A}_2 \mathbf{A}_1 \tilde{\mathbf{T}}(\tilde{\mathbf{f}}^{-1}(\mathbf{x})) + \mathbf{A}_2 \mathbf{c}_1 + \mathbf{c}_2 \\ &= \mathbf{A}_3 \tilde{\mathbf{T}}(\tilde{\mathbf{f}}^{-1}(\mathbf{x})) + \mathbf{c}_3, \end{aligned} \tag{2.22}$$

and thus $(\tilde{\mathbf{f}}, \tilde{\mathbf{T}}, \tilde{\boldsymbol{\lambda}}) \sim (\overline{\mathbf{f}}, \overline{\mathbf{T}}, \overline{\boldsymbol{\lambda}})$. □

Proof of Theorem 2.9. The proof of this Theorem is done in three steps:

1. In the first step, we use a simple convolutional trick made possible by assumption (i), to transform the equality of observed data distributions into equality of noiseless distributions. In other words, it simplifies the noisy case into a noiseless case. This step results in equation (2.32).

2. The second step consists of removing all terms that are either a function of observations $\mathbf{x}$ or auxiliary variables $\mathbf{u}$. This is done by introducing the points provided by assumption (iv), and using $\mathbf{u}_0$ as a "pivot". This is simply done in equations equations (2.32) to (2.35).

3. The last step of the proof is slightly technical. The the goal is to show that the linear transformation is invertible thus resulting in an equivalence relation. This is where we use assumption (iii).

Step I We introduce here the volume of a matrix denoted $\operatorname{vol}\mathbf{A}$ as the product of the singular values of $\mathbf{A}$. When $\mathbf{A}$ is full column rank, $\operatorname{vol}\mathbf{A} = \sqrt{\det \mathbf{A}^\top \mathbf{A}}$, and when $\mathbf{A}$ is invertible, $\operatorname{vol}\mathbf{A} = |\det \mathbf{A}|$. The matrix volume can be used in the change of variable formula as a replacement for the absolute determinant of the Jacobian (Ben-Israel, 1999). This is most useful when the Jacobian is a rectangular matrix ($n < d$). Suppose we have two sets of parameters $(\mathbf{f}, \mathbf{T}, \boldsymbol{\lambda})$ and $(\tilde{\mathbf{f}}, \tilde{\mathbf{T}}, \tilde{\boldsymbol{\lambda}})$ such that $p_{\mathbf{f},\mathbf{T},\boldsymbol{\lambda}}(\mathbf{x}|\mathbf{u}) = p_{\tilde{\mathbf{f}},\tilde{\mathbf{T}},\tilde{\boldsymbol{\lambda}}}(\mathbf{x}|\mathbf{u})$ for all pairs $(\mathbf{x}, \mathbf{u})$. Then:

$$\int_{\mathcal{Z}} p_{\mathbf{T},\boldsymbol{\lambda}}(\mathbf{z}|\mathbf{u}) p_{\mathbf{f}}(\mathbf{x}|\mathbf{z}) \mathrm{d}\mathbf{z} = \int_{\mathcal{Z}} p_{\tilde{\mathbf{T}},\tilde{\boldsymbol{\lambda}}}(\mathbf{z}|\mathbf{u}) p_{\tilde{\mathbf{f}}}(\mathbf{x}|\mathbf{z}) \mathrm{d}\mathbf{z} \tag{2.23}$$

$$\Rightarrow \quad \int_{\mathcal{Z}} p_{\mathbf{T},\boldsymbol{\lambda}}(\mathbf{z}|\mathbf{u}) p_{\varepsilon}(\mathbf{x} - \mathbf{f}(\mathbf{z})) \mathrm{d}\mathbf{z} = \int_{\mathcal{Z}} p_{\tilde{\mathbf{T}},\tilde{\boldsymbol{\lambda}}}(\mathbf{z}|\mathbf{u}) p_{\varepsilon}(\mathbf{x} - \tilde{\mathbf{f}}(\mathbf{z})) \mathrm{d}\mathbf{z} \tag{2.24}$$

$$\Rightarrow \quad \int_{\mathcal{X}} p_{\mathbf{T},\boldsymbol{\lambda}}(\mathbf{f}^{-1}(\bar{\mathbf{x}})|\mathbf{u}) \operatorname{vol} \mathbf{J}_{\mathbf{f}^{-1}}(\bar{\mathbf{x}}) p_{\varepsilon}(\mathbf{x} - \bar{\mathbf{x}}) \mathrm{d}\bar{\mathbf{x}} = \int_{\mathcal{X}} p_{\tilde{\mathbf{T}},\tilde{\boldsymbol{\lambda}}}(\tilde{\mathbf{f}}^{-1}(\bar{\mathbf{x}})|\mathbf{u}) \operatorname{vol} \mathbf{J}_{\tilde{\mathbf{f}}^{-1}}(\bar{\mathbf{x}}) p_{\varepsilon}(\mathbf{x} - \bar{\mathbf{x}}) \mathrm{d}\bar{\mathbf{x}} \tag{2.25}$$

$$\Rightarrow \quad \int_{\mathbb{R}^d} \tilde{p}_{\mathbf{T},\boldsymbol{\lambda},\mathbf{f},\mathbf{u}}(\bar{\mathbf{x}}) p_{\varepsilon}(\mathbf{x} - \bar{\mathbf{x}}) \mathrm{d}\bar{\mathbf{x}} = \int_{\mathbb{R}^d} \tilde{p}_{\tilde{\mathbf{T}},\tilde{\boldsymbol{\lambda}},\tilde{\mathbf{f}},\mathbf{u}}(\bar{\mathbf{x}}) p_{\varepsilon}(\mathbf{x} - \bar{\mathbf{x}}) \mathrm{d}\bar{\mathbf{x}} \tag{2.26}$$

$$\Rightarrow \quad (\tilde{p}_{\mathbf{T},\boldsymbol{\lambda},\mathbf{f},\mathbf{u}} * p_{\varepsilon})(\mathbf{x}) = (\tilde{p}_{\tilde{\mathbf{T}},\tilde{\boldsymbol{\lambda}},\tilde{\mathbf{f}},\mathbf{u}} * p_{\varepsilon})(\mathbf{x}) \tag{2.27}$$

$$\Rightarrow \quad F[\tilde{p}_{\mathbf{T},\boldsymbol{\lambda},\mathbf{f},\mathbf{u}}](\omega) \varphi_{\varepsilon}(\omega) = F[\tilde{p}_{\tilde{\mathbf{T}},\tilde{\boldsymbol{\lambda}},\tilde{\mathbf{f}},\mathbf{u}}](\omega) \varphi_{\varepsilon}(\omega) \tag{2.28}$$

$$\Rightarrow \quad F[\tilde{p}_{\mathbf{T},\boldsymbol{\lambda},\mathbf{f},\mathbf{u}}](\omega) = F[\tilde{p}_{\tilde{\mathbf{T}},\tilde{\boldsymbol{\lambda}},\tilde{\mathbf{f}},\mathbf{u}}](\omega) \tag{2.29}$$

$$\Rightarrow \quad \tilde{p}_{\mathbf{T},\boldsymbol{\lambda},\mathbf{f},\mathbf{u}}(\mathbf{x}) = \tilde{p}_{\tilde{\mathbf{T}},\tilde{\boldsymbol{\lambda}},\tilde{\mathbf{f}},\mathbf{u}}(\mathbf{x}) \tag{2.30}$$

where:

- in equation (2.25), $\mathbf{J}$ denotes the Jacobian, and we made the change of variable $\bar{\mathbf{x}} = \mathbf{f}(\mathbf{z})$ on the left hand side, and $\bar{\mathbf{x}} = \tilde{\mathbf{f}}(\mathbf{z})$ on the right hand side.
- in equation (2.26), we introduced

$$\tilde{p}_{\mathbf{T},\boldsymbol{\lambda},\mathbf{f},\mathbf{u}}(\mathbf{x}) = p_{\mathbf{T},\boldsymbol{\lambda}}(\mathbf{f}^{-1}(\mathbf{x})|\mathbf{u}) \operatorname{vol} \mathbf{J}_{\mathbf{f}^{-1}}(\mathbf{x}) \mathbb{1}_{\mathcal{X}}(\mathbf{x}) \tag{2.31}$$

  on the left hand side, and similarly on the right hand side.
- in equation (2.27), we used $*$ for the convolution operator.
- in equation (2.28), we used $F[.]$ to designate the Fourier transform, and where $\varphi_{\varepsilon} = F[p_{\varepsilon}]$ (by definition of the characteristic function).

- in equation (2.29), we dropped $\varphi_\varepsilon(\omega)$ from both sides as it is nonzero almost everywhere (by assumption (i)).

Equation (2.30) is valid for all $(\mathbf{x}, \mathbf{u}) \in \mathcal{X} \times \mathcal{U}$. What is basically says is that for the distributions to be the same after adding the noise, the noise-free distributions have to be the same. Note that $\mathbf{x}$ here is a general variable and we are actually dealing with the noise-free probability densities.

Step II By taking the logarithm on both sides of equation (2.30) and replacing $p_{\mathbf{T},\boldsymbol{\lambda}}$ by its expression from (2.9), we get:

$$\begin{aligned}\log \operatorname{vol} \mathbf{J}_{\mathbf{f}^{-1}}(\mathbf{x}) + \sum_{i=1}^{n}(\log Q_i(f_i^{-1}(\mathbf{x})) - \log Z_i(\mathbf{u}) + \sum_{j=1}^{k} T_{i,j}(f_i^{-1}(\mathbf{x}))\lambda_{i,j}(\mathbf{u})) = \\ \log \operatorname{vol} \mathbf{J}_{\tilde{\mathbf{f}}^{-1}}(\mathbf{x}) + \sum_{i=1}^{n}(\log \tilde{Q}_i(\tilde{f}_i^{-1}(\mathbf{x})) - \log \tilde{Z}_i(\mathbf{u}) + \sum_{j=1}^{k} \tilde{T}_{i,j}(\tilde{f}_i^{-1}(\mathbf{x}))\tilde{\lambda}_{i,j}(\mathbf{u})).\end{aligned} \tag{2.32}$$

Let $\mathbf{u}_0, \ldots, \mathbf{u}_{nk}$ be the points provided by assumption (iv) of the Theorem, and define $\overline{\boldsymbol{\lambda}}(\mathbf{u}) = \boldsymbol{\lambda}(\mathbf{u}) - \boldsymbol{\lambda}(\mathbf{u}_0)$. We plug each of those $\mathbf{u}_l$ in (2.32) to obtain $nk+1$ such equations. We subtract the first equation for $\mathbf{u}_0$ from the remaining nk equations to get for $l = 1, \ldots, nk$:

$$\langle \mathbf{T}(\mathbf{f}^{-1}(\mathbf{x})), \overline{\boldsymbol{\lambda}}(\mathbf{u}_l)\rangle + \sum_i \log \frac{Z_i(\mathbf{u}_0)}{Z_i(\mathbf{u}_l)} = \langle \tilde{\mathbf{T}}(\tilde{\mathbf{f}}^{-1}(\mathbf{x})), \overline{\tilde{\boldsymbol{\lambda}}}(\mathbf{u}_l)\rangle + \sum_i \log \frac{\tilde{Z}_i(\mathbf{u}_0)}{\tilde{Z}_i(\mathbf{u}_l)}. \tag{2.33}$$

Let $\mathbf{L}$ bet the matrix defined in assumption (iv), and $\tilde{\mathbf{L}}$ similarly defined for $\tilde{\boldsymbol{\lambda}}$ ($\tilde{\mathbf{L}}$ is not necessarily invertible). Define $b_l = \sum_i \log \frac{\tilde{Z}_i(\mathbf{u}_0) Z_i(\mathbf{u}_l)}{Z_i(\mathbf{u}_0)\tilde{Z}_i(\mathbf{u}_l)}$ and $\mathbf{b}$ the vector of all b_l for $l = 1, \ldots, nk$. Expressing (2.33) for all points $\mathbf{u}_l$ in matrix form, we get:

$$\mathbf{L}^\top \mathbf{T}(\mathbf{f}^{-1}(\mathbf{x})) = \tilde{\mathbf{L}}^\top \tilde{\mathbf{T}}(\tilde{\mathbf{f}}^{-1}(\mathbf{x})) + \mathbf{b}. \tag{2.34}$$

We multiply both sides of (2.34) by the transpose of the inverse of $\mathbf{L}^\top$ from the left to find:

$$\mathbf{T}(\mathbf{f}^{-1}(\mathbf{x})) = \mathbf{A}\tilde{\mathbf{T}}(\tilde{\mathbf{f}}^{-1}(\mathbf{x})) + \mathbf{c}, \tag{2.35}$$

where $\mathbf{A} = \mathbf{L}^{-\top}\tilde{\mathbf{L}}$ and $\mathbf{c} = \mathbf{L}^{-\top}\mathbf{b}$.

Step III Now by definition of $\mathbf{T}$ and according to assumption (iii), its Jacobian exists and is an $nk \times n$ matrix of rank n. This implies that the

Jacobian of $\tilde{\mathbf{T}} \circ \tilde{\mathbf{f}}^{-1}$ exists and is of rank n and so is $\mathbf{A}$. We distinguish two cases:

- If $k = 1$, then this means that $\mathbf{A}$ is invertible (because $\mathbf{A}$ is $n \times n$).
- If $k > 1$, define $\bar{\mathbf{x}} = \mathbf{f}^{-1}(\mathbf{x})$ and $\mathbf{T}_i(\bar{x}_i) = (T_{i,1}(\bar{x}_i), \dots T_{i,k}(\bar{x}_i))$. According to Lemma 2.17, for each $i \in [1, \dots, n]$ there exist k points $\bar{x}_i^1, \dots, \bar{x}_i^k$ such that $(\mathbf{T}_i'(\bar{x}_i^1), \dots, \mathbf{T}_i'(\bar{x}_i^k))$ are linearly independent. Collect those points into k vectors $(\bar{\mathbf{x}}^1, \dots, \bar{\mathbf{x}}^k)$, and concatenate the k Jacobians $\mathbf{J}_{\mathbf{T}}(\bar{\mathbf{x}}^l)$ evaluated at each of those vectors horizontally into the matrix $\mathbf{V} = (\mathbf{J}_{\mathbf{T}}(\bar{\mathbf{x}}^1), \dots, \mathbf{J}_{\mathbf{T}}(\bar{\mathbf{x}}^k))$ (and similarly define $\tilde{\mathbf{V}}$ as the concatenation of the Jacobians of $\tilde{\mathbf{T}}(\tilde{\mathbf{f}}^{-1} \circ \mathbf{f}(\bar{\mathbf{x}}))$ evaluated at those points). Then the matrix $\mathbf{V}$ is invertible (through a combination of Lemma 2.17 and the fact that each component of $\tilde{\mathbf{T}}$ is univariate). By differentiating (2.35) for each $\mathbf{x}^l$, we get (in matrix form):
$$\mathbf{V} = \mathbf{A}\tilde{\mathbf{V}}. \tag{2.36}$$
The invertibility of $\mathbf{V}$ implies the invertibility of $\mathbf{A}$ and $\tilde{\mathbf{V}}$.

Hence, (2.35) and the invertibility of $\mathbf{A}$ mean that $(\tilde{\mathbf{f}}, \tilde{\mathbf{T}}, \tilde{\boldsymbol{\lambda}}) \sim (\mathbf{f}, \mathbf{T}, \boldsymbol{\lambda})$.

Moreover, we have the following observations:

- the invertibility of $\mathbf{A}$ and $\mathbf{L}$ imply that $\tilde{\mathbf{L}}$ is invertible,
- because the Jacobian of $\tilde{\mathbf{T}} \circ \tilde{\mathbf{f}}^{-1}$ is full rank and $\tilde{\mathbf{f}}$ is injective (hence its Jacobian is full rank too), $\mathbf{J}_{\tilde{\mathbf{T}}}$ has to be full rank too, and $\tilde{T}_{i,j}'(z) \neq 0$ almost everywhere.
- the real equivalence class of identifiability may actually be narrower that what is defined by $\sim$, as the matrix $\mathbf{A}$ and the vector $\mathbf{c}$ here have very specific forms, and are functions of $\boldsymbol{\lambda}$ and $\tilde{\boldsymbol{\lambda}}$.

This concludes the proof. □

Proof of Theorem 2.11. The proof of this Theorem is done in two main steps.

1. The first step is to show that $\tilde{\mathbf{f}}^{-1} \circ \mathbf{f}$ is a pointwise function. This is done by showing that the product of any two distinct partial derivatives of any component is always zero. Along with invertibility, this means that each

component depends exactly on one variable. This is where we use the two additional assumptions required by the Theorem.

2. In the second step, we plug the result of the first step in the equation that resulted from Theorem 2.9 (see equation (2.42)). The fact that $\mathbf{T}$, $\tilde{\mathbf{T}}$ and $\tilde{\mathbf{f}}^{-1} \circ \mathbf{f}$ are all pointwise functions implies that $\mathbf{A}$ is necessarily a permutation matrix.

Step I In this Theorem we suppose that $k \geq 2$. The assumptions of Theorem 2.9 hold, and so we have

$$\mathbf{T}(\mathbf{f}^{-1}(\mathbf{x})) = \mathbf{A}\tilde{\mathbf{T}}(\tilde{\mathbf{f}}^{-1}(\mathbf{x})) + \mathbf{c} \tag{2.37}$$

for an invertible $A \in \mathbb{R}^{nk \times nk}$. We will index $\mathbf{A}$ by four indices (i, l, a, b), where $1 \leq i \leq n, 1 \leq l \leq k$ refer to the rows and $1 \leq a \leq n, 1 \leq b \leq k$ to the columns. Let $\mathbf{v}(\mathbf{z}) = \tilde{\mathbf{f}}^{-1} \circ \mathbf{f}(\mathbf{z}) : \mathcal{Z} \to \mathcal{Z}$. Note that $\mathbf{v}$ is bijective because $\mathbf{f}$ and $\tilde{\mathbf{f}}$ are injective. Our goal is to show that $v_i(\mathbf{z})$ is a function of only one z_{j_i}, for all i. We will denote by $v_i^s := \frac{\partial v_i}{\partial z_s}(\mathbf{z})$, and $v_i^{st} := \frac{\partial^2 v_i}{\partial z_s \partial z_t}(\mathbf{z})$. For each $1 \leq i \leq n$ and $1 \leq l \leq k$, we get by differentiating (2.37) with respect to z_s:

$$\delta_{is} T'_{i,l}(z_i) = \sum_{a,b} A_{i,l,a,b} \tilde{T}'_{a,b}(v_a(\mathbf{z})) v_a^s(\mathbf{z}), \tag{2.38}$$

and by differentiating (2.38) with respect to $z_t, t > s$:

$$0 = \sum_{a,b} A_{i,l,a,b} \left(\tilde{T}'_{a,b}(v_a(\mathbf{z})) v_a^{s,t}(\mathbf{z}) + \tilde{T}''_{a,b}(v_a(\mathbf{z})) v_a^s(\mathbf{z}) v_a^t(\mathbf{z}) \right). \tag{2.39}$$

This equation is valid for all pairs $(s, t), t > s$. Define

$$\begin{aligned}
\mathbf{B}_a(\mathbf{z}) &:= \left(v_a^{1,2}(\mathbf{z}), \ldots, v_a^{n-1,n}(\mathbf{z}) \right) \in \mathbb{R}^{\frac{n(n-1)}{2}}, \\
\mathbf{C}_a(\mathbf{z}) &:= \left(v_a^1(\mathbf{z}) v_a^2(\mathbf{z}), \ldots, v_a^{n-1}(\mathbf{z}) v_a^n(\mathbf{z}) \right) \in \mathbb{R}^{\frac{n(n-1)}{2}}, \\
\mathbf{M}(\mathbf{z}) &:= (\mathbf{B}_1(\mathbf{z}), \mathbf{C}_1(\mathbf{z}), \ldots, \mathbf{B}_n(\mathbf{z}), \mathbf{C}_n(\mathbf{z})) \in \mathbb{R}^{\frac{n(n-1)}{2} \times 2n},
\end{aligned}$$

and let

$$\mathbf{e}^{(a,b)} := (0, \ldots, 0, T'_{a,b}, T''_{a,b}, 0, \ldots, 0) \in \mathbb{R}^{2n}$$

such that the nonzero entries are at indices $(2a, 2a+1)$ and

$$\bar{\mathbf{e}}(\mathbf{z}) := (\mathbf{e}^{(1,1)}(z_1), \ldots, \mathbf{e}^{(1,k)}(z_1), \ldots, \mathbf{e}^{(n,1)}(z_n), \ldots, \mathbf{e}^{(n,k)}(z_n)) \in \mathbb{R}^{2n \times nk}.$$

Finally, denote by $\mathbf{A}_{i,l}$ the (i,l)-th row of $\mathbf{A}$. Then by grouping equation (2.39) for all valid pairs (s,t) and pairs (i,l) and writing it in matrix form, we get:

$$\mathbf{M}(\mathbf{z})\overline{\mathbf{e}}(\mathbf{z})\mathbf{A} = 0. \tag{2.40}$$

Now by Lemma 2.20, we know that $\overline{\mathbf{e}}(\mathbf{z})$ has rank $2n$ almost surely on $\mathcal{Z}$. Since $\mathbf{A}$ is invertible, it is full rank, and thus $\operatorname{rank}(\overline{\mathbf{e}}(\mathbf{z})\mathbf{A}) = 2n$ almost surely on $\mathcal{Z}$. It suffices then to multiply by its pseudo-inverse from the right to get

$$\mathbf{M}(\mathbf{z}) = 0. \tag{2.41}$$

In particular, $\mathbf{C}_a(\mathbf{z}) = 0$ for all $1 \leq a \leq n$. This means that the Jacobian of $\mathbf{v}$ at each $\mathbf{z}$ has at most one nonzero entry in each row. By invertibility and continuity of $\mathbf{J}_{\mathbf{v}}$, we deduce that the location of the nonzero entries are fixed and do not change as a function of $\mathbf{z}$. This proves that $\tilde{\mathbf{f}}^{-1} \circ \mathbf{f}$ is pointwise nonlinearity.

Step II Let $\overline{\mathbf{T}}(\mathbf{z}) = \tilde{\mathbf{T}}(\mathbf{v}(\mathbf{z})) + \mathbf{A}^{-1}\mathbf{c}$. $\overline{\mathbf{T}}$ is a composition of a permutation and pointwise nonlinearity. Without any loss of generality, we assume that the permutation in $\overline{\mathbf{T}}$ is the identity. Plugging this back into equation (2.37) yields:

$$\mathbf{T}(\mathbf{z}) = \mathbf{A}\overline{\mathbf{T}}(\mathbf{z}). \tag{2.42}$$

Let $\mathbf{D} = \mathbf{A}^{-1}$. The last equation is valid for every component:

$$\overline{T}_{i,l}(z_i) = \sum_{a,b} D_{i,l,a,b} T_{a,b}(z_a). \tag{2.43}$$

By differentiating both sides with respect to z_s where $s \neq i$ we get

$$0 = \sum_{b} D_{i,l,s,b} T'_{s,b}(z_s). \tag{2.44}$$

By Lemma 1.7, we get $D_{i,l,s,b} = 0$ for all $1 \leq b \leq k$. Since equation (2.44) is valid for all l and all $s \neq i$, we deduce that the matrix $\mathbf{D}$ has a block diagonal form:

$$\mathbf{D} = \begin{pmatrix} D_1 & & \\ & \ddots & \\ & & D_n \end{pmatrix}. \tag{2.45}$$

We conclude that $\mathbf{A}$ has the same block diagonal form. Each block i transforms $\mathbf{T}_i(\mathbf{z})$ into $\overline{\mathbf{T}}_i(\mathbf{z})$, which achieves the proof. □

Proof of Theorem 2.12. This proof uses concepts borrowed from differential geometry. A good reference is the monograph by Lee (2003).

By defining $\mathbf{v} = \mathbf{f}^{-1} \circ \tilde{\mathbf{f}}$, equation (2.35) implies that each function $T_i \circ v_i$ can be written as a separable sum, *i.e.* a sum of n maps where each map $h_{i,a}$ is function of only one component z_a.

Intuitively, since T_i is not monotonic, it admits a local extremum (supposed to be a minimum). By working locally around this minimum, we can suppose that it is global and attained at a unique point y_i. The smoothness condition on $\mathbf{v}$ imply that the manifold where $T_i \circ v_i$ is minimized has dimension $n-1$. This is where we need assumption (ii) of the Theorem.

On the other hand, because of the separability in the sum, each non constant $h_{i,k}$ (minimized as a consequence of minimizing $T_i \circ v_i$) introduces a constraint on this manifold that reduces its dimension by 1. That's why we can only have one non constant $h_{i,k}$ for each i.

In this Theorem we suppose that $k = 1$. For simplicity, we drop the exponential family component index: $T_i \coloneqq T_{i,1}$. By introducing $\mathbf{v} = \mathbf{f}^{-1} \circ \tilde{\mathbf{f}}$ and $h_{i,a}(z_a) = A_{i,a}\tilde{T}_a(z_a) + \frac{c_i}{n}$ into equation (2.35), we can rewrite it as:

$$T_i(v_i(\mathbf{z})) = \sum_{a=1}^{n} h_{i,a}(z_a) \tag{2.46}$$

for all $1 \leq i \leq n$.

By assumption, $h_{i,a}$ is not monotonic, and so is T_i. So for each a, there exists $\tilde{y}_{i,a}$ where $h_{i,a}$ reaches an extremum, which we suppose is a minimum without loss of generality. This implies that $T_i \circ v_i$ reaches a minimum at $\tilde{\mathbf{y}}_i \coloneqq (\tilde{y}_{i,1}, \ldots, \tilde{y}_{i,n})$, which in turn implies that $y_i \coloneqq v_i(\tilde{\mathbf{y}}_i)$ is a point where T_i reaches a local minimum. Let U be an open set centered around y_i, and let $\tilde{V} \coloneqq v_i^{-1}[U]$ the preimage of U by v_i. Because v_i is continuous, $\tilde{V}$ is open in $\mathbb{R}^n$ and non-empty because $\tilde{\mathbf{y}}_i \in \tilde{V}$. We can then restrict ourselves to a cube $V \subset \tilde{V}$ that contains $\tilde{\mathbf{y}}_i$ which can be written as $V = V_1 \times \cdots \times V_n$ where each V_a is an open interval in $\mathbb{R}$.

We can chose U such that T_i has only one minimum that is reached at y_i. This is possible because $T_i' \neq 0$ almost everywhere by hypothesis. Similarly, we chose the cube V such that each $h_{i,a}$ either has only one minimum that is

reached at $\tilde{y}_{i,a}$, or is constant (possible by setting $A_{i,a} = 0$). Define

$$m_i = \min_{\mathbf{z} \in V} T_i \circ v_i(\mathbf{z}) \in \mathbb{R}, \tag{2.47}$$

$$\mu_{i,a} = \min_{z_a \in V_a} h_{i,a}(z_a) \in \mathbb{R}, \tag{2.48}$$

for which we have $m_i = \sum_a \mu_{i,a}$.

Define the sets $C_i = \{\mathbf{z} \in V | T_i \circ v_i(\mathbf{z}) = m_i\}$, $\tilde{C}_{i,a} = \{\mathbf{z} \in V | h_{i,a}(z_a) = \mu_{i,a}\}$ and $\tilde{C}_i = \cap_a \tilde{C}_{i,a}$. We trivially have $\tilde{C}_i \subset C_i$. Next, we prove that $C_i \subset \tilde{C}_i$. Let $\mathbf{z} \in C_i$, and suppose $\mathbf{z} \notin \tilde{C}_i$. Then there exist an index k, $\varepsilon \in \mathbb{R}$ and $\tilde{\mathbf{z}} = (z_1, \ldots, z_k + \varepsilon, \ldots, z_n)$ such that $m_i = \sum_a h_{i,a}(z_a) > \sum_a h_{i,a}(\tilde{z}_a) \geq \sum_a \mu_{i,a} = m_i$ which is not possible. Thus $\mathbf{z} \in \tilde{C}_i$. Hence, $\tilde{C}_i = C_i$.

Since m_i is only reached at y_i, we have $C_i = \{\mathbf{z} \in V | v_i(\mathbf{z}) = y_i\}$. By hypothesis, v_i is of class $\mathcal{C}^1$, and its Jacobian is nonzero everywhere on V (by invertibility of $\mathbf{v}$). Then, by Corollary 5.14 in Lee (2003), we conclude that C_i is a smooth ($\mathcal{C}^1$) submanifold of co-dimension 1 in $\mathbb{R}^n$, and so is $\tilde{C}_i$ by equality.

On the other hand, if $h_{i,a}$ is not constant, then it reaches its minimum $\mu_{i,a}$ at only one point $\tilde{y}_{i,a}$ in V_a. In this case, $\tilde{C}_{i,a} = V_{[1,i-1]} \times \{\tilde{y}_{i,a}\} \times V_{[i+1,n]}$. Suppose that there exist two different indices $a \neq b$, such that $h_{i,a}$ and $h_{i,b}$ are not constant. Then $\tilde{C}_{i,a} \cap \tilde{C}_{i,b}$ is a submanifold of co-dimension 2. This would contradict the fact that the co-dimension of $\tilde{C}_i$ is 1.

Thus, exactly one of the $h_{i,a}$ is not constant for each i. This implies that the i-th row of matrix $\mathbf{A}$ has exactly one nonzero entry. The nonzero entry should occupy a different position in each row to guarantee invertibility, which proves that $\mathbf{A}$ is a scaled permutation matrix. Plugging this back into equation (2.35) implies that $\tilde{\mathbf{f}} \circ \mathbf{f}$ is a pointwise nonlinearity. □

Proof of Proposition 2.13. For simplicity, denote

$$Q(\mathbf{z}) \coloneqq \prod_i Q_i(z_i),$$

$$Z(\mathbf{u}) \coloneqq \prod_i Z_i(\mathbf{u}).$$

Let $\mathbf{A}$ be an orthogonal matrix and $\tilde{\mathbf{z}} = \mathbf{A}\mathbf{z}$ It is easy to check that $\tilde{\mathbf{z}} \sim p_{\tilde{\boldsymbol{\theta}}}(\tilde{\mathbf{z}}|\mathbf{u})$ where this new exponential family is defined by the quantities $\tilde{Q} = Q$, $\tilde{\mathbf{T}} = \mathbf{T}$, $\tilde{\boldsymbol{\lambda}} = \mathbf{A}\boldsymbol{\lambda}$ and $\tilde{Z} = Z$. In particular, the base measure Q does not change when $Q_i(z_i) = 1$ or $Q_i(z_i) = e^{-z_i^2}$ because such a Q is a rotationally invariant function

of $\mathbf{z}$. Further, we have

$$\langle \mathbf{z}, \boldsymbol{\lambda}(\mathbf{u}) \rangle = \langle \mathbf{A}^\top \tilde{\mathbf{z}}, \boldsymbol{\lambda}(\mathbf{u}) \rangle = \langle \tilde{\mathbf{z}}, \mathbf{A}\boldsymbol{\lambda}(\mathbf{u}) \rangle = \langle \tilde{\mathbf{z}}, \tilde{\boldsymbol{\lambda}}(\mathbf{u}) \rangle. \tag{2.49}$$

Finally let $\tilde{\mathbf{f}} = \mathbf{f} \circ \mathbf{A}^\top$, and $\tilde{\boldsymbol{\theta}} \coloneqq \left(\tilde{\mathbf{f}}, \tilde{\mathbf{T}}, \tilde{\boldsymbol{\lambda}}\right)$. We get:

$$\begin{aligned}
p_{\boldsymbol{\theta}}(\mathbf{x}|\mathbf{u}) &= \int p_{\boldsymbol{\theta}}(\mathbf{x}|\mathbf{z}) p_{\boldsymbol{\theta}}(\mathbf{z}|\mathbf{u}) \mathrm{d}\mathbf{z} && (2.50)\\
&= \int p_\varepsilon(x - \mathbf{f}(\mathbf{z})) \frac{Q(\mathbf{z})}{Z(\mathbf{u})} \exp(\langle \mathbf{z}, \boldsymbol{\lambda}(\mathbf{u}) \rangle) \mathrm{d}\mathbf{z} && (2.51)\\
&= \int p_\varepsilon(x - \tilde{\mathbf{f}}(\tilde{\mathbf{z}})) \frac{\tilde{Q}(\tilde{\mathbf{z}})}{\tilde{Z}(\mathbf{u})} \exp(\langle \tilde{\mathbf{z}}, \tilde{\boldsymbol{\lambda}}(\mathbf{u}) \rangle) \mathrm{d}\tilde{\mathbf{z}} && (2.52)\\
&= p_{\tilde{\boldsymbol{\theta}}}(\mathbf{x}|\mathbf{u}), && (2.53)
\end{aligned}$$

where in equation (2.52) we made the change of variable $\tilde{\mathbf{z}} = \mathbf{A}\mathbf{z}$, and removed the Jacobian because it is equal to 1. We then see that it is not possible to distinguish between $\boldsymbol{\theta}$ and $\tilde{\boldsymbol{\theta}}$ based on the observed data distribution. □

2.B.2 Identifiability under alternative assumptions

The weak identifiability of Theorem 2.9 can be derived under an alternative set of assumptions, as is summarized by the following result.

Theorem 2.22. *Assume that we observe data sampled from a generative model defined according to (2.5)-(2.9), with parameters $(\mathbf{f}, \mathbf{T}, \boldsymbol{\lambda})$. Assume the following holds:*

(i) The set $\{\mathbf{x} \in \mathcal{X} | \varphi_\varepsilon(\mathbf{x}) = 0\}$ has measure zero, where φ_ε is the characteristic function of the density p_ε defined in (2.6).

(ii) The mixing function $\mathbf{f}$ in (2.6) is injective.

(iii) The sufficient statistics $T_{i,j}$ in (2.9) are differentiable almost everywhere, and $T'_{i,j} \neq 0$ almost everywhere for all $1 \leq i \leq n$ and $1 \leq j \leq k$.

(iv) $\boldsymbol{\lambda}$ is differentiable, and there exists $\mathbf{u}_0 \in \mathcal{U}$ such that $\mathbf{J}_{\boldsymbol{\lambda}}(\mathbf{u}_0)$ is invertible.

then the parameters $(\mathbf{f}, \mathbf{T}, \boldsymbol{\lambda})$ are $\sim$-identifiable. Moreover, if there exists $(\tilde{\mathbf{f}}, \tilde{\mathbf{T}}, \tilde{\boldsymbol{\lambda}})$ such that $p_{\tilde{\mathbf{f}}, \tilde{\mathbf{T}}, \tilde{\boldsymbol{\lambda}}}(\mathbf{x}|\mathbf{u}) = p_{\mathbf{f}, \mathbf{T}, \boldsymbol{\lambda}}(\mathbf{x}|\mathbf{u})$, then $\tilde{\mathbf{T}}$ and $\tilde{\boldsymbol{\lambda}}$ verify assumptions (iii) and (iv).

Proof. The start of the proof is similar to the proof of Theorem 2.9. When we get to equation (2.32):

$$\log \operatorname{vol} \mathbf{J}_{\mathbf{f}^{-1}}(\mathbf{x}) + \sum_{i=1}^{n} (\log Q_i(f_i^{-1}(\mathbf{x})) - \log Z_i(\mathbf{u}) + \sum_{j=1}^{k} T_{i,j}(f_i^{-1}(\mathbf{x}))\lambda_{i,j}(\mathbf{u})) =$$
$$\log \operatorname{vol} \mathbf{J}_{\tilde{\mathbf{f}}^{-1}}(\mathbf{x}) + \sum_{i=1}^{n} (\log \tilde{Q}_i(\tilde{f}_i^{-1}(\mathbf{x})) - \log \tilde{Z}_i(\mathbf{u}) + \sum_{j=1}^{k} \tilde{T}_{i,j}(\tilde{f}_i^{-1}(\mathbf{x}))\tilde{\lambda}_{i,j}(\mathbf{u})). \tag{2.54}$$

We take the derivative of both sides with respect to $\mathbf{u}$ (assuming that $\tilde{\boldsymbol{\lambda}}$ is also differentiable). All terms depending on $\mathbf{x}$ only disappear, and we are left with:ivae:

$$\mathbf{J}_{\boldsymbol{\lambda}}(\mathbf{u})^\top \mathbf{T}(\mathbf{f}^{-1}(\mathbf{x})) - \sum_i \nabla \log Z_i(\mathbf{u}) = \mathbf{J}_{\tilde{\boldsymbol{\lambda}}}(\mathbf{u})^\top \tilde{\mathbf{T}}(\tilde{\mathbf{f}}^{-1}(\mathbf{x})) - \sum_i \nabla \log \tilde{Z}_i(\mathbf{u}). \tag{2.55}$$

By evaluating both sides at $\mathbf{u}_0$ provided by assumption (iv), and multiplying both sides by $\mathbf{J}_{\boldsymbol{\lambda}}(\mathbf{u}_0)^{-\top}$ (invertible by hypothesis), we find:

$$\mathbf{T}(\mathbf{f}^{-1}(\mathbf{x})) = \mathbf{A}\tilde{\mathbf{T}}(\tilde{\mathbf{f}}^{-1}(\mathbf{x})) + \mathbf{c}, \tag{2.56}$$

where $\mathbf{A} = \mathbf{J}_{\boldsymbol{\lambda}}(\mathbf{u}_0)^{-\top} \mathbf{J}_{\tilde{\boldsymbol{\lambda}}}(\mathbf{u}_0)^\top$ and $\mathbf{c} = \sum_i \nabla \log \frac{Z_i(\mathbf{u}_0)}{\tilde{Z}_i(\mathbf{u}_0)}$. The rest of the proof follows proof of Theorem 2.9, where in the last part we deduce that $\mathbf{J}_{\tilde{\boldsymbol{\lambda}}}(\mathbf{u}_0)$ is invertible. □

2.B.3 Consistency proof

Proof of Theorem 2.14. The loss (2.10) can be written as follows:

$$\mathcal{L}(\boldsymbol{\theta}, \boldsymbol{\phi}) = \log p_{\boldsymbol{\theta}}(\mathbf{x}|\mathbf{u}) - \mathrm{KL}(q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x}, \mathbf{u}) \| p_{\boldsymbol{\theta}}(\mathbf{z}|\mathbf{x}, \mathbf{u})). \tag{2.57}$$

If the family $q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x}, \mathbf{u})$ is large enough to include $p_{\boldsymbol{\theta}}(\mathbf{z}|\mathbf{x}, \mathbf{u})$, then by optimizing the loss over its parameter $\boldsymbol{\phi}$, we will minimize the KL term, eventually reaching zero, and the loss will be equal to the log-likelihood. The VAE in this case inherits all the properties of maximum likelihood estimation. In this particular case, since our identifiability is guaranteed up to equivalence classes, the consistency of MLE means that we converge to the equivalence class (Theorem 2.9) of true parameter $\boldsymbol{\theta}^*$ *i.e.* in the limit of infinite data. This is easy to show because true identifiability is one of the assumptions for MLE consistency, replacing it by identifiability up to equivalence class doesn't change the proof but only the conclusion. □

2.C Unidentifiability of generative models with unconditional prior

In this section, we present two well-known proofs of unidentifiability of generative models. The first proof is simpler and considers factorial priors, which are widely-used in deep generative models and the VAE literature. The second proof is extremely general, and shows how any random vector can be transformed into independent components, in particular components which are standardized Gaussian. Thus, we see how in the general nonlinear case, there is little hope of finding the original latent variables based on the (unconditional, marginal) statistics of $\mathbf{x}$ alone.

2.C.1 Factorial priors

Let us start with factorial, Gaussian priors. In other words, let $\mathbf{z} \sim p_{\boldsymbol{\theta}}(\mathbf{z}) = N(\mathbf{0}, \mathbf{I})$. Now, a well-known result says that any orthogonal transformation of $\mathbf{z}$ has exactly the same distribution. Thus, we could transform the latent variable by any orthogonal transformation $\mathbf{z}' = \mathbf{M}\mathbf{z}$, and cancel that transformation in $p(\mathbf{x}|\mathbf{z})$ (e.g. in the first layer of the neural network), and we would get exactly the same observed data (and thus obviously the same distribution of observed data) with $\mathbf{z}'$.

Formally we have

$$p_{\mathbf{z}'}(\boldsymbol{\xi}) = p_{\mathbf{z}}(\mathbf{M}^\top \boldsymbol{\xi}) |\det \mathbf{M}| = \frac{1}{(2\pi)^{d/2}} \exp(-\frac{1}{2}\|\mathbf{M}^\top \boldsymbol{\xi}\|^2) \tag{2.58}$$

$$= \frac{1}{(2\pi)^{d/2}} \exp(-\frac{1}{2}\|\boldsymbol{\xi}\|^2) = p_{\mathbf{z}}(\boldsymbol{\xi}), \tag{2.59}$$

where we have used the fact that the determinant of an orthogonal matrix is equal to unity.

This result applies easily to any factorial prior. For z_i of any distribution, we can transform it to a uniform distribution by $F_i(z_i)$ where F_i is the cumulative distribution function of z_i. Next, we can transform it into standardized Gaussian by $\Phi^{-1}(F_i(z_i))$ where Φ is the standardized Gaussian CDF. After this transformation, we can again take any orthogonal transformation without changing the distribution. And we can even transform back to the

same marginal distributions by $F_i^{-1}(\Phi(.))$. Thus, the original latents are not identifiable.

2.C.2 General priors

The second proof comes from the theory of nonlinear ICA (Hyvärinen and Pajunen, 1999), from which the following theorem is adapted.

Theorem 2.23 (Hyvärinen and Pajunen (1999), Theorem 1)**.** *Let* $\mathbf{z}$ *be a* d*-dimensional random vector of any distribution. Then there exists a transformation* $\mathbf{g} : \mathbb{R}^d \to \mathbb{R}^d$ *such that the components of* $\mathbf{z}' \coloneqq \mathbf{g}(\mathbf{z})$ *are independent, and each component has a standardized Gaussian distribution. In particular,* z_1' *equals a monotonic transformation of* z_1*.*

The proof is based on an iterative procedure reminiscent of Gram-Schmidt, where a new variable can always be transformed to be independent of any previously considered variables, which is why z_1 is essentially unchanged.

This Theorem means that there are infinitely many ways of defining independent components $\mathbf{z}$ that nonlinearly generated an observation $\mathbf{x}$. This is because we can first transform $\mathbf{z}$ any way we like and then apply the Theorem. The arbitrariness of the components is seen in the fact that we will always find that one arbitrary chosen variable in the transformation is one of the independent components. This is in some sense an alternative kind of indeterminacy to the one in the previous section.

In particular, we can even apply this Theorem on the observed data, taking $\mathbf{x}$ instead of $\mathbf{z}$. Then, in the case of factorial priors, just permuting the data variables, we would arrive at the conclusion that any of the x_i can be taken to be one of the independent components, which is absurd.

Now, to apply this theory in the case of a general prior on $\mathbf{z}$, it is enough to point out that we can transform any variable into independent Gaussian variables, apply any orthogonal transformation, then invert the transformation in the Theorem, and we get a nonlinear transformation $\mathbf{z}' = \mathbf{g}^{-1}(\mathbf{M}\mathbf{g}(\mathbf{z}))$ which has exactly the same distribution as $\mathbf{z}$ but is a complex nonlinear transformation. Thus, no matter what the prior may be, by looking at the data alone, it is not possible to recover the true latents based an unconditional prior distribution, in the general nonlinear case.

2.D Identifiability up to equivalence class: examples

As an illustration of identifiability up to equivalence class, let's consider the identifiability in linear ICA.

Example 2.24 (Identifiability of linear ICA)**.** Reconsider the linear ICA setting, where we have observations that are a linear mixing of independent source variables:

$$\begin{aligned} \mathbf{x} &= \mathbf{A}\mathbf{s}, \\ p_{\mathbf{s}}(\mathbf{s}) &= \prod_i p_i(s_i), \end{aligned} \tag{2.60}$$

where $\mathbf{A} \in \mathbb{R}^{d \times d}$ is an orthogonal mixing matrix (we can always transform the linear ICA problem into one where the mixing is orthogonal — this is called whitening (see Hyvärinen et al., 2001, for more details)).

The theory of linear ICA says that a solution to the problem (2.60) exists if at most one of the components s_i is normally distributed. This solution is unique up to a permutation and a scaling of the components, as stated by Theorem 1.1.

To formalize this in terms of an equivalence relation, let $\mathcal{F}_{\text{ICA}} = \mathcal{A} \times \mathcal{P}$, where $\mathcal{A}$ is the set of orthogonal matrices and $\mathcal{P}$ is the set of distributions with at most one Gaussian marginal. We define the following equivalence relation on $\mathcal{F}_{\text{ICA}}$:

$$(\mathbf{A}, p) \sim_{\text{ICA}} (\tilde{\mathbf{A}}, \tilde{p}) \iff \exists \mathbf{D}, \mathbf{P} \text{ s.t. } (\mathbf{A}, p) = \left(\tilde{\mathbf{A}}\mathbf{P}^{-1}\mathbf{D}^{-1}, (\mathbf{D}\mathbf{P})_{\#}\tilde{p}\right), \tag{2.61}$$

where $\mathbf{D}$ is a diagonal matrix, $\mathbf{P}$ is a permutation matrix, and $(\mathbf{D}\mathbf{P})_{\#}\tilde{p}$ is the push-forward density of $\tilde{p}$ by $\mathbf{D}\mathbf{P}$. The equivalence relation $\sim_{\text{ICA}}$ characterizes the solutions to the linear ICA problem in terms of the equivalence class of the true generative model $(\mathbf{A}^*, p^*)$. In other words, $(\mathbf{A}, p)$ is a solution of (2.60) if and only if $(\mathbf{A}, p) \sim_{\text{ICA}} (\mathbf{A}^*, p^*)$. □

Another useful example of equivalence class is in representation learning, where we often use a neural network to learn a set of features that are subsequently used in a classification task. Features can be considered equivalent if they do not change the boundaries of this task.

Example 2.25 (Indeterminacy of the parameters in linear classification)**.** Given a set of observations and binary labels $\mathcal{D} = \left\{(\mathbf{x}^1, y^1), \ldots, (\mathbf{x}^N, y^N)\right\}$, we seek to learn a classifier to predict a label y from an observation $\mathbf{x}$. To this end, we consider a simple liner classifier f with weight vector $\mathbf{w}$, combined with a feature extractor $\boldsymbol{\phi}$:

$$f(\mathbf{x}; \mathbf{w}, \boldsymbol{\phi}) = \mathbf{w}^\top \boldsymbol{\phi}(\mathbf{x}).$$

We ignore the bias term as it can simply be absorbed into $\mathbf{w}$.

The vector $\boldsymbol{\phi}(\mathbf{x})$ represents a set of features that are learnt from the observations, and can tremendously improve the flexibility of the linear classifier. In fact, it is common in the representation learning community to learn sophisticated features from data, which are then used in a simple downstream classification or regression task. Here, we will similarly consider that learning the feature extractor is decoupled from the classification task, where we only look to find an optimal $\mathbf{w}$.

Learning the weights $\mathbf{w}$ is often done by solving

$$\min_{\mathbf{w}} \frac{1}{N} \sum_{i=1}^{N} L\left(f(\mathbf{x}^i; \mathbf{w}, \boldsymbol{\phi}), y^i\right) + \lambda \|\mathbf{w}\|_2, \tag{2.62}$$

where λ is a hyperparameter which ensures that the norm of the weights is well controlled, and L is a loss function that measures the discrepancy between the score $f^i \coloneqq f(\mathbf{x}^i; \mathbf{w}, \boldsymbol{\phi})$ and the label y^i. An example of such loss is the hinge loss used in support vector machines: $L(f^i, y^i) = \max(0, 1 - y^i f^i)$.

Denote by $\mathbf{w}^*$ the solution to the above classification problem when using the features $\boldsymbol{\phi}$. Now suppose that we used another feature extractor $\tilde{\boldsymbol{\phi}}$ that is equal to $\boldsymbol{\phi}$ up to an orthogonal linear transformation: $\tilde{\boldsymbol{\phi}} = \mathbf{A}\boldsymbol{\phi}$. Then the optimal solution $\tilde{\mathbf{w}}^*$ to the classification problem (2.62) when using $\tilde{\boldsymbol{\phi}}$ is related to $\mathbf{w}^*$ via: $\tilde{\mathbf{w}}^* = \mathbf{A}\mathbf{w}^*$. The pairs $(\tilde{\mathbf{w}}^*, \tilde{\boldsymbol{\phi}})$ and $(\mathbf{w}^*, \boldsymbol{\phi})$ both impose the same boundary between the two classes in this binary classification task.

Thus, we can define an equivalence relation $\sim_{\text{lin}}$ over the set of admissible weights and feature extractors:

$$(\tilde{\mathbf{w}}, \tilde{\boldsymbol{\phi}}) \sim_{\text{lin}} (\mathbf{w}, \boldsymbol{\phi}) \iff (\tilde{\mathbf{w}}, \tilde{\boldsymbol{\phi}}) = (\mathbf{A}\mathbf{w}, \mathbf{A}\boldsymbol{\phi}). \tag{2.63}$$

The equivalence relation $\sim_{\text{lin}}$ characterizes the solutions of this supervised classification task in the sense that multiplying the feature extractor by a linear transformation doesn't change the solution to the problem. □

2.E Link between maximum likelihood and total correlation

Consider the noiseless case:

$$\mathbf{x} = \mathbf{f}(\mathbf{z}), \tag{2.64}$$

$$p(\mathbf{z}|\mathbf{u}) = \prod_i p_i(z_i|\mathbf{u}), \tag{2.65}$$

where the components of the latent variable are independent given the auxiliary variable $\mathbf{u}$. We can relate the log-likelihood of the data to the total correlation of the latent variables. To see this connection, let's use the change of variable formula in the expression of the log-likelihood:

$$\mathbb{E}_{p(\mathbf{x},\mathbf{u})}\left[\log p(\mathbf{x}|\mathbf{u})\right] = \mathbb{E}_{p(\mathbf{z},\mathbf{u})}\left[\sum_i \log p_i(z_i|\mathbf{u}) - \log\left|\mathbf{J}_\mathbf{f}(\mathbf{z})\right|\right] \tag{2.66}$$

$$= -\mathbb{E}_{p(\mathbf{z},\mathbf{u})}\left[\log\left|\mathbf{J}_\mathbf{f}(\mathbf{z})\right|\right] - \sum_i H(z_i|\mathbf{u}), \tag{2.67}$$

where $H(z_i|\mathbf{u})$ is the conditional differential entropy of z_i given $\mathbf{u}$. The same change of variable formula applied to $H(\mathbf{x}|\mathbf{u})$ yields:

$$H(\mathbf{x}|\mathbf{u}) = H(\mathbf{z}|\mathbf{u}) + \mathbb{E}_{p(\mathbf{z},\mathbf{u})}\left[\log\left|\mathbf{J}_\mathbf{f}(\mathbf{z})\right|\right], \tag{2.68}$$

which we then use in the expression of the conditional total correlation:

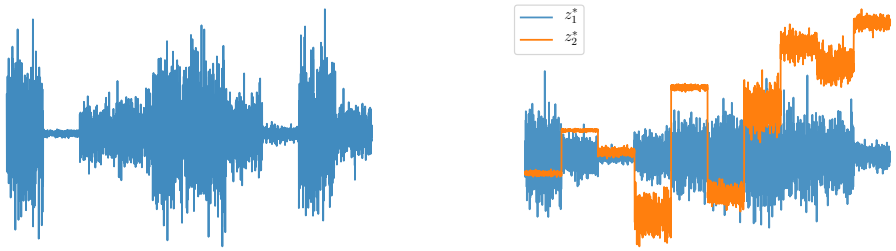
$$\begin{aligned} \mathrm{TC}(\mathbf{z}|\mathbf{u}) &\coloneqq \sum_i H(z_i|\mathbf{u}) - H(\mathbf{z}|\mathbf{u}) \\ &= \sum_i H(z_i|\mathbf{u}) - H(\mathbf{x}|\mathbf{u}) + \mathbb{E}_{p(\mathbf{z},\mathbf{u})}\left[\log\left|\mathbf{J}_\mathbf{f}(\mathbf{z})\right|\right]. \end{aligned} \tag{2.69}$$

Putting equations (2.67) and (2.69) together, we get:

$$\mathbb{E}_{p(\mathbf{x},\mathbf{u})}[\log p(\mathbf{x}|\mathbf{u})] = -\,\mathrm{TC}(\mathbf{z}|\mathbf{u}) - H(\mathbf{x}|\mathbf{u}). \tag{2.70}$$

The last term in this equation is a function of the data only and is thus a constant. An algorithm which learns to maximize the data likelihood is decreasing the total correlation of the latent variable. The total correlation is measure of independence as it is equal to zero if and only if the components of the latent variable are independent. Thus, by using a VAE to maximize a lower bound on the data likelihood, we are trying to learn an estimate of the inverse of the mixing function that gives the most independent components.

2.F Experimental protocol and additional experiments



(a) (b)

Figure 2.7: Visualization of various sources following the generative distribution detailed in equation (2.9). (*a*) single source with segment modulated variance; (*b*) two sources where the mean of the second source, z_2^*, is significantly modulated as a function of the segment, thus potentially serving to greatly facilitate the surrogate classification task performed in TCL.

2.F.1 Details of implementation for VAE experiments

We give here more detail on the data generation process for our simulations. The dataset is described in Section 2.5.2. The conditioning variable $\mathbf{u}$ (subsequently denoted u because it is scalar) is the segment label, and its distribution is uniform on the integer set $[\![1, M]\!]$. Within each segment, the conditional prior distribution is chosen from the family (2.9), where $k = 1$, $T_{i,1}(z_i) = z_i^2$ and $Q_i(z_i) = 1$, and the true $\boldsymbol{\lambda}_i$ were randomly and independently generated across the segments and the components so that the variances have a uniform distribution on $[.5, 3]$. We sample latent variable $\mathbf{z}$ from these distribution, and then mix them using a 4-layer multi-layer perceptron (MLP). An example of what the sources look like is plotted in Figure [2.7a]. We finally add small noise ($\sigma^2 = 0.01$) to the observations. When comparing to previous ICA methods, we omit this step, as these methods are for the noiseless case.

For the decoder (2.6), we chose $p_\varepsilon = \mathcal{N}(0, \sigma_\varepsilon^2 I)$ a zero mean Gaussian, where the scalar σ_ε^2 controls the noise level. We fix the noise level $\sigma^2 = 0.01$. As for the inference model, we let $q_\phi(\mathbf{z}|\mathbf{x}, u) = \mathcal{N}(\mathbf{z}|\mathbf{g}(\mathbf{x}, u; \boldsymbol{\phi}_\mathbf{g}), \operatorname{diag} \boldsymbol{\sigma}^2(\mathbf{x}, u; \boldsymbol{\phi}_{\boldsymbol{\sigma}}))$ be

a multivariate Gaussian with a diagonal covariance. The functional parameters of the decoder ($\mathbf{f}$) and the inference model ($\mathbf{g}$, $\boldsymbol{\sigma}^2$) as well as the conditional prior ($\boldsymbol{\lambda}$) are chosen to be MLPs, where the dimension of the hidden layers is varied between 10 and 200, the activation function is a leaky ReLU, and the number of layers is chosen from $\{3, 4, 5, 6\}$. Mini-batches are of size 64, and the learning rate of the Adam optimizer is chosen from $\{0.01, 0.001\}$. We also use a scheduler to decay the learning rate as a function of epochs.

To implement the VAE, we followed Kingma and Welling (2014). We made sure the range of the hyperparameters (mainly number of layers and dimension of hidden layers) of the VAE is large enough for it to be comparable in complexity to our method (which has the extra λ network to learn). To implement a β-VAE, we followed the instructions of Higgins et al. (2017) for the choice of hyperparameter β, which was chosen in the set $[1, 45]$. Similarly, we followed Chen et al. (2018) for the choice of the hyperparameters α, β and γ when implementing a β-TC-VAE: we chose $\alpha = \gamma = 1$ and β was chosen in the set $[1, 35]$.

2.F.2 Hippocampal fMRI data

Here we provide further details relating to the resting-state Hippocampal data provided by Poldrack et al. (2015) and studied in Section 2.5.3, closely following the earlier causal work using TCL by Monti et al. (2019). The data corresponds to daily fMRI scans from a single individual (Caucasian male, aged 45) collected over a period of 84 successive days. We consider data collected from each day as corresponding to a distinct segment, encoded in $\mathbf{u}$. Within each day 518 BOLD observations are provided across the following six brain regions: perirhinal cortex (PRc), parahippocampal cortex (PHc), entorhinal cortex (ERc), subiculum (Sub), CA1 and CA3/Dentate Gyrus (DG).

2.F.3 Additional experiments

As a further visualization, we compare iVAE to VAE on a series of experiments on real and simulated data. The results reported in Figures [2.8] to [2.15] show that iVAE is better suited for learning identifiable representations than a basic VAE.

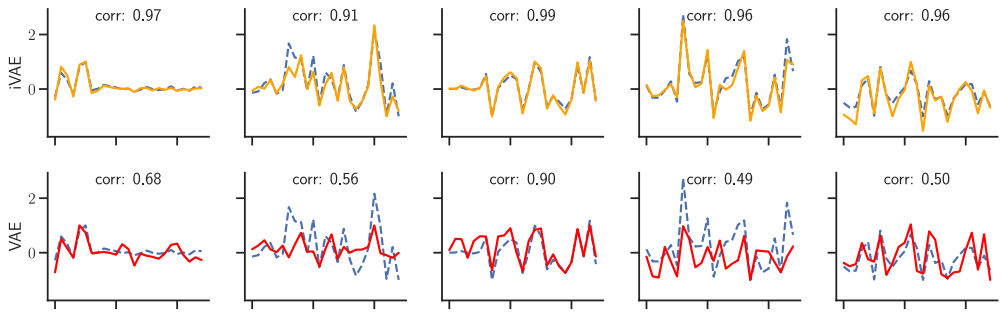


Figure 2.8: Comparison of the recovered latents of our model to the latents recovered by a vanilla VAE. The dashed blue line is the true source signal, and the recovered latents are in solid coloured lines. We also reported the correlation coefficients for every (source, latent) pair.

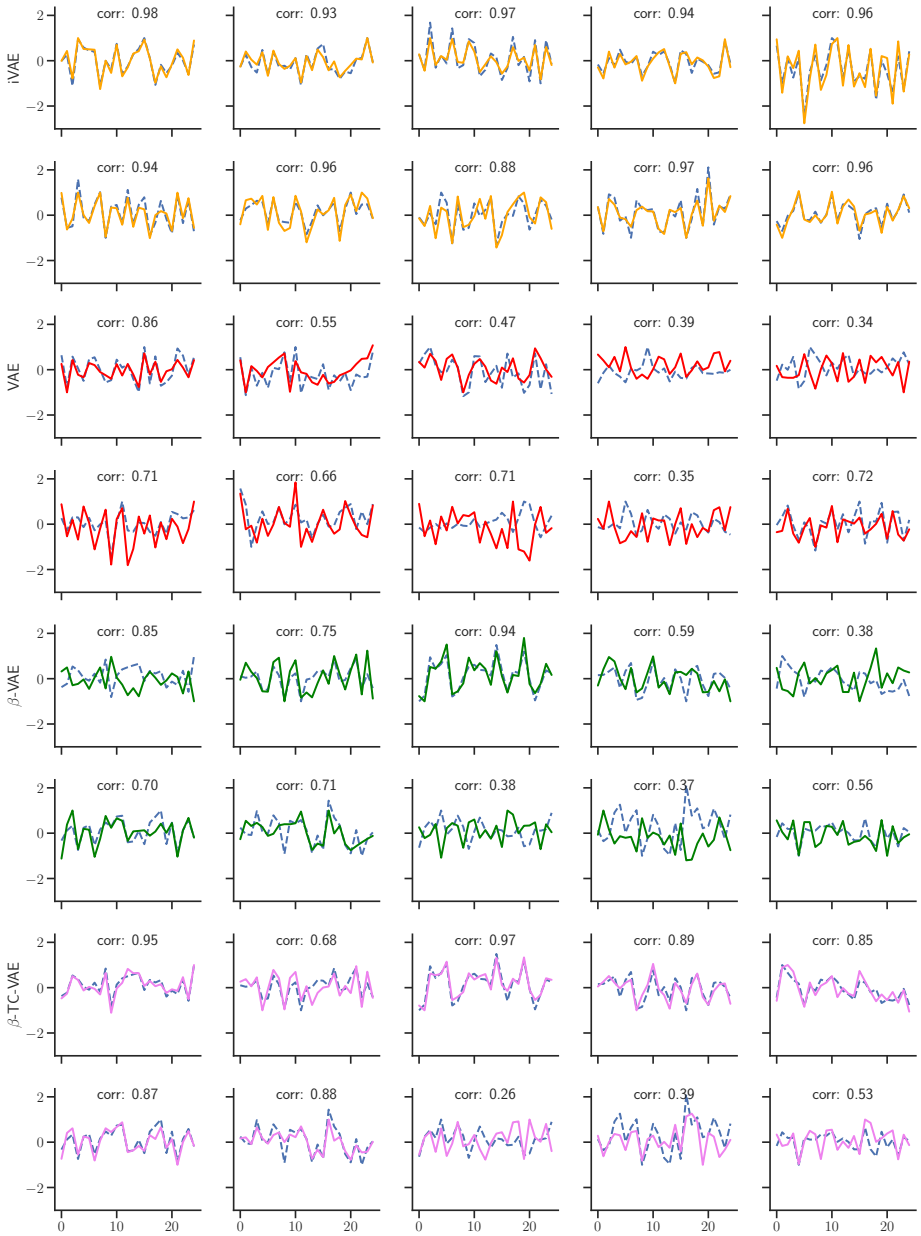
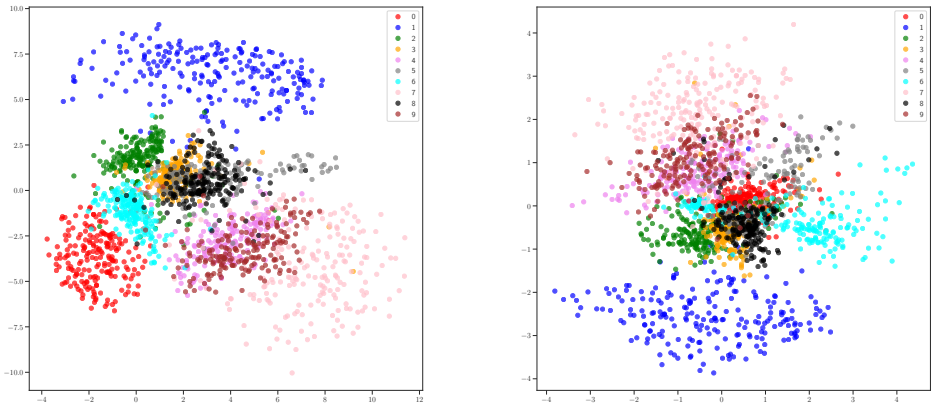
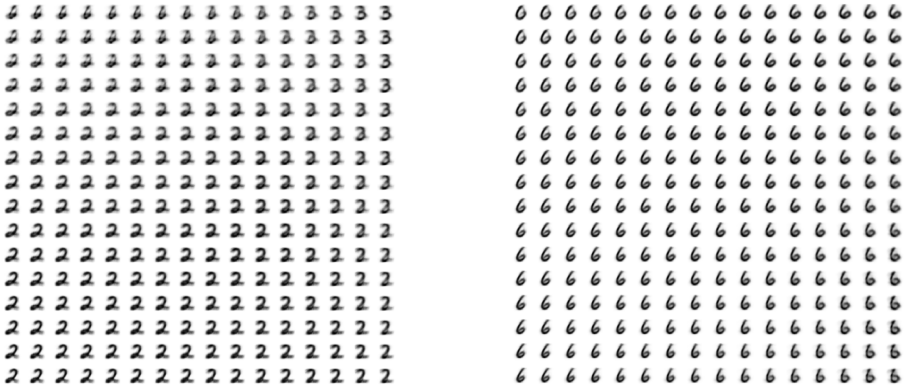


Figure 2.9: Comparison of the recovered latents of our model to the latents recovered by a vanilla VAE, a β-VAE and a β-TC-VAE, where the dimension of the data is $d = 40$, and the dimension of the latents is $n = 10$, the number of segments is $M = 40$ and the number of samples per segment is $L = 4000$. The dashed blue line is the true source signal, and the recovered latents are in solid coloured lines. We reported the correlation coefficients for every (source, latent) pair.



(a) iVAE (b) VAE

Figure 2.10: MNIST: Visualization of the latent space in 2D. The iVAE latents occupy more space, and are slightly more separated than the VAE latents, especially in the centre near zero. We still see the same clustering of classes for both models.

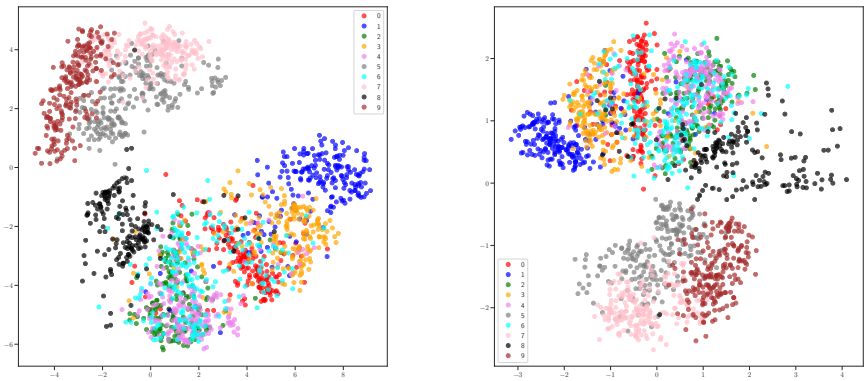


(a) $u = 2$ (b) $u = 6$

Figure 2.11: MNIST: Visualizations of the data manifold generated from a traversal of the latent space in 2D, conditioned on $u = 2$ and $u = 6$. The traversal is achieved by transforming the unit square through the inverse cumulative distribution function of a Gaussian parametrized by the learned means and variances for each class.

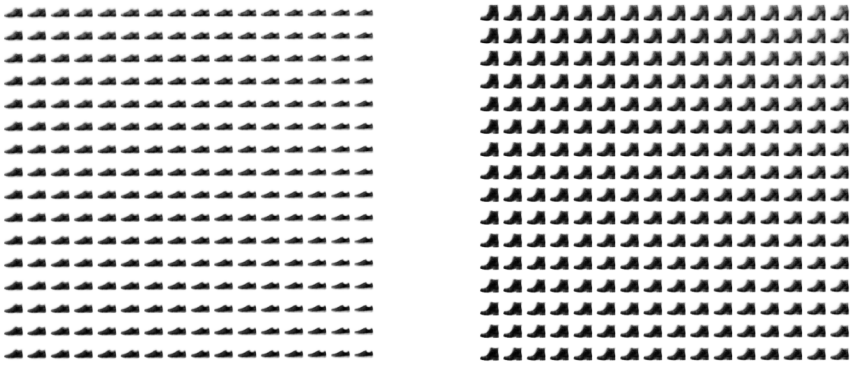
(a) VAE (b) iVAE (c) restricted iVAE

Figure 2.12: MNIST: Samples from the learned generative model after learning for 10 epochs. *a*) We sample from the Gaussian prior, and then through the decoder *b*) We sample labels uniformly between 1 and 9, then sample from the conditional Gaussian prior, and finally through the decoder. *c*) We sample latents from the conditional prior conditioned on $u = 4$, then through the decoder.



(a) iVAE (b) VAE

Figure 2.13: FMNIST: Visualization of the latent space in 2D. Similarly to MNIST, the iVAE latents occupy more space, and are slightly more separated than the VAE latents.



(a) $u = 7$ (b) $u = 9$

Figure 2.14: FMNIST: Visualizations of the data manifold generated from a traversal of the latent space in 2D, conditioned on $u = 7$ and $u = 9$. The traversal is achieved by transforming the unit square through the inverse cumulative distribution function of a Gaussian parametrized by the learned means and variances for each class. *a*) The latent dimensions encode the shoe type (from sneakers to dress shoes) and the shoe size (from small to big). *b*) The y dimension encodes the heel size, but the x dimension doesn't seem to encode any information about the data.

(a) VAE (b) iVAE (c) restricted iVAE

Figure 2.15: FMNIST: Samples from the learned generative model after learning for 10 epochs. *a*) We sample from the Gaussian prior, and then through the decoder *b*) We sample labels uniformly between 1 and 9, then sample from the conditional Gaussian prior, and finally through the decoder. *c*) We sample latents from the conditional prior conditioned on one class, then through the decoder.

Chapter 3

Independently modulated component analysis

The identifiability of nonlinear probabilistic models has been receiving considerable attention lately in the machine learning community. The most recent identifiability results for latent variable models can be found in the nonlinear ICA literature. Most notably, these results required that the latent variables are independent given an additionally observed auxiliary variable. However, the need for (conditional) independence can sometimes be seen as a limitation, for example in the context of learning disentangled representations. In the present chapter, we prove that we can relax the assumption of independence while maintaining identifiability, providing a very flexible and general framework for principled disentangled representation learning. We show that the access to an auxiliary variable ensures a weak form of identifiability regardless of independence. More importantly, we introduce the novel Independently Modulated Component Analysis (IMCA) framework: it requires that conditioning by an auxiliary variable changes the joint distribution of the latents in a factorizable way, while allowing them to share an arbitrary base measure. IMCA enjoys the same identifiability guarantees as nonlinear ICA, while being more general and flexible.

This chapter is based on Khemakhem et al. (2020b).

3.1 Introduction

Unsupervised feature learning is one of the fundamental challenges in machine learning. Whilst such a field has recently enjoyed great empirical success, it is widely acknowledged that existing methods do not enjoy strong theoretical guarantees. An increasingly important area for theoretical research is to demonstrate that specific models are identifiable, which means that the true parameters of the underlying model can be uniquely recovered in the limit of infinite data.

Within representation learning, results relating to identifiability were constrained to consider linear latent variable models and mainly studied within the context of independent component analysis (Comon, 1994, ICA), which assumes that the observed variable $\mathbf{x}$ is a result of a mixing of source variables $\mathbf{z}$ with independent components. It was acknowledged that generalizing the theory of ICA to the realm of nonlinear transformations was not possible, primarily due to the flexibility of such maps, which could yield arbitrary latent variables (Hyvärinen and Pajunen, 1999).

Thanks to recent advances, it is now understood that nonlinear latent variable models may also be identifiable given some additional auxiliary variables (Hyvärinen and Morioka, 2016; Hyvärinen and Morioka, 2017; Hyvärinen et al., 2019; Khemakhem et al., 2020a). The purpose of this auxiliary variable serves to introduce additional constraints over the distribution over latent variables, which are typically required to be conditionally independent given the auxiliary variable. More precisely, the latent variables $\mathbf{z}$ have a density of the form $p(\mathbf{z}|\mathbf{u}) = \prod_i p_i(z_i|\mathbf{u})$.

This avenue of research has thus formalized the trade-off between the expressivity of the mapping between latent variables to observations (from linear to nonlinear) and the distributional assumptions over latent variables (from independent to conditionally independent given auxiliary variables). However, hardly any identifiability theory exists for the non-independent case, and the need for (conditional) independence in order to obtain identifiability results remains a limitation. This is particularly the case in disentangled representation learning, where independence is seen as too severe a restriction.

In this chapter, we propose to relax the assumption of independence while maintaining identifiability. This was achieved before in the linear case (Hyväri-

nen and Hurri, 2004; Monti and Hyvärinen, 2018), and we believe it can also be achieved in the nonlinear setting. To this effect, we introduce the Independently Modulated Component Analysis (IMCA) framework: a generative model where the latent variables are *dependent* through an arbitrary base measure, leading to an arbitrary global dependency structure. To achieve identifiability, we assume that the latent joint density has a part that is modulated in a factorizable way when conditioned by an auxiliary variable (such as time index, history, or another data source). The central contribution of this chapter is a thorough analysis of the identifiability of such a model, which generalizes nonlinear ICA.

3.2 Independently modulated component analysis

3.2.1 Definition of the generative model

Assume we observe a random variable $\mathbf{x} \in \mathbb{R}^d$ as a result of a nonlinear transformation $\mathbf{f}$ of a latent variable (also called *source*) $\mathbf{z} \in \mathbb{R}^d$ whose distribution is conditioned on an auxiliary variable $\mathbf{u}$ that is also observed:

$$\begin{aligned} \mathbf{z} &\sim p_{\boldsymbol{\theta}}(\mathbf{z}|\mathbf{u}), \\ \mathbf{x} &= \mathbf{f}(\mathbf{z}). \end{aligned} \tag{3.1}$$

With a slight abuse of notations, we will use $\boldsymbol{\theta}$ to denote the parameters of the prior as well as the mixing function $\mathbf{f}$.

In the nonlinear ICA model of Chapter 2, the prior components $z_i \in \mathbf{z}$ each has a univariate exponential family distribution given $\mathbf{u}$. In this chapter, we look at a specific instance of the multivariate exponential family that generalises the densities produced by a product of univariate exponential families by only having a part of its density factor across the components of the random variable $\mathbf{z}$.

Definition 3.1 (Exponentially factorial distributions)**.** *We say that a multivariate exponential family distribution is* exponentially factorial *if its density* $p(\mathbf{z})$ *has the form*

$$p(\mathbf{z}) = Q(\mathbf{z}) \prod_{i=1}^{d} e^{\mathbf{T}_i(z_i)^\top \boldsymbol{\lambda}_i - \Gamma(\boldsymbol{\lambda})}. \tag{3.2}$$

We similarly define the conditional exponentially factorial distribution by requiring the natural parameter $\boldsymbol{\lambda} := (\boldsymbol{\lambda}_1, \ldots, \boldsymbol{\lambda}_d)$ *to be a function of the conditioning variable* $\mathbf{u}$*:*

$$p(\mathbf{z}|\mathbf{u}) = Q(\mathbf{z}) e^{\sum_i \mathbf{T}_i(z_i)^\top \boldsymbol{\lambda}_i(\mathbf{u}) - \Gamma(\mathbf{u})}. \tag{3.3}$$

The exponential term of an exponentially factorial distribution factors across components. This means that the sufficient statistic is decomposed of d mappings, each function of only one component of the random variable $\mathbf{z}$. We emphasise that the base measure $Q(\mathbf{z})$ is *not* necessarily factorial, which can lead to the latent components $z_i \in \mathbf{z}$ being dependent on each other.

Equations (3.1) and (3.3) together define a nonparametric model with parameters $\boldsymbol{\theta} = (\mathbf{f}, \mathbf{T}, \boldsymbol{\lambda}, Q)$. For the special case $Q(\mathbf{z}) = \prod_i Q_i(z_i)$, the distribution of $\mathbf{z}$ factors across dimensions, and the components z_i are independent. Then the generative model gives rise to a nonlinear ICA model, and it was studied to a great depth in Chapter 2. In particular, that chapter presented sufficient conditions that lead to strong identifiability of the model which allows for the recovery of the latent variables up to permutation and nonlinear scaling.

It is possible to generalise this model by allowing for an arbitrary base measure $Q(\mathbf{z})$, *i.e.* the components of the latent variable are no longer independent, as Q does not necessarily factorise across dimensions. However, it is crucial that the components of the latent variables are independently modulated given the auxiliary variable $\mathbf{u}$, and that through the term $\exp(\sum_i \mathbf{T}_i(z_i)^\top \boldsymbol{\lambda}_i(\mathbf{u}))$.

This new framework is called *Independently Modulated Component Analysis* (IMCA). Section 3.2.3 shows that the strong identifiability guarantees developed for nonlinear ICA can be extended to IMCA, yielding a more general and more flexible principled framework for representation learning and disentanglement.

3.2.2 Identifiability

The concept of identifiability is core to this chapter. As such, it is important to understand the different views one can have of this concept. Recall the definition of identifiability, seen in equation (2.4) of Section 2.4.1:

$$\mathcal{P}_{\boldsymbol{\theta}_1} = \mathcal{P}_{\boldsymbol{\theta}_2} \implies \boldsymbol{\theta}_1 = \boldsymbol{\theta}_2.$$

In other words, a model is identifiable if different values of the parameters must generate different probability distributions. This definition, however, is

very restrictive and impractical. Often, the identifiability form we can prove for a model is equality of the parameters *up to some indeterminacies*. This can be understood as an equivalence relation between parameters, as detailed in Section 2.4.1. For this reason, we will use the concept of identifiability up to equivalence class, introduced in Definition 2.5. Identifiability in this context implies that the equivalence class of the ground truth parameter can be uniquely recovered from observations.

An example of indeterminacy that is relevant to us here comes from the literature on the variational inference of latent variable models: two parameters are equivalent if they map to the same *inference* distribution (Chapter 2). In this chapter, we will say that a generative model is identifiable if we can uniquely recover the latent variables up to two ambiguities, namely a linear mapping and pointwise nonlinear transformations:

Definition 3.2. *Consider two different sets of parameters* $\boldsymbol{\theta} = (\mathbf{f}, \mathbf{T}, \boldsymbol{\lambda}, Q)$ *and* $\boldsymbol{\theta}' = (\mathbf{f}', \mathbf{T}', \boldsymbol{\lambda}', Q')$*, defining two densities* $p_{\boldsymbol{\theta}}$ *and* $p_{\boldsymbol{\theta}'}$*. We say that the IMCA model is strongly identifiable if*

$$\begin{aligned} p_{\boldsymbol{\theta}}(\mathbf{x}|\mathbf{u}) = p_{\boldsymbol{\theta}'}(\mathbf{x}|\mathbf{u}) &\implies \boldsymbol{\theta} \sim_P \boldsymbol{\theta}', \\ \boldsymbol{\theta} \sim_P \boldsymbol{\theta}' &\iff \forall i, \mathbf{T}_i(z_i) = \mathbf{A}_i \mathbf{T}'_{\gamma(i)}(z'_{\gamma(i)}) + \mathbf{b}_i, \end{aligned} \tag{3.4}$$

where γ *is a permutation,* $\mathbf{A}_i$ *is an invertible matrix, and* $\mathbf{b}_i$ *a vector,* $\forall i \in [\![1, d]\!]$*. We say that it is weakly identifiable if*

$$\begin{aligned} p_{\boldsymbol{\theta}}(\mathbf{x}|\mathbf{u}) = p_{\boldsymbol{\theta}'}(\mathbf{x}|\mathbf{u}) &\implies \boldsymbol{\theta} \sim_A \boldsymbol{\theta}', \\ \boldsymbol{\theta} \sim_A \boldsymbol{\theta}' &\iff \mathbf{T}(\mathbf{z}) = \mathbf{A}\mathbf{T}'(\mathbf{z}') + \mathbf{b}, \end{aligned} \tag{3.5}$$

where $\mathbf{A}$ *is an invertible matrix, and* $\mathbf{b}$ *a vector.*

3.2.3 Theoretical analysis

There is a particular class of exponential families for which we can only prove a weak form of identifiability. Informally, the auxiliary variable should be able, through modulation of the prior distribution, to break the symmetries of the latent space. A similar situation arises in linear ICA with Gaussian distributions, the symmetries of which are problematic when proving identifiability.

Definition 3.3 (Quasi-location exponential distributions)**.** *We say that a univariate exponential family distribution with density* $p(y) = Q(y)e^{\mathbf{T}(y)^\top \boldsymbol{\lambda} - \Gamma(\boldsymbol{\lambda})}$ *is in the* quasi-location *family if:*

(i) $\dim(\mathbf{T}) = 1$.

(ii) $\mathbf{T}$ *is monotonic (either non-decreasing or non-increasing).*

We say that that a multivariate distribution is quasi-location exponential if all its univariate marginals are quasi-location exponential.

As a simple illustration, the Gaussian family with fixed variance is a quasi-location family, but with fixed mean it is not. This is because in the first case, the sufficient statistic is $T(y) = y$ is a monotonic scalar function, while in the second case it is $T(y) = y^2$, a non-monotonic scalar function.

3.2.3.1 Identifiability of the general case

As mentioned in Section 3.2, the IMCA model described by equations (3.1) and (3.3) generalises previous nonlinear ICA models by relaxing the independence assumption required for the latent variables. We propose here to extend the identifiability theory of nonlinear ICA developed in Hyvärinen et al. (2019) and Khemakhem et al. (2020a) to this new framework.

We start by providing a weaker form of identifiability guarantee that applies to the general case, including quasi-location families.

Theorem 3.4. *Assume the following:*

(i) The observed data follows the exponential IMCA model of equations (3.1) *and* (3.3)*.*

(ii) The mixing function $\mathbf{f} : \mathbb{R}^d \to \mathbb{R}^d$ *is invertible.*

(iii) The conditional latent distribution $p(\mathbf{z}|\mathbf{u})$ *is* strongly exponential *(Definition 2.7), and its sufficient statistic is differentiable.*

(iv) There exist $k + 1$ *distinct points* $\mathbf{u}^0, \ldots, \mathbf{u}^k$ *such that the matrix*

$$\mathbf{L} = (\boldsymbol{\lambda}(\mathbf{u}_1) - \boldsymbol{\lambda}(\mathbf{u}_0), \ldots, \boldsymbol{\lambda}(\mathbf{u}_k) - \boldsymbol{\lambda}(\mathbf{u}_0))$$

of size $k \times k$ *is invertible, where* $k = \sum_{i=1}^{d} \dim(\mathbf{T}_i)$.

Then the IMCA model is weakly identifiable.

See Appendix 3.A for the proof. This theorem extends the basic identifiability result of Theorem 2.9 in Chapter 2. It proves a general identifiability results without the restriction of having independent latent variables. This was previously not considered to be possible and could only be demonstrated in very specific circumstances and under very restrictive additional assumptions (e.g., Monti and Hyvärinen (2018) require *both* non-negativity and orthonormality of a mixing matrix in the *linear* case). In the nonlinear case, to prove Theorem 3.4, we still require that the latent variables are only dependent through the base measure, while still being independently modulated through the auxiliary variable $\mathbf{u}$. This (and the necessity of having an auxiliary variable) is the price to pay for obtaining identifiability in a nonlinear setting.

3.2.3.2 Identifiability of the non quasi-location family

The identifiability result of Theorem 3.4 is weak because of the presence of the linear transformation $\mathbf{A}$ in equation (3.5). It turns out that by excluding the quasi-location family (Definition 3.3), we can remove this matrix and achieve a stronger form of identifiability. The main technical result of this chapter is the following theorem, proved in Appendix 3.A.

Theorem 3.5. *Assume that the assumptions of Theorem 3.4 hold. Further assume one of the two following sets of assumptions:*

(v) *The sufficient statistic in (3.3) is twice differentiable and* $\dim(\mathbf{T}_l) \geq 2$, $\forall l$.

(vi) *The mixing function* $\mathbf{f}$ *is a* $\mathcal{D}^2$*-diffeomorphisms*[1].

or

(v)' $\dim(\mathbf{T}_l) = 1$ *and* $\mathbf{T}_l$ *is non-monotonic* $\forall l$.

(vi)' *The mixing function* $\mathbf{f}$ *is a* $\mathcal{C}^1$*-diffeomorphism*[2].

Then the IMCA model is strongly identifiable.

[1] invertible, all second order cross-derivatives of the function and its inverse exist but aren't necessarily continuous

[2] invertible, all partial derivatives of the function and its inverse exist and are continuous

This form of identifiability mirrors the strongest results proven for nonlinear ICA in Theorems 2.9 and 2.11 of Chapter 2, without requiring that the latent components be independent. As far as we know, this is the first proof of the kind for nonlinear representation learning. We further note that this theorem generalises even the existing identifiability theory of the linear case.

3.2.3.3 Identifiability in the total absence of independence

The identifiability results presented above by Theorems 3.4 and 3.5 are made possible by the assumption of independent modulation (3.3). It might be desirable to forgo such an assumption, and consider a general exponential family distribution for the latent variable:

$$p(\mathbf{z}|\mathbf{u}) = Q(\mathbf{z})e^{\langle \mathbf{T}(\mathbf{z}), \boldsymbol{\lambda}(\mathbf{u})\rangle - \Gamma(\boldsymbol{\lambda})}. \tag{3.6}$$

It turns out that, thanks to the modulation of the parameters $\boldsymbol{\lambda}$ by the auxiliary variable $\mathbf{u}$, the weak identifiability result of equation (3.5) still holds for a model described by equations (3.1) and (3.6), as summarized by the following result, which we prove in Appendix 3.A.

Theorem 3.6. *Assume the following:*

1. *The observed data follows the model described by equations* (3.1) *and* (3.6).

2. *The mixing function* $\mathbf{f} : \mathbb{R}^d \to \mathbb{R}^d$ *is invertible.*

3. *The conditional latent distribution* $p(\mathbf{z}|\mathbf{u})$ *is* strongly exponential *(Definition 2.7), and its sufficient statistic is differentiable.*

4. *There exist* $k+1$ *distinct points* $\mathbf{u}^0, \ldots, \mathbf{u}^k$ *such that the matrix*

$$\mathbf{L} = (\boldsymbol{\lambda}(\mathbf{u}_1) - \boldsymbol{\lambda}(\mathbf{u}_0), \ldots, \boldsymbol{\lambda}(\mathbf{u}_k) - \boldsymbol{\lambda}(\mathbf{u}_0))$$

of size $k \times k$ *is invertible, where* $k = \sum_{i=1}^{d} \dim(\mathbf{T}_i)$.

Then the model is weakly identifiable.

3.3 Estimation of IMCA

A recent development in nonlinear ICA is given by Hyvärinen et al. (2019) where the authors assume they observe data $\mathbf{x} = \mathbf{f}(\mathbf{z})$ following a noiseless conditional nonlinear ICA model $p(\mathbf{z}|\mathbf{u}) = \prod_i p_i(z_i|\mathbf{u})$. For estimation, they rely on a self-supervised binary discrimination task based on randomization to learn the unmixing function. It turns out that this approach can also be used to estimate the sources in an IMCA model.

More specifically, we suppose that we observe data $(\mathbf{x}, \mathbf{u})$ that follows the exponential IMCA model of equations (3.1) and (3.3). Following Hyvärinen et al. (2019) we start by constructing new data from the observations $\mathbf{x}$ and $\mathbf{u}$ to obtain two datasets

$$\tilde{\mathbf{x}} = (\mathbf{x}, \mathbf{u}), \tag{3.7}$$

$$\tilde{\mathbf{x}}^* = (\mathbf{x}, \mathbf{u}^*), \tag{3.8}$$

where $\mathbf{u}^*$ is a random value from the distribution of $\mathbf{u}$ and independent of $\mathbf{x}$. We then proceed by defining a binary classification task, where we consider the set of all $\{\tilde{\mathbf{x}}, \tilde{\mathbf{x}}^*\}$ as data points to be classified, and whether they come from the randomised dataset or not as labels. In particular, we train a deep neural network using binary logistic regression to perform this classification task. The last hidden layer of the neural network is a feature extractor denoted $\mathbf{s}(\mathbf{x})$. The purpose of the feature extractor is therefore to extract the relevant features which will allow to distinguish between the true dataset $\tilde{\mathbf{x}}$ and the randomised dataset $\tilde{\mathbf{x}}^*$. The final layer of the network is simply linear, and the regression function takes the form

$$r(\mathbf{x}, \mathbf{u}) = \mathbf{s}(\mathbf{x})^\top \mathbf{v}(\mathbf{u}) + a(\mathbf{x}) + b(\mathbf{u}). \tag{3.9}$$

By learning to optimize for this regression task, the feature extractor $\mathbf{s}$ will approximate the true source variables, as is summarized by the following theorem, proved in Appendix 3.B.

Theorem 3.7 (Hyvärinen et al. (2019), adapted)**.** *Assume that the assumptions of Theorems 3.4 and 3.5. Further assume that we train a nonlinear logistic regression with universal approximation capability to discriminate between $\tilde{\mathbf{x}}$ in equation (3.7) and $\tilde{\mathbf{x}}^*$ in equation (3.8) with the regression function $r(\mathbf{x}, \mathbf{u})$ in equation (3.9), where the feature extractor has dimension d.*

Then in the limit of infinite data, the components $s_i(\mathbf{x})$ of the regression function (3.9) give the latent components up to pointwise nonlinearities.

Another recent development in nonlinear ICA estimation was detailed in Chapter 2, where a different approach for the estimation of nonlinear ICA was proposed. We can recover the independent sources using an identifiable VAE (iVAE) conditional on an auxiliary variable $\mathbf{u}$. The theory of iVAE is premised on the consistency of maximum likelihood training and the flexibility of VAEs in approximating densities. It was shown that given enough data, a variational posterior $q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x},\mathbf{u})$ learns to approximate the true posterior distribution $p_{\boldsymbol{\theta}}(\mathbf{z}|\mathbf{x},\mathbf{u})$, and can thus be used to invert the mixing function. The iVAE, like a regular VAE, is trained by maximizing a lower bound (ELBO) on the data log-likelihood (Kingma and Welling, 2014). Given a dataset $\mathcal{D}$ of observations $(\mathbf{x},\mathbf{u})$, the ELBO $\mathcal{L}(\boldsymbol{\theta},\boldsymbol{\phi})$ is given by

$$\mathbb{E}_{p_\mathcal{D}}\left[\log p_{\boldsymbol{\theta}}(\mathbf{x}|\mathbf{u})\right] \geq \mathcal{L}(\boldsymbol{\theta},\boldsymbol{\phi}) = \mathbb{E}_{p_\mathcal{D}}\left[\mathbb{E}_{q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x},\mathbf{u})}\left[\log p_{\boldsymbol{\theta}}(\mathbf{x},\mathbf{z}|\mathbf{u}) - \log q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x},\mathbf{u})\right]\right]. \tag{3.10}$$

While this framework has been developed for nonlinear ICA, the proofs for the identifiability results can be generalized for IMCA. This is summarized by the following theorem, proved in Appendix 3.B.

Theorem 3.8. *Assume that the assumptions of Theorems 3.4 and 3.5. Further assume that the family of distributions $q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x},\mathbf{u})$ contains $p_{\boldsymbol{\theta}}(\mathbf{z}|\mathbf{x},\mathbf{u})$, and that we maximize $\mathcal{L}(\boldsymbol{\theta},\boldsymbol{\phi})$ in equation (3.10) with respect to both $\boldsymbol{\theta}$ and $\boldsymbol{\phi}$.*

Then in the limit of infinite data, the VAE learns the true parameters $\boldsymbol{\theta}^*$ of the model (3.1), up to the equivalence class of strong identifiability, defined in equation (3.4).

Finally, while both these nonlinear ICA methods can be adapted for the estimation of IMCA, they are not without their limitations. On the one hand, the self-supervised approaches rely on heuristic schemes and are statistically inefficient. On the other hand, the likelihood-based approach used in iVAE requires normalized densities. Normalizing constants are notoriously hard to compute in high dimensions, especially when the densities at play are multivariate and do not factor into products of univariate densities due to a lack of independence. That is why we propose a novel estimation method based on energy-based modelling in Chapter 4, which attempts to solve these

limitations. We also relegate all the experimental validation of the estimation of IMCA to the next chapter.

3.4 Conclusion

We introduce the Independently Modulated Component Analysis (IMCA) framework: a generative model that generalizes previous nonlinear ICA models (Hyvärinen et al., 2019; Khemakhem et al., 2020a) by allowing the latent variables to be dependent, while retaining the identifiability guarantees. The dependence between the latent components is achieved by using an arbitrary base measure for their joint distribution.

The key assumption for identifiability is to assume that the latent joint density has a part that is modulated in a factorizable way when conditioned by an auxiliary variable (such as time index, history, or another data source). The central contribution of this chapter is a thorough analysis of the identifiability of IMCA. Our proofs extend previous ones to the non-independent case, and are the most general to date, even when considering linear ICA theory.

Finally, we show that some of the previous nonlinear ICA estimation methods can be easily adapted for the estimation of IMCA. The empirical validation is relegated to Chapter 4 after introducing a novel estimation technique.

Appendices to Chapter 3

3.A Identifiability proofs

If a distribution is both strongly exponential and exponentially factorial, then it satisfies many of the properties discussed in Appendix 2.A for the univariate strongly exponential family, as is summarized by the following proposition.

Proposition 3.9. *If a density $p_{\boldsymbol{\theta}}$ is both strongly exponential and exponentially factorial, then the conclusions of Lemmas 2.16 to 2.20 apply to the individual sufficient statistics $\mathbf{T}_i$ in equation (3.2).*

This means that we can use Lemmas 2.16 to 2.20 in the proofs above, even if the densities in question are not univariate.

Proof of Theorem 3.4. Consider two different sets of parameters $(\mathbf{f}, \mathbf{T}, \boldsymbol{\lambda}, Q)$ and $(\tilde{\mathbf{f}}, \tilde{\mathbf{T}}, \tilde{\boldsymbol{\lambda}}, \tilde{Q})$, defining two conditional latent densities $p(\mathbf{z}|\mathbf{u})$ and $\tilde{p}(\mathbf{z}|\mathbf{u})$. Let $\mathbf{g} \coloneqq \mathbf{f}^{-1}$ and $\tilde{\mathbf{g}} \coloneqq \tilde{\mathbf{f}}^{-1}$ Suppose that the density of the observations arising from these two different models are equal:

$$p(\mathbf{x}|\mathbf{u}) = \tilde{p}(\mathbf{x}|\mathbf{u}), \tag{3.11}$$

$$\log p(\mathbf{g}(\mathbf{x})|\mathbf{u}) - \log|\det \mathbf{J}_{\mathbf{g}}(\mathbf{x})| = \log p(\tilde{\mathbf{g}}(\mathbf{x})|\mathbf{u}) - \log|\det \mathbf{J}_{\tilde{\mathbf{g}}}(\mathbf{x})|\,, \tag{3.12}$$

$$\begin{aligned}\log Q(\mathbf{g}(\mathbf{x})) + \mathbf{T}(\mathbf{g}(\mathbf{z}))^\top \boldsymbol{\lambda}(\mathbf{u}) - \Gamma(\mathbf{u}) - \log|\det \mathbf{J}_{\mathbf{g}}(\mathbf{x})| = \\ \log \tilde{Q}(\tilde{\mathbf{g}}(\mathbf{x})) + \tilde{\mathbf{T}}(\tilde{\mathbf{g}}(\mathbf{z}))^\top \tilde{\boldsymbol{\lambda}}(\mathbf{u}) - \tilde{\Gamma}(\mathbf{u}) - \log|\det \mathbf{J}_{\tilde{\mathbf{g}}}(\mathbf{x})|\,.\end{aligned} \tag{3.13}$$

Let $\mathbf{u}_0, \ldots, \mathbf{u}_k$ be the points provided by assumption (iii) of the theorem for $\mathbf{T}$, where $k = \sum_i k_i$, and $k_i = \dim(\mathbf{T}_i)$. We plug each of those $\mathbf{u}_l$ in equation (3.13) to obtain $k+1$ such equations. Then, we subtract the first equation for $\mathbf{u}_0$ from the remaining k equations to get for $l = 1, \ldots, k$:

$$\mathbf{T}(\mathbf{z})^\top(\boldsymbol{\lambda}(\mathbf{u}_l) - \boldsymbol{\lambda}(\mathbf{u}_0)) - G(\mathbf{u}_l) = \tilde{\mathbf{T}}(\mathbf{z})^\top(\tilde{\boldsymbol{\lambda}}(\mathbf{u}_l) - \tilde{\boldsymbol{\lambda}}(\mathbf{u}_0)) - \tilde{G}(\mathbf{u}_l), \tag{3.14}$$

where we grouped terms that are only a function of $\mathbf{u}_l$ in G and $\tilde{G}$.

The **crucial point** here is that the non factorial terms $Q(\mathbf{g}(\mathbf{x}))$ and $\tilde{Q}(\tilde{\mathbf{g}}(\mathbf{x}))$ cancel out when we take these differences. This is what allows us to generalize the identifiability results of nonlinear ICA to IMCA.

Let $\mathbf{L}$ bet the matrix defined in assumption (iv), and $\tilde{\mathbf{L}} \coloneqq (\dots, \tilde{\boldsymbol{\lambda}}(\mathbf{u}_l) - \tilde{\boldsymbol{\lambda}}(\mathbf{u}_0), \dots)$. Define $\mathbf{b} = (\dots, G(\mathbf{u}_l) - \tilde{G}(\mathbf{u}_l), \dots)$. Expressing equation (3.14) for all points $\mathbf{u}_l$ in matrix form, we get:

$$\tilde{\mathbf{L}}^\top \tilde{\mathbf{T}}(\mathbf{z}) = \mathbf{L}^\top \mathbf{T}(\mathbf{z}) - \mathbf{b} \tag{3.15}$$

By assumption (iv), $\mathbf{L}$ is invertible, and thus we can write

$$\mathbf{T}(\mathbf{z}) = \mathbf{A}\tilde{\mathbf{T}}(\mathbf{z}) + \mathbf{c}, \tag{3.16}$$

where $\mathbf{c} = -\mathbf{L}^{-\top}\mathbf{b}$ and $\mathbf{A} = \mathbf{L}^{-\top}\tilde{\mathbf{L}}^\top$.

To prove that $\mathbf{A}$ is invertible, we first take the gradient of equation (3.16) with respect to $\mathbf{z}$. The Jacobian $\mathbf{J}_\mathbf{T}$ of $\mathbf{T}$ is a matrix of size $k \times d$. Its columns are independent because each $\mathbf{T}_i$ is only a function of z_i, and thus the nonzero entries of each column are in different rows. This means that its rank is d (since $k = \sum_{i=1}^d k_i \geq d$). This is not enough to prove that $\mathbf{A}$ is invertible though. For that, we consider the functions $\mathbf{T}_i$ for which $k_i > 1$: for each of these functions, using Lemma 2.17, there exists points $z_i^{(1)}, \dots, z_i^{(k_i)}$ such that $(\mathbf{T}_i'(z_i^{(1)}), \dots, \mathbf{T}_i'(z_i^{(k_i)}))$ are independent. Collate these points into $k_{\max} \coloneqq \max_i k_i$ vectors $\mathbf{z}^{(j)} \coloneqq (z_1^{(j)}, \dots z_d^{(j)})$, where for each i, $z_i^{(j)} = z_i^{(1)}$ if $j > k_i$, and $z_i^{(1)}$ is a point such that $T_i(z_i^{(1)}) \neq 0$ if $k_i = 1$. We plug these vectors into equation (3.16) after differentiating it, and collate the $dk_{\max}$ equations in vector form:

$$\mathbf{M} = \mathbf{A}\tilde{\mathbf{M}}, \tag{3.17}$$

where $\mathbf{M} \coloneqq \left(\dots, \mathbf{J}_\mathbf{T}(\mathbf{z}^{(j)}), \dots\right)$ and $\tilde{\mathbf{M}} \coloneqq \left(\dots, \mathbf{J}_{\tilde{\mathbf{T}}}(\mathbf{z}^{(j)}), \dots\right)$. Now the matrix $\mathbf{M}$ is of size $k \times dk_{\max}$, and it has exactly k independent columns by definition of the points $\mathbf{z}^{(j)}$. This means that $\mathbf{M}$ is of rank k, which in turn implies that $\operatorname{rank}(\mathbf{A}) \geq k$. Since $\mathbf{A}$ is a $k \times k$ matrix, we conclude that $\mathbf{A}$ is invertible. □

Proof of Theorem 3.5. The conclusion of Theorem 3.4 mirrors the one developed by Khemakhem et al. (2020a) for nonlinear ICA. In fact, the idea behind the IMCA framework was to show that the factorial modulation of the latent variable by the auxiliary variable is a strong assumption, which allows us to relax independence without sacrificing any identifiability guarantees. To prove the stronger form of identifiability of equation (3.4), we can simply make the same assumptions as Khemakhem et al. (2020a, Theorems 2, 3), and refer to their proof. □

Proof of Theorem 3.6. We start by considering two different sets of parameters $\boldsymbol{\theta} = (\mathbf{f}, \mathbf{T}, \boldsymbol{\lambda}, Q)$ and $\tilde{\boldsymbol{\theta}} = (\tilde{\mathbf{f}}, \tilde{\mathbf{T}}, \tilde{\boldsymbol{\lambda}}, \tilde{Q})$, defining two conditional latent densities $p(\mathbf{z}|\mathbf{u})$ and $\tilde{p}(\mathbf{z}|\mathbf{u})$. Note that in the proof of Theorem 3.4, equations (3.11) to (3.16) do not make use of the fact that the prior distribution is exponentially factorial. In fact the latter is only invoked to justify the use of Lemma 2.17 to prove that the matrix $\mathbf{A}$ in equation (3.16) is invertible.

Here, we will use a slightly different approach. We start from

$$\mathbf{T}(\mathbf{z}) = \mathbf{A}\tilde{\mathbf{T}}(\mathbf{z}) + \mathbf{c}. \tag{3.18}$$

Let $\mathbf{z}^{(0)} \in \mathbb{R}^d$, and define $\mathbf{g}(\mathbf{z}) = \mathbf{T}(\mathbf{z}) - \mathbf{T}(\mathbf{z}^{(0)})$ and similarly for $\tilde{\mathbf{g}}(\mathbf{z})$. By Lemma 2.18, since $p(\mathbf{z}|\mathbf{u})$ is strongly exponential, there exists k points $\mathbf{z}^{(1)}, \ldots, \mathbf{z}^{(k)}$ such that $\mathbf{R} = \left(\mathbf{T}(\mathbf{z}^{(1)}) - \mathbf{T}(\mathbf{z}^{(0)}), \ldots, \mathbf{T}(\mathbf{z}^{(k)}) - \mathbf{T}(\mathbf{z}^{(0)})\right) \in \mathbb{R}^{k \times k}$ is invertible. By defining $\tilde{\mathbf{R}}$ similarly to $\mathbf{R}$ and plugging all this in equation (3.18), we get

$$\mathbf{R} = \mathbf{A}\tilde{\mathbf{R}}. \tag{3.19}$$

Since $\mathbf{R}$ is invertible, we can conclude that $\mathbf{A}$ is also invertible, and that the IMCA model is weakly identifiable. □

3.B Estimation proofs

Proof of Theorem 3.7. The proof of this theorem is inspired from Hyvärinen et al. (2019). By well known theory, after convergence of logistic regression, the regression function equals the difference of the log-densities of the two classes:

$$\sum_{i=1}^{d} s_i(\mathbf{x}) v_i(\mathbf{u}) + a(\mathbf{x}) + b(\mathbf{u}) = \log p_{\tilde{\mathbf{x}}}(\mathbf{x}, \mathbf{u}) - \log p_{\tilde{\mathbf{x}}^*}(\mathbf{x}, \mathbf{u}^*) \tag{3.20}$$

$$\begin{aligned} &= \log p(\mathbf{z}, \mathbf{u}) + \log |\mathbf{J}_{\mathbf{g}}(\mathbf{x})| \\ &\quad - \log p(\mathbf{z}) p(\mathbf{u}) - \log |\mathbf{J}_{\mathbf{g}}(\mathbf{x})| \end{aligned} \tag{3.21}$$

$$= \log p(\mathbf{z}|\mathbf{u}) - \log p(\mathbf{z}) \tag{3.22}$$

$$\begin{aligned} &= \log Q(\mathbf{z}) - \log Z(\mathbf{u}) + \sum_{i=1}^{d} \mathbf{T}_i(z_i)^\top \boldsymbol{\lambda}_i(\mathbf{u}) \\ &\quad - \log p(\mathbf{z}), \end{aligned} \tag{3.23}$$

where $\mathbf{g} = \tilde{\mathbf{f}}^{-1}$ and $\mathbf{J}_{\mathbf{g}}(\mathbf{x})$ is the Jacobian matrix of $\mathbf{g}$ at point $\mathbf{x}$. Let $\mathbf{u}_0, \ldots, \mathbf{u}_k$ be the point provided by assumption (iv). We plug each of those $\mathbf{u}_k$ in

equation (3.20) to obtain $k+1$ such equations. We subtract the first equation for $\mathbf{u}_0$ from the remaining k equations to get for $l = 1, \ldots, k$:

$$\sum_{i=1}^{d} s_i(\mathbf{x})(v_i(\mathbf{u}_l) - v_i(\mathbf{u}_0)) + (b(\mathbf{u}_l) - b(\mathbf{u}_0)) - \log \frac{Z(\mathbf{u}_l)}{Z(\mathbf{u}_0)} = \sum_{i=1}^{d} \mathbf{T}_i(z_i)^\top (\boldsymbol{\lambda}_i(\mathbf{u}_l) - \boldsymbol{\lambda}_i(\mathbf{u}_0)). \quad (3.24)$$

Interestingly, the term $\log Q(\mathbf{z})$ cancels out. The rest of the proof is similar to Theorems 3.4 and 3.5. By defining and $\mathbf{L} = (\ldots, \boldsymbol{\lambda}(\mathbf{u}_l) - \boldsymbol{\lambda}(\mathbf{u}_0), \ldots) \in \mathbb{R}^{k \times k}$ and $\mathbf{V} = (\ldots, \mathbf{v}(\mathbf{u}_l) - \mathbf{v}(\mathbf{u}_0), \ldots) \in \mathbb{R}^{k \times d}$, and grouping all other terms in $\mathbf{b}$, we get

$$\mathbf{L}^\top \mathbf{T}(\mathbf{z}) = \mathbf{V}^\top \mathbf{s}(\mathbf{x}) + \mathbf{b}. \quad (3.25)$$

Since $\mathbf{L}$ is invertible, this equation is equivalent to

$$\mathbf{T}(\mathbf{z}) = \mathbf{A}\mathbf{s}(\mathbf{x}) + \mathbf{c}, \quad (3.26)$$

where $\mathbf{A} = \mathbf{L}^{-\top}\mathbf{V}^\top \in \mathbb{R}^{k \times d}$ and $\mathbf{c} = \mathbf{L}^{-\top}\mathbf{b}$. This looks very similar to equation (3.16), but this time, the matrix $\mathbf{A}$ is not square, albeit full column rank. This makes it harder to mimic the proof of Theorem 3.5 to prove that the self-supervised learning scheme can be used to learn the true latent components up to permutation and nonlinear scaling (strong identifiability).

We can alleviate this by slightly altering the form of the regression function (3.9). We need to make the effective dimension of the feature extractor $\mathbf{s}$ match $k = \sum_i k_i$, the total dimension of the sufficient statistic $\mathbf{T}$.

Let $\mathbf{H}_{k_i}$ the function that repeats its scalar input k_i times:

$$\mathbf{H}_{k_i}(x) = (x, \ldots, x) \in \mathbb{R}^{k_i} \quad (3.27)$$

By using the alternative regression function

$$\tilde{r}(\mathbf{x}, \mathbf{u}) = \sum_{i=1}^{d} \mathbf{H}_{k_i}(s_i(\mathbf{x}))^\top \mathbf{v}_i(\mathbf{u}) + a(\mathbf{x}) + b(\mathbf{u}), \quad (3.28)$$

where $\mathbf{v}_i \in \mathbb{R}^{k_i}$, and similar arguments as above, we get

$$\sum_{i=1}^{d} \mathbf{H}_{k_i}(s_i(\mathbf{x}))^\top (\mathbf{v}_i(\mathbf{u}_l) - \mathbf{v}_i(\mathbf{u}_0)) + (b(\mathbf{u}_l) - b(\mathbf{u}_0)) - \log \frac{Z(\mathbf{u}_l)}{Z(\mathbf{u}_0)} = \sum_{i=1}^{d} \mathbf{T}_i(z_i)^\top (\boldsymbol{\lambda}_i(\mathbf{u}_l) - \boldsymbol{\lambda}_i(\mathbf{u}_0)). \quad (3.29)$$

We then proceed as above, with the difference that the matrix $\mathbf{A} \in \mathbb{R}^{k \times k}$ is now square.

By simply substituting $\tilde{\mathbf{T}}$ by $\mathbf{s}$ in the proof of Theorems 3.4 and 3.5, we can use the same reasoning to conclude. □

Proof of Theorem 3.8. The identifiability results of Theorems 3.4 and 3.5 match those developed for nonlinear ICA models by Khemakhem et al. (2020a). Their consistency proof builds on the identifiability up to equivalence class of the model and can thus be applied to IMCA without modification. We give a summary of the proof below.

The ELBO (3.10) can be written as

$$\mathcal{L}(\boldsymbol{\theta}, \boldsymbol{\phi}) = \log p_{\boldsymbol{\theta}}(\mathbf{x}|\mathbf{u}) - \mathrm{KL}(q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x}, \mathbf{u}) \| p_{\boldsymbol{\theta}}(\mathbf{z}|\mathbf{x}, \mathbf{u})). \tag{3.30}$$

If the family $q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x}, \mathbf{u})$ is large enough to include $p_{\boldsymbol{\theta}}(\mathbf{z}|\mathbf{x}, \mathbf{u})$, then by optimizing the loss over its parameter $\boldsymbol{\phi}$, we will minimize the KL term, eventually reaching zero, and the loss will be equal to the log-likelihood. The VAE in this case inherits all the properties of maximum likelihood estimation. In particular, MLE is consistent. This consistency entails that MLE will converge to the true parameter $\boldsymbol{\theta}^*$ if a model is identifiable. In our case, since we only have identifiability up to an equivalence class, the consistency of MLE can only guarantee that we learn the true parameter up to this equivalence class, which is what we seek to achieve. □

Chapter 4

Identifiable conditional energy-based models

We consider the identifiability theory of probabilistic models and establish sufficient conditions under which the representations learned by a very broad family of conditional energy-based models are unique in function space, up to a simple transformation. In our model family, the energy function is the dot-product between two feature extractors, one for the dependent variable, and one for the conditioning variable. We show that under mild conditions, the features are unique up to scaling and permutation. Our results extend recent developments in nonlinear ICA, and in fact, they lead to an important generalization of ICA models. In particular, we show that our model can be used to estimate the components in the framework of Independently Modulated Component Analysis (IMCA), a new generalization of nonlinear ICA that relaxes the independence assumption. A thorough empirical study shows that representations learned by our model from real-world image datasets are identifiable and improve performance in transfer learning and semi-supervised learning tasks.

This chapter is based on Khemakhem et al. (2020b).

4.1 Introduction

A central question in unsupervised deep learning is how to learn nonlinear representations that are a faithful reconstruction of the true latent variables behind the data. This allows us to learn representations that are semantically meaningful, interpretable and useful for downstream tasks. Identifiability is fundamental for meaningful and principled disentanglement and in applications such as causal discovery. However, this is a challenging task: by definition, we never observe the latent variables; the only information directly available to us is given by the observed variables. Learning the true representations is only possible when the representation is identifiable: if, in the limit of infinite data, only a single representation function can fit the data. Conversely, if multiple representation functions can fit the observations in the limit of infinite data, then the true representation function is unidentifiable.

Until recently (Hyvärinen and Morioka, 2016; Hyvärinen and Morioka, 2017), results relating to identifiability of (explicit and implicit) latent variable models were mainly constrained to linear models (as in linear ICA for example), as it was acknowledged that the flexibility of nonlinear mappings could yield arbitrary latent variables which fulfil model assumptions such as independence (Hyvärinen and Pajunen, 1999). However, it is now understood that nonlinear deep latent variable models can be identifiable provided we observe some additional auxiliary variables such that the latent variables are conditionally independent given the auxiliary variable. The approach was introduced using self-supervised learning by Hyvärinen et al. (2019), and Khemakhem et al. (2020a) explicited a connection between nonlinear ICA and the framework of variational autoencoders. It was shortly followed by work by Sorrenson et al. (2020), where a similar connection was made to flow-based models (Rezende and Mohamed, 2015). This signals the importance of identifiability in popular deep generative models. These works formalized a trade-off between distributional assumptions over latent variables (from linear and independent to nonlinear but conditionally independent given auxiliary variables) that would lead to identifiability.

We extend this trend to a broad family of (unnormalized) conditional energy-based models (EBM), using insight from the nonlinear ICA theory. EBMs offer unparalleled flexibility, mainly because they do not require the modelled

densities to be normalized nor easy to sample from. The energy function we will consider is defined in two steps: we learn two feature extractors, parametrized by neural networks, one for each of the observed variables (dependent and conditioning); then, we set the energy function to be the dot-product of the learned features. The modelled conditional densities are defined to be the exponential of the negative energy function.

The theoretical contribution of this chapter is to provide a set of sufficient mild conditions to be satisfied by the feature extractors, which would guarantee their identifiability: they learn representations that are unique up to a linear transformation. In addition, by slightly altering the definition of the energy function, we prove that the linear transformation is essentially a permutation. These conditions are functional, *i.e.* they abstract away the architecture of the networks. Moreover, our conditional EBM generalizes previous results by altogether dropping any distributional assumptions on the representations—which are ubiquitous in the latent variable case. Finally, most of our theoretical results hold for overcomplete representations, which means that unlike the earlier works cited above, our model can even be shown to have universal approximation capabilities. Effectively, this makes our family of models very flexible and adaptable to practical problems. We call this model Identifiable Conditional Energy-Based deep Models, or ICE-BeeM for short.

Besides, while recent identifiability theory focused on providing functional conditions for identifiability, such work is a bit removed from the reality of neural network training. Here, we provide a neural network architecture based on fully connected layers, for which the functional conditions hold and is thus identifiable. This is the first step to bridge the gap between theoretically identifiable models and ones that can be implemented in practice.

As an application of our identifiability results, we show that, after fitting the conditional model to the data, a careful design of ICE-BeeM allows the feature extractor to uniquely recover the latent variables that generated the observations according to a nonlinear ICA, or, more generally, an IMCA model. This solves many of the limitations of previously proposed estimation methods (Hyvärinen et al., 2019; Khemakhem et al., 2020a). As a further, somewhat different application of our results, we show how identifiability of ICE-BeeM can be leveraged for transfer and semi-supervised learning. Finally, we show empirically that ICE-BeeM learns identifiable representations from real-world

image datasets. In fact, we believe that the identifiability results are generally important for a principled application of EBMs, whether for the purposes of disentanglement or otherwise.

4.2 Identifiable conditional energy-based deep models

In this section, we define ICE-BeeM, and study its properties. All proofs can be found in Appendix 4.C.

4.2.1 Model definition

We collect a dataset of observations consisting of tuples $(\mathbf{x}, \mathbf{y})$, where $\mathbf{x} \in \mathcal{X} \subset \mathbb{R}^{d_x}$ is the main variable of interest, also called the dependent variable, and $\mathbf{y} \in \mathcal{Y} \subset \mathbb{R}^{d_y}$ is an auxiliary variable also called the conditioning variable.

Consider two feature extractors $\mathbf{f}_{\boldsymbol{\theta}}(\mathbf{x}) \in \mathbb{R}^{d_z}$ and $\mathbf{g}_{\boldsymbol{\theta}}(\mathbf{y}) \in \mathbb{R}^{d_z}$, which we parametrize by neural networks, and where $\boldsymbol{\theta}$ is the vector of weights and biases. To alleviate notations, we will drop $\boldsymbol{\theta}$ when it's clear which quantities we refer to. These feature extractors are used to define the conditional energy function

$$\mathcal{E}_{\boldsymbol{\theta}}(\mathbf{x}|\mathbf{y}) = \mathbf{f}_{\boldsymbol{\theta}}(\mathbf{x})^\top \mathbf{g}_{\boldsymbol{\theta}}(\mathbf{y}). \tag{4.1}$$

The parameter $\boldsymbol{\theta}$ lives in the space Θ, which is defined such that the normalizing constant $Z(\mathbf{y}, \boldsymbol{\theta})$ is finite:

$$\Theta = \left\{ \boldsymbol{\theta} : Z(\mathbf{y}; \boldsymbol{\theta}) = \int_{\mathcal{X}} \exp(-\mathcal{E}_{\boldsymbol{\theta}}(\mathbf{x}|\mathbf{y}))\mathrm{d}\mathbf{x} < \infty \right\}$$

Our family of conditional energy-based models has the form:

$$p_{\boldsymbol{\theta}}(\mathbf{x}|\mathbf{y}) = \frac{\exp(-\mathbf{f}_{\boldsymbol{\theta}}(\mathbf{x})^\top \mathbf{g}_{\boldsymbol{\theta}}(\mathbf{y}))}{Z(\mathbf{y}; \boldsymbol{\theta})}. \tag{4.2}$$

As we will see later, this choice of the energy function is not restrictive, as our model has powerful theoretical guarantees: universal approximation capabilities and strong identifiability properties. There exists a multitude of methods we can use to estimate this model (Hyvärinen, 2005; Gutmann and Hyvärinen, 2010; Ceylan and Gutmann, 2018; Uehara et al., 2020). In this chapter, we

will use Flow Contrastive Estimation (Gao et al., 2020) and Denoising Score Matching (Vincent, 2011), which are discussed and extended to the conditional case in Appendix 4.B.

4.2.2 Identifiability

As stated earlier, we want our model to learn meaningful representations of the dependent and conditioning variables. In particular, when learning two different models of the family (4.2) from the same dataset, we want the learned features to be very similar.

This similarity between representations is better expressed as an equivalence relation on the parameters $\boldsymbol{\theta}$ of the network, which would characterize the form of identifiability we will end up with for our energy model. This notion of identifiability up to equivalence class was introduced in Section 2.4.1 to address the fact that there typically exist many choices of neural network parameters $\boldsymbol{\theta}$ that map to the same point in function-space. In our case, it is given by the following definitions:

Definition 4.1 (Weak identifiability)**.** *Let $\sim_w^{\mathbf{f}}$ and $\sim_w^{\mathbf{g}}$ be equivalence relations on Θ defined as:*

$$\begin{aligned} \boldsymbol{\theta} \sim_w^{\mathbf{f}} \boldsymbol{\theta}' &\Leftrightarrow \forall \mathbf{x}, \mathbf{f}_{\boldsymbol{\theta}}(\mathbf{x}) = \mathbf{A}\mathbf{f}_{\boldsymbol{\theta}'}(\mathbf{x}) + \mathbf{c}, \\ \boldsymbol{\theta} \sim_w^{\mathbf{g}} \boldsymbol{\theta}' &\Leftrightarrow \forall \mathbf{x}, \mathbf{g}_{\boldsymbol{\theta}}(\mathbf{y}) = \mathbf{B}\mathbf{g}_{\boldsymbol{\theta}'}(\mathbf{y}) + \mathbf{e}, \end{aligned} \tag{4.3}$$

where $\mathbf{A}$ and $\mathbf{B}$ are $(d_z \times d_z)$-matrices, whose ranks are at least $\min(d_z, d_x)$ and $\min(d_z, d_y)$ respectively, and $\mathbf{c}$ and $\mathbf{e}$ are vectors.

Definition 4.2 (Strong identifiability)**.** *Let $\sim_s^{\mathbf{f}}$ and $\sim_s^{\mathbf{g}}$ be the equivalence relations on Θ defined as:*

$$\begin{aligned} \boldsymbol{\theta} \sim_s^{\mathbf{f}} \boldsymbol{\theta}' &\Leftrightarrow \forall i, \forall \mathbf{x}, f_{i,\boldsymbol{\theta}}(\mathbf{x}) = a_i f_{\sigma(i),\boldsymbol{\theta}'}(\mathbf{x}) + c_i, \\ \boldsymbol{\theta} \sim_s^{\mathbf{g}} \boldsymbol{\theta}' &\Leftrightarrow \forall i, \forall \mathbf{x}, g_{i,\boldsymbol{\theta}}(\mathbf{x}) = b_i g_{\gamma(i),\boldsymbol{\theta}'}(\mathbf{x}) + e_i, \end{aligned} \tag{4.4}$$

where σ and γ are permutations of $[\![1, n]\!]$, a_i and b_i are nonzero scalars and c_i and e_i are scalars.

Two parameters are thus considered equivalent if they parametrize two feature extractors that are equal up to a linear transformation (4.3) or a

scaled permutation (4.4). The subscripts w and s stand for weak and strong, respectively.

Remark 4.3. The relation $\sim_w^{\mathbf{f}}$ in equation (4.3) is an equivalence relation in the strict term only if $\mathbf{A}$ is full rank. If $\mathbf{A}$ is not full rank (which is only possible if $d_z > d_x$, given the rest of assumptions), then the relation is not necessarily symmetric. This is not a real problem, and can be fixed by changing the definition to: "there exists $\mathbf{A}_1, \mathbf{A}_2$ such that $\mathbf{f}_{\boldsymbol{\theta}} = \mathbf{A}_1 \mathbf{f}_{\boldsymbol{\theta}'} + \mathbf{c}_1$ and $\mathbf{f}_{\boldsymbol{\theta}'} = \mathbf{A}_2 \mathbf{f}_{\boldsymbol{\theta}} + \mathbf{c}_2$". In the remainder, we use the simpler version of Definition 4.1 for clarity.

Identifiability in the context of probability densities modelled by neural networks can be seen as a study of the degeneracy of the networks. In applications where the representations are used in a downstream classification task, the weak identifiability (4.3) may be enough. It guarantees that the hyperplanes defining the boundaries between classes in the feature space are consistent, up to a global rotation, and thus the downstream task may be unaffected. Strong identifiability (4.4), on the other hand, is crucial in applications where such rotation is undesirable. For example, Monti et al. (2019) propose an algorithm for causal discovery based on independence tests between the observations and latent variables learnt by solving a nonlinear ICA task. The tested independences only hold for the true latent noise variables. If one were to learn the latent variables only up to a rotation, such a causal analysis method would not work.

4.2.2.1 Weak identifiability

This initial form of identifiability requires very few assumptions on the feature extractors $\mathbf{f}$ and $\mathbf{g}$. In fact, the conditions we develop here are easy to satisfy in practice, and we will see how in Section 4.3. Most importantly, our result also covers the case where the number of features is larger than the number of observed variables. As far as we know, this is the first identifiability result that extends to *overcomplete* representations in the *nonlinear* setting. The following theorem summarizes the main result.

Theorem 4.4. *Let $\sim_w^{\mathbf{f}}$ and $\sim_w^{\mathbf{g}}$ be the equivalence relations in equation (4.3). Assume that for any choice of parameter $\boldsymbol{\theta}$:*

1. *The feature extractor* $\mathbf{f}_{\boldsymbol{\theta}}$ *is differentiable, and its Jacobian* $\mathbf{J}_{\mathbf{f}_{\boldsymbol{\theta}}}$ *is full rank.*

2. *There exist* $d_z + 1$ *points* $\mathbf{y}^0, \ldots, \mathbf{y}^{d_z}$ *such that the matrix*

$$\mathbf{R}_{\boldsymbol{\theta}} = \Big(\mathbf{g}_{\boldsymbol{\theta}}(\mathbf{y}^1) - \mathbf{g}_{\boldsymbol{\theta}}(\mathbf{y}^0), \ldots, \mathbf{g}_{\boldsymbol{\theta}}(\mathbf{y}^{d_z}) - \mathbf{g}_{\boldsymbol{\theta}}(\mathbf{y}^0)\Big)$$

of size $d_z \times d_z$ *is invertible.*

then $p_{\boldsymbol{\theta}}(\mathbf{x}|\mathbf{y}) = p_{\boldsymbol{\theta}'}(\mathbf{x}|\mathbf{y}) \implies \boldsymbol{\theta} \sim_w^{\mathbf{f}} \boldsymbol{\theta}'$, *where* $\sim_w^{\mathbf{f}}$ *is defined in equation* (4.3).

With $\mathbf{f}_{\boldsymbol{\theta}}$ *and* $\mathbf{g}_{\boldsymbol{\theta}}$ *switched, the same conclusion applies to* $\mathbf{g}_{\boldsymbol{\theta}}$*:* $p_{\boldsymbol{\theta}}(\mathbf{x}|\mathbf{y}) = p_{\boldsymbol{\theta}'}(\mathbf{x}|\mathbf{y}) \implies \boldsymbol{\theta} \sim_w^{\mathbf{g}} \boldsymbol{\theta}'$, *where* $\sim_w^{\mathbf{g}}$ *is defined in equation* (4.3).

Finally, if both assumptions 1 and 2 are satisfied by both feature extractors $\mathbf{f}_{\boldsymbol{\theta}}$ *and* $\mathbf{g}_{\boldsymbol{\theta}}$*, then the matrices* $\mathbf{A}$ *and* $\mathbf{B}$ *in equation* (4.3) *have full row rank equal to* d_z.

Remark 4.5 (Intuition behind assumption 2)**.** Assumption 2 requires that the conditioning feature extractor $\mathbf{g}$ has an image that is rich enough. Intuitively, this relaxes the amount of flexibility the main feature extractor $\mathbf{f}$ would need to have if $\mathbf{g}$ were to be very simple. It implies that the search for $\mathbf{f}$ will be naturally restricted to a smaller space, for which we can prove identifiability.

Remark 4.6 (Proof under weaker assumptions)**.** Assumption 1 of full rank Jacobian can be weakened without changing the conclusion of Theorem 4.4, by instead supposing that:

1.' There exists a point $\mathbf{x}^0 \in \mathbb{R}^d$ where the Jacobian $\mathbf{J}_{\mathbf{f}_{\boldsymbol{\theta}}}$ of $\mathbf{f}_{\boldsymbol{\theta}}$ exists and is invertible.

In addition, this condition can be scrapped altogether if we relax the definition of the equivalence class in remark 4.3 to have no conditions on the ranks of matrices $\mathbf{A}_1$ and $\mathbf{A}_2$. This, however, comes at the expense of a relatively weak and potentially meaningless equivalence class.

Finally, assumption 2 of Theorem 4.4 can be replaced by requiring the Jacobian of $\mathbf{g}_\theta$ to be differentiable and full rank in at least one point, but this is only possible if the conditioning variable is continuous.

4.2.2.2 Strong identifiability

We propose two different alterations to our energy function which will both allow for the stronger form of identifiability defined by $\sim_s^{\mathbf{f}}$ and $\sim_s^{\mathbf{g}}$ in equation (4.4).

We will focus on $\mathbf{f}$, but the same results hold for $\mathbf{g}$ by a simple transposition of assumptions. Importantly, we will suppose that the output dimension d_z is smaller than the input dimension d_x.

The first approach is based on restricting the feature extractor $\mathbf{f}$ to be non-negative. This will put constraints on the matrix $\mathbf{A}$ defining the equivalence relation $\sim_w^{\mathbf{f}}$: loosely speaking, if $\mathbf{A}$ induces a rotation in space, then it will violate the non-negativity constraint since the only rotation that maps the positive orthant of the plan to itself is the identity. As a result, the matrix $\mathbf{A}$ can only be a scaled permutation matrix, as is summarized by the following theorem.

Theorem 4.7. *Assume that $d_z \leq d_x$ and that the assumptions of Theorem 4.4 hold. Further assume that, for any choice of parameter $\boldsymbol{\theta}$:*

3. *The feature extractor $\mathbf{f}_{\boldsymbol{\theta}}$ is surjective, and its image is $\mathbb{R}_+^{d_z}$.*

Then $p_{\boldsymbol{\theta}}(\mathbf{x}|\mathbf{y}) = p_{\boldsymbol{\theta}'}(\mathbf{x}|\mathbf{y}) \implies \boldsymbol{\theta} \sim_s^{\mathbf{f}} \boldsymbol{\theta}'$, where $\sim_s^{\mathbf{f}}$ is defined in equation (4.4).

The second alteration is based on augmenting $\mathbf{f}$ by its square, computed element-wise. This effectively results in the $2d_z$-dimensional feature extractor

$$\tilde{\mathbf{f}}(\mathbf{x}) = (\dots, f_i(\mathbf{x}), f_i^2(\mathbf{x}), \dots) \in \mathbb{R}^{2d_z}. \tag{4.5}$$

This augmented feature map is combined with a $2d_z$-dimensional feature map $\tilde{\mathbf{g}}(\mathbf{y}) \in \mathbb{R}^{2d_z}$ for the conditioning variable $\mathbf{y}$, to define an augmented energy function

$$\tilde{\mathcal{E}}(\mathbf{x}|\mathbf{y}) = \tilde{\mathbf{f}}(\mathbf{x})^\top \tilde{\mathbf{g}}(\mathbf{y}). \tag{4.6}$$

The augmented feature extractor contains positive entries (the squares). By applying Theorem 4.4 to $\tilde{\mathbf{f}}$, these positive entries will put constraints on the form of the matrix $\mathbf{A}$ defining the equivalence relation $\sim_w^{\mathbf{f}}$. This will ultimately lead to stronger identifiability results for the original feature extractor $\mathbf{f}$. The advantage of this modelling trick is that it does not require the features to be positive. In fact, the squared entries only serve the purpose of constraining $\mathbf{A}$. However, it makes the effective size of the feature extractor equal to $2d_z$. This is summarized by the following identifiability result:

Theorem 4.8. *Assume that $d_z \leq d_x$ and that the assumptions of Theorem 4.4 hold. Further assume that we use the augmented energy function $\tilde{\mathcal{E}}(\mathbf{x}|\mathbf{y})$ in equation (4.6), and that, for any choice of parameter $\boldsymbol{\theta}$, the following holds:*

4. *The feature extractor $\mathbf{f}_{\boldsymbol{\theta}}$ is differentiable and surjective and its Jacobian $\mathbf{J}_{\mathbf{f}_{\boldsymbol{\theta}}}$ is full rank.*

5. *There exist $2d_z + 1$ points $\mathbf{y}^0, \ldots, \mathbf{y}^{2d_z}$ such that the matrix*

$$\tilde{\mathbf{R}}_{\boldsymbol{\theta}} = \left(\tilde{\mathbf{g}}_{\boldsymbol{\theta}}(\mathbf{y}^1) - \tilde{\mathbf{g}}_{\boldsymbol{\theta}}(\mathbf{y}^0), \ldots, \tilde{\mathbf{g}}_{\boldsymbol{\theta}}(\mathbf{y}^{2d_z}) - \tilde{\mathbf{g}}_{\boldsymbol{\theta}}(\mathbf{y}^0)\right)$$

   of size $2d_z \times 2d_z$ is invertible.

Then $p_{\boldsymbol{\theta}}(\mathbf{x}|\mathbf{y}) = p_{\boldsymbol{\theta}'}(\mathbf{x}|\mathbf{y}) \implies \boldsymbol{\theta} \sim_s^{\mathbf{f}} \boldsymbol{\theta}'$, where $\sim_s^{\mathbf{f}}$ is defined in equation (4.4).

These two theorems are important as they prove very strong identifiability results for a conditional energy-based model. As far as we know, our results require the least amount of assumptions in recent theoretical work for functional identifiability of deep learning models (Khemakhem et al., 2020a; Sorrenson et al., 2020). Most importantly, we do not make any assumption on the distribution of the latent features.

4.2.3 Universal approximation capability

With a potentially overcomplete network, we can further achieve a universal approximation of the data distribution. It might initially seem that this is an impossible endeavour given the somehow restricted form of the energy function. However, if we also consider the dimension d_z of $\mathbf{f}$ and $\mathbf{g}$ as an additional architectural parameter that we can change at will, then we can always find an arbitrarily good approximation of the conditional probability density function. This is summarized by the following theorem:

Theorem 4.9. *Let $p(\mathbf{x}|\mathbf{y})$ be a conditional probability density. Assume that $\mathcal{X}$ and $\mathcal{Y}$ are compact Hausdorff spaces*[1] *and that $p(\mathbf{x}|\mathbf{y}) > 0$ almost surely $\forall(\mathbf{x}, \mathbf{y}) \in \mathcal{X} \times \mathcal{Y}$. Further assume that the class of functions $\mathbf{f}_{\boldsymbol{\theta}}$ and $\mathbf{g}_{\boldsymbol{\theta}}$ defined by $\boldsymbol{\theta} \in \Theta$ are dense in the set of continuous functions on $\mathcal{X}$ and $\mathcal{Y}$, respectively.*

[1] See Definitions 4.15 and 4.16 in Appendix 4.C.3

Then for each $\varepsilon > 0$, *there exists* $(\boldsymbol{\theta}, d_z) \in \Theta \times \mathbb{N}$, *where* d_z *is the dimension of* $\mathbf{f}$, *such that*

$$\sup_{(\mathbf{x},\mathbf{y}) \in \mathcal{X} \times \mathcal{Y}} |p_{\boldsymbol{\theta}}(\mathbf{x}|\mathbf{y}) - p(\mathbf{x}|\mathbf{y})| < \varepsilon.$$

This means that our model can approximate any conditional distribution that is positive on its compact support arbitrarily well. In practice, the optimal dimension d_z of the feature extractors can be estimated using cross-validation for instance. It is possible that to achieve a near-perfect approximation, we require a value of d_z that is larger than the dimension of the input. This is why it is crucial that our identifiability result from Theorem 4.4 covers the overcomplete case as well and highlights the importance of our contribution in comparison to previous identifiable deep models.

4.3 An identifiable neural network architecture

In this section, we give a concrete example of a neural network architecture that satisfies the functional assumptions of Theorems 4.4, 4.7 and 4.8. We suppose that the networks $\mathbf{f}$ and $\mathbf{g}$ are parametrized as multi-layer perceptrons (MLP). More specifically, consider an MLP with L layers, where each layer consists of a linear mapping with weight matrix $\mathbf{W}_l \in \mathbb{R}^{d_l \times d_{l-1}}$ and bias $\mathbf{b}_l \in \mathbb{R}^{d_l}$, followed by an activation function h_l. Consider the following architecture:

(A) The activation functions h_l are LeakyReLUs[2], $\forall l \in [\![1, L-1]\!]$.

(B) The weight matrices $\mathbf{W}_l$ are full rank (its rank is equal to its smaller dimension), $\forall l \in [\![1, L]\!]$.

(C) The row dimension of the weight matrices are either monotonically increasing or decreasing: $d_l \geq d_{l+1}, \forall l \in [\![0, L-1]\!]$ or $d_l \leq d_{l+1}, \forall l \in [\![0, L-1]\!]$.

(D) All submatrices of $\mathbf{W}_l$ of size $d_l \times d_l$ are invertible if $d_l < d_{l+1}$, $\forall l \in [\![0, L-1]\!]$.

This architecture satisfies the assumptions of Theorems 4.4, 4.7 and 4.8, as is stated by the propositions below.

[2] A *LeakyReLU* has the form $h_l(x) = \max(0, x) + \alpha \min(0, x),\ \alpha \in (0, 1)$.

Proposition 4.10. *Consider an MLP* $\mathbf{f}$ *whose architecture satisfies assumptions (A) to (C), then* $\mathbf{f}$ *satisfies assumption 1. If in addition,* $d_L \leq d_0$*, then* $\mathbf{f}$ *satisfies assumption 4. Finally, if on top of that, we apply a ReLU (or any positive activation function) to the output of the network, then* $\mathbf{f}$ *satisfies assumption 3.*

Proposition 4.11. *Consider a* nonlinear *MLP* $\mathbf{g}$ *whose architecture satisfies assumptions (A), (B) and (D). Then,* $\mathbf{g}$ *satisfies assumptions 2 and 5.*

While assumptions (A) to (D) might seem a bit restrictive, they serve the important goal of giving sufficient *architectural* conditions that correspond to the purely *functional* assumptions of Theorems 4.4, 4.7 and 4.8. Note that the full rank assumptions are necessary to ensure that the learnt representations are not degenerate since we lose information with low-rank matrices. In practice, random initialisation of floating-point parameters, which are then optimized with stochastic updates (SGD), will result in weight matrices that are almost certainly full rank.

Remark 4.12 (Linear MLPs)**.** The particular case of linear feature extractors is quite interesting. If $d_z \leq d_y$ and the feature extractor $\mathbf{g}$ satisfies the assumptions of Proposition 4.10, then assumption 2 is trivially satisfied. On the other hand, if $d_z > d_y$, then assumption 2 can't hold when the network is linear. This signals that it is important to use *deep* nonlinear networks to parametrize the feature extractors, at least in the overcomplete case.

4.4 Applications

4.4.1 Estimation of identifiable latent variable models

Next, we show how ICE-BeeM relates to the independently modulated component analysis framework (IMCA), a generative latent variable model introduced in Chapter 3 We show how we can use our energy-based model to estimate the latent components. The proofs can be found in Appendix 4.E.

Model definition. Assume we observe a random variable $\mathbf{x} \in \mathbb{R}^{d_x}$ as a result of a nonlinear transformation $\mathbf{h}$ of a latent variable $\mathbf{z} \in \mathbb{R}^{d_z}$. We assume the

distribution of $\mathbf{z}$ is conditioned on an auxiliary variable $\mathbf{y} \in \mathbb{R}^{d_y}$, which is also observed:

$$\begin{aligned} \mathbf{z} &\sim p(\mathbf{z}|\mathbf{y}), \\ \mathbf{x} &= \mathbf{h}(\mathbf{z}). \end{aligned} \tag{4.7}$$

We start by supposing here that $d_x = d_z = d$. The main modelling assumption we make on the latent variable is that its density has the following form:

$$p(\mathbf{z}|\mathbf{y}) = Q(\mathbf{z}) e^{\sum_{i=1}^{d_z} \mathbf{T}_i(z_i)^\top \boldsymbol{\lambda}_i(\mathbf{y}) - \Gamma(\mathbf{y})}, \tag{4.8}$$

where $Q(\mathbf{z})$ is a base measure and $\Gamma(\mathbf{y})$ is the conditional normalizing constant. Crucially, the exponential term factorizes across components: the sufficient statistic $\mathbf{T}$ of this exponential family is composed of d functions that are each a function of only one component z_i of the latent variable $\mathbf{z}$.

Equations (4.7) and (4.8) together define a nonparametric model with parameters $(\mathbf{h}, \mathbf{T}, \boldsymbol{\lambda}, Q)$. For the special case $Q(\mathbf{z}) = \prod_i Q_i(z_i)$, the distribution of $\mathbf{z}$ factorizes across dimensions, and the components z_i are independent. Then the generative model gives rise to a nonlinear ICA model, and it was studied to a great depth in Chapter 2.

In Chapter 3, we proposed to generalize such earlier models by allowing for an arbitrary base measure $Q(\mathbf{z})$, *i.e.* the components of the latent variable are no longer independent, as Q does not necessarily factorize across dimensions.

Estimation by ICE-BeeM. Guided by the strong identifiability results above, we suggest augmenting our feature extractor $\mathbf{f}$ by output activation functions, resulting in the modified feature map

$$\tilde{\mathbf{f}}(\mathbf{x}) = (\mathbf{H}_1(f_1(\mathbf{x})), \ldots, \mathbf{H}_d(f_d(\mathbf{x}))). \tag{4.9}$$

In Section 4.2.2.2 for instance, we used $\mathbf{H}_i(x) = (x, x^2)$. These output nonlinearities play the role of sufficient statistics to the learnt representation $\mathbf{f}_{\boldsymbol{\theta}}(\mathbf{x})$, and have a double purpose: to allow for strong identifiability results, and to match the dimensions of the components $\mathbf{T}_i$ of sufficient statistic in equation (4.8). This augmented feature map is used to define an ICE-BeeM, which in turn is fitted to the data. The identifiability properties of ICE-BeeM, in conjunction with those of IMCA, result in the feature extractor $\mathbf{f}$ to learn the true latent variables, as summarized by the following Theorem.

Theorem 4.13. *Assume:*

(i) The observed data follows the exponential IMCA model of equations (4.7) and (4.8).

(ii) The mixing function $\mathbf{h}$ *is a* $\mathcal{D}^2$*-diffeomorphism*[3]*.*

(iii) The sufficient statistics $\mathbf{T}_i$ *are twice differentiable, and the functions* $T_{ij} \in \mathbf{T}_i$ *are linearly independent on any subset of* $\mathcal{X}$ *of measure greater than zero*[4]*. Furthermore, they all satisfy* $\dim(\mathbf{T}_i) \geq 2$, $\forall i$*;* or $\dim(\mathbf{T}_i) = 1$ *and* $\mathbf{T}_i$ *is non-monotonic* $\forall i$*.*

(iv) There exist $k+1$ *distinct points* $\mathbf{y}^0, \ldots, \mathbf{y}^k$ *such that the matrix*

$$\mathbf{L} = (\boldsymbol{\lambda}(\mathbf{y}_1) - \boldsymbol{\lambda}(\mathbf{y}_0), \ldots, \boldsymbol{\lambda}(\mathbf{y}_k) - \boldsymbol{\lambda}(\mathbf{y}_0))$$

of size $k \times k$ *is invertible, where* $k = \sum_{i=1}^{d} \dim(\mathbf{T}_i)$*.*

(v) We use a consistent estimator to fit the model (4.2) to the conditional density $p(\mathbf{x}|\mathbf{y})$*, where we assume the feature extractor* $\mathbf{f}(\mathbf{x})$ *to be a* $\mathcal{D}^2$*-diffeomorphism and d-dimensional, and the vector-valued pointwise nonlinearities* $\mathbf{H}_i$ *to be differentiable, and their dimensions to be chosen from* $(\dim(\mathbf{T}_1), \ldots, \dim(\mathbf{T}_d))$ *without replacement.*

Then, in the limit of infinite data,

$$\mathbf{H}_i(f_i(\mathbf{x})) = \mathbf{A}_i \mathbf{T}_{\gamma(i)}(z_{\gamma(i)}) + \mathbf{b}_i, \tag{4.10}$$

where γ *is a permutation of* $[\![1, d]\!]$ *such that* $\dim(\mathbf{H}_i) = \dim(\mathbf{T}_{\gamma(i)})$ *and* $\mathbf{A}_i$ *is an invertible square matrix; that is: we can recover the latent variables up to a block permutation linear transformation and pointwise nonlinearities.*

Dimensionality reduction. In practice, it is a natural desire to have the feature extractor reduce the dimension of the data, as it is usually very large. This has been achieved in nonlinear ICA before (Hyvärinen and Morioka, 2016; Khemakhem et al., 2020a). It turns out that we can also incorporate dimensionality reduction in IMCA and its estimation by ICE-BeeM, under some assumptions.

[3] $\mathbf{h}$ is invertible, all second order cross-derivatives of the function and its inverse exist.
[4] In other words, $p(\mathbf{z}|\mathbf{u})$ is strongly exponential, see Definition 2.7.

Suppose that only n out of d components of the latent variable are modulated by the auxiliary variable $\mathbf{y}$. In other words, we assume that the parameters $\boldsymbol{\lambda}_{n+1:d}(\mathbf{y})$ are constant, and we can write its density as

$$p(\mathbf{z}|\mathbf{y}) = Q(\mathbf{z}) e^{\sum_{i=1}^{n} \mathbf{T}_i(z_i)^\top \boldsymbol{\lambda}_i(\mathbf{y}) - \Gamma(\mathbf{y})}. \tag{4.11}$$

The term $e^{\sum_{i=n+1}^{d} \mathbf{T}_i(z_i)^\top \boldsymbol{\lambda}_i}$ is simply absorbed into $Q(\mathbf{z})$. In this case, the feature extractor $\mathbf{f}(\mathbf{x}) \in \mathbb{R}^d$ is still capable of recovering the n modulated components. This is summarised by the following theorem.

Theorem 4.14. *Assume the assumptions of Theorem 4.13 hold. Further assume:*

(vi) Only $n < d$ components of the latent variable are modulated, and its density has the form (4.11).

(vii) The feature extractor $\mathbf{f}$ has the form $\mathbf{f}(\mathbf{x}) = (\mathbf{f}^*(\mathbf{x}), \mathbf{f}^\dagger(\mathbf{x}))$ where $\mathbf{f}^*(\mathbf{x}) \in \mathbb{R}^n$, and the auxiliary feature extractor $\mathbf{g}$ has the form $\mathbf{g}(\mathbf{y}) = (\mathbf{g}^*(\mathbf{y}), \mathbf{g}^\dagger)$ where $\mathbf{g}^*(\mathbf{y}) \in \mathbb{R}^n$ and $\mathbf{g}^\dagger$ is constant.

Then $\mathbf{f}^*$ recovers the n modulated latent components as per equation (4.10).

4.4.2 Transfer learning

As a second practical application of our framework where identifiability is important, we consider meta-learning, particularly multi-task and transfer learning. Assume we have N datasets, which could be, for instance, different subjects in biomedical settings or different image datasets. This fits well with our framework, where $y = 1, \ldots, N$ is now the index of the dataset, or "task". The key question in such a setting is how we can leverage all the observations to better model every single dataset and especially transfer knowledge of existing models to a new dataset.

To this end, we propose an intuitively appealing approach, where we approximate the unnormalized log-pdf in y-th dataset $p(\mathbf{x}; y)$ by a linear combination of a learned "basis" functions $f_{i,\boldsymbol{\theta}}$ as

$$\log p(\mathbf{x}; y) + \log Z(\boldsymbol{\theta}) \approx \sum_{i=1}^{k} g_i(y) f_{i,\boldsymbol{\theta}}(\mathbf{x}), \tag{4.12}$$

where the $g_i(y)$ are scalar parameters as a function of y, which act as coefficients in the basis $(f_{i,\boldsymbol{\theta}})_i$. This linear approximation is nothing else than a special case of ICE-BeeM, but here, we interpret such an approximation as a linear approximation in log-pdf space. In fact, what we are doing is a kind of PCA in the set of probability distributions $p(\mathbf{x}; y)$. Such "probability space" PCA allows the models for the different datasets to learn from each other, as in the classical idea of denoising by projection onto the PCA subspace.

In transfer learning, we observe a new dataset, with distribution $p(\mathbf{x}; y_{\text{new}})$ for $y_{\text{new}} = N + 1$. Based on our decomposition, we approximate $p(\mathbf{x}; y_{\text{new}})$ as in equation (4.12). This leads to a drastic simplification: we can learn the basis functions $f_{i,\boldsymbol{\theta}}$ from the first N datasets, then we only need to estimate the k scalar parameters $g_i(y_{\text{new}})$ for the new dataset. The coefficients are likely to be sparse as well, which provides an additional penalty.

Reducing the transfer learning to the estimation of the $g_i(y_{\text{new}})$ clearly requires that we have estimated the true f_i up to a linear transformation, which is the weaker form of identifiability in Theorem 4.4. This is because the sum in equation (4.12) is essentially a dot-product, which is invariant to linear transformations. Moreover, using a sparsity penalty is only meaningful if we have the true f_i without any linear mixing, which requires the stronger identifiability in Theorems 4.7 and 4.8.

Finally, training can be done by any method for EBM estimation. In particular, it is very easy by score matching because equation (4.12) is an exponential family for fixed f_i (Hyvärinen, 2007).

4.5 Experiments[5]

4.5.1 Identifiability of representations on image datasets

We explore the importance of identifiability and the applicability of ICE-BeeM in a series of experiments on image datasets (MNIST, FashionMNIST, CIFAR10 and CIFAR100). First, we investigate the identifiability of ICE-BeeM by comparing representations obtained from different random initialisations,

[5]Code to reproduce the experiments is available at https://github.com/ilkhem/icebeem.

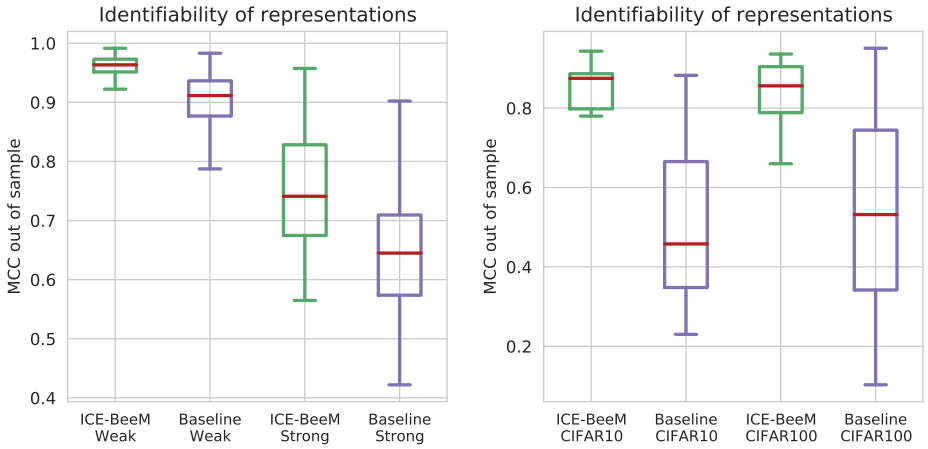
using an unconditional EBM as a baseline. We further present applications to transfer and semi-supervised learning, where we find identifiability leads to significant improvements. Throghout these experiments, the class label is used as the conditioning variable $\mathbf{y}$. The different architectures used throughout these experiments are described in Appendix 4.A.1.

Quantifying identifiability. We start by quantifying the identifiability of the representations learned from image datasets, which serves to empirically validate Theorems 4.4, 4.7 and 4.8. Briefly, these theorems provided conditions for weak and strong identifiability of latent representations, respectively.

We propose to study the weak and strong identifiability properties of both conditional and unconditional EBMs by training such models multiple times using distinct random initialisations. We subsequently compute the mean correlation coefficient (MCC, see Appendix 4.A.2) between learned representations obtained via distinct random initialisations; consistent high MCCs indicate the model is identifiable. In the context of weak identifiability, we consider the MCC up to a linear transformation $\mathbf{A}$, as defined in (4.3). Throughout experiments, we employ canonical correlation analysis (CCA) to learn the linear mapping $\mathbf{A}$. However, our main interest is studying the strong identifiability of EBM architectures, defined in (4.4). To this end we consider the MCC directly on inferred representations (i.e., without a linear mapping $\mathbf{A}$).

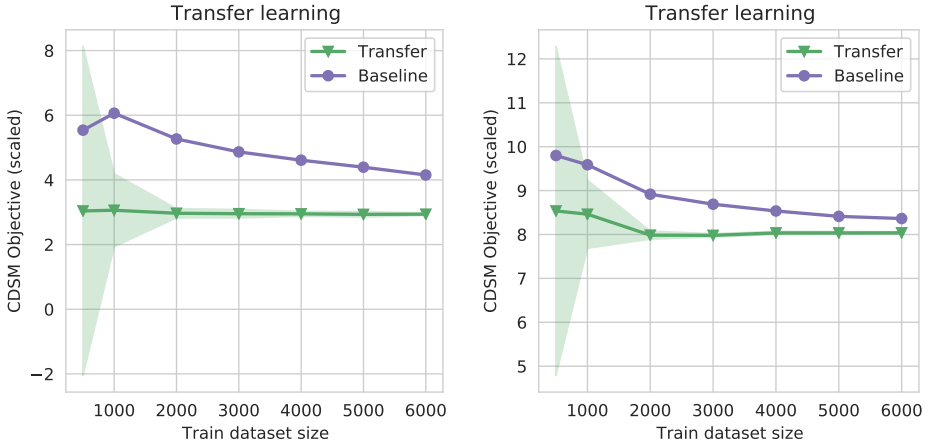
Both an ICE-BeeM model and an unconditional EBM were trained on three different image datasets: MNIST, CIFAR 10 and 100. For each dataset, we train models using 20 different random initialisations and compare inferred representations. Conditional denoising score matching (CDSM, see Appendix 4.B.1) was employed to train all networks. Results presented in Figures [4.1a] and [4.1b] show that for ICE-BeeM, the representations were more consistent, both in the weak and the strong case, thus validating our theory. See Appendix 4.A.3.1 for further details and experiments.

Application to transfer learning. Second, we present an application of ICE-BeeM to transfer learning, as discussed in Section 4.4.2. We suppose that the auxiliary variable $y \in \mathbb{R}$ is the index of a dataset or a task. We propose to approximate the unnormalized log-pdf in y-th dataset $\log p(\mathbf{x}|y)$ by a linear combination of a learned "basis" functions $f_{i,\boldsymbol{\theta}}$ as $\log p(\mathbf{x}; y) + \log Z(\boldsymbol{\theta}) \approx$



(a) MNIST - weak and strong iden. (b) CIFAR10/100 - strong iden.

Figure 4.1: Quantifying the identifiability of learnt representations using MCC (higher is better).



(a) MNIST (b) CIFAR10

Figure 4.2: Transfer learning onto unseen classes using denoising score matching objective (lower is better).

Dataset	$\mathbf{f} \cdot \mathbf{g}_{\boldsymbol{\theta}}$	$\mathbf{f} \cdot \mathbf{1}$	$\mathbf{f}_{\boldsymbol{\theta}} \cdot \mathbf{g}_{\boldsymbol{\theta}}$	$\mathbf{f}_{\boldsymbol{\theta}} \cdot \mathbf{1}$
MNIST	**2.95**	23.43	4.22	3.64
CIFAR10	**8.03**	23.08	8.37	8.16

Table 4.1: Transfer learning — CDSM objective (lower is better)

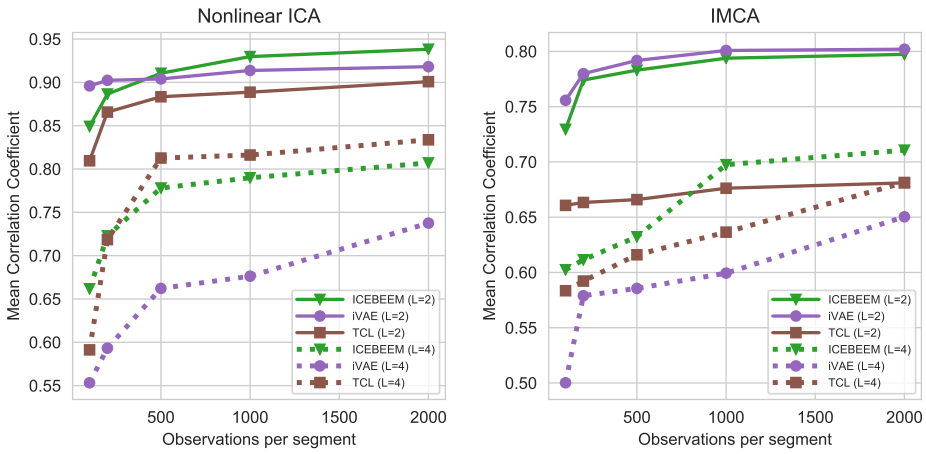
Dataset	ICE-BeeM	Uncond. EBM
FMNIST	$\mathbf{77.07 \pm 1.39}$	56.33 ± 3.18
CIFAR10	$\mathbf{64.42 \pm 1.09}$	51.88 ± 1.33

Table 4.2: Semi-supervised learning — Classification Accuracy (higher is better)

$\sum_{i=1}^{k} g_i(y) f_{i,\boldsymbol{\theta}}(\mathbf{x})$, where the $g_i(y)$ are scalar parameters that act as coefficients in the basis $(f_{i,\boldsymbol{\theta}})$. The basis functions are first learned from the available datasets. For a new dataset y_{new}, the basis is fixed, while the scalar coefficients $g_i(y_{\text{new}}$ are learned.

To this end, an ICE-BeeM model was trained on classes 0-7 of MNIST and CIFAR10 using the CDSM objective. After training, we fix $\mathbf{f}$ and learn $\mathbf{g}_{\boldsymbol{\theta}}(y_{\text{new}})$ for the unseen classes (we denote this by $\mathbf{f} \cdot \mathbf{g}_{\boldsymbol{\theta}}$; unseen classes are 8 & 9). We allow $\mathbf{g}_{\boldsymbol{\theta}}$ to be parametrized by a vector for each class, which leads to a drastic simplification for the new classes. We compare against a baseline where both $\mathbf{f}_{\boldsymbol{\theta}}$ and $\mathbf{g}_{\boldsymbol{\theta}}$ are trained directly on data from unseen classes only (*i.e.* there is no transfer learning—denoted $\mathbf{f}_{\boldsymbol{\theta}} \cdot \mathbf{g}_{\boldsymbol{\theta}}$). Results are presented in Figures [4.2a] and [4.2b] where we vary the sample size of the unseen classes and report the CDSM objective. Overall, the use of a *pretrained* $\mathbf{f}$ network improves performance, demonstrating effective transfer learning. We also compare against a baseline where we just evaluate the pretrained $\mathbf{f}$ on the new classes, while fixing $\mathbf{g} = \mathbf{1}$ (without learning the new coefficients—denoted $\mathbf{f} \cdot \mathbf{1}$); and a baseline where we estimate an unconditional EBM using new classes only (no transfer—denoted $\mathbf{f}_{\boldsymbol{\theta}} \cdot \mathbf{1}$).

The average CDSM scores are reported in Table [4.1], where the transfer learning with an identifiable EBM (i.e., using ICE-BeeM) performs best. See Appendix 4.A.3.2 for further details and experiments. We note here that based on strong identifiability, we could impose sparsity on the coefficients $g_i(y)$, which might improve the results even further.



(a) Simulated nonlinear ICA data (b) Simulated IMCA data

Figure 4.3: Simulations on synthetic nonlinear ICA/IMCA data. The performance is measured using the MCC metric (higher is better).

Application to semi-supervised learning. Finally, we also highlight the benefits of identifiability in the context of semi-supervised learning. We compared training both an identifiable ICE-BeeM model and an unconditional (non-identifiable) EBM on classes 0-7 and employing the learned features $\mathbf{f}_{\boldsymbol{\theta}}$ to classify unseen classes 8-9 using logistic regression. In both cases, training proceeded via CDSM.

Table [4.2] reports the classification accuracy over unseen classes. We note that ICE-BeeM obtains significantly higher classification accuracy, which we attribute to the identifiable nature of its representations. See Appendix 4.A.3.3 for further details and experiments.

4.5.2 IMCA and nonlinear ICA simulations

We run a series of simulations comparing ICE-BeeM to previous nonlinear ICA methods such as iVAE (Khemakhem et al., 2020a) and TCL (Hyvärinen and Morioka, 2016). We generate non-stationary 5-dimensional synthetic datasets, where data is divided into segments, and the conditioning variable $\mathbf{y}$ is defined as a segment index.

First, we let the data follow a nonlinear ICA model, which is a special case of equation (4.7) where the base measure $\mu(\mathbf{z})$, is factorial. Following Hyvärinen and Morioka (2016), the $\mathbf{z}$ are generated according to isotropic

Gaussian distributions with distinct precisions $\boldsymbol{\lambda}(\mathbf{y})$ determined by the segment index.

Second, we let the data follow an IMCA model where the base measure $\mu(\mathbf{z})$ is *not factorial*. We set it to be a Gaussian term with a *fixed* but *non-diagonal* covariance matrix. More specifically, we randomly generate an invertible and symmetric matrix $\boldsymbol{\Sigma}_0 \in \mathbb{R}^{d \times d}$, such that $\mu(\mathbf{z}) \propto e^{-0.5\mathbf{z}^\top \Sigma_0^{-1}\mathbf{z}}$. The covariance matrix of each segment is now equal to $\Sigma(\mathbf{y}) = (\Sigma_0^{-1} + \text{diag}(\boldsymbol{\lambda}(\mathbf{y})))^{-1}$, meaning the latent variables are no longer conditionally independent. In both cases, a randomly initialized neural network with a varying number of layers, $L \in \{2, 4\}$, was employed to generate the nonlinear mixing function $\mathbf{h}$. The data generation process and the employed architectures are detailed in Appendix 4.A.3.4.

In the case of ICE-BeeM, conditional flow contrastive estimation (CFCE, see Appendix 4.B.2) was employed to estimate network parameters. To evaluate the performance of the method, we compute the mean correlation coefficient (MCC, see Appendix 4.A.2) between the true latent variables and the recovered latent variables estimated by all three methods. Results for nonlinear ICA are provided in Figure [4.3a], where we note that ICE-BeeM performs competitively with respect to both iVAE and TCL. We note that as the depth L of the mixing network increases, the performance of all methods decreases. Results for IMCA are provided in Figure [4.3b] where ICE-BeeM outperforms alternative nonlinear ICA methods, particularly when $L = 4$. This is because such other methods implicitly assume latent variables are conditionally independent and are therefore misspecified, whereas in ICE-BeeM, no distributional assumptions on the latent space are made.

4.6 Conclusion

We proposed a new identifiable conditional energy-based deep model, or ICE-BeeM for short, for unsupervised representation learning. This is probably the first energy-based model to benefit from rigorous identifiability results. Crucially, the model benefits from the tremendous flexibility and generality of EBMs. We even prove a universal approximation capability for the model.

Empirically, we showed on real-world image datasets that this model learns identifiable representations in the sense that the representations do not change

arbitrarily from one run to another, and that such representations improve performance in a transfer learning and semi-supervised learning applications. We believe this paves the way for many new applications of EBMs, by giving them a theoretically sound basis.

Appendices to Chapter 4

This chapter has 5 main appendices:

- Appendix 4.A: we give extensive details on the experimental setup, as well as additional experiments.
- Appendix 4.B: we discuss the estimation algorithms we used with ICE-BeeM and how they can be extended to the conditional setting.
- Appendix 4.C: we prove the identifiability of ICE-BeeM and its universal approximation capability.
- Appendix 4.D: we prove the identifiability of a neural network architecture based on fully connected layers.
- Appendix 4.E: we show how ICE-BeeM estimates IMCA.

4.A Experimental protocol

4.A.1 Architectures and hyperparameters

In this section, we describe the neural network architectures used for the experiments of Section 4.5.1, on the image datasets (MNIST, FashionMNIST, CIFAR10 and CIFAR100).

We can distinguish three different types of configurations:

1. A series of fully connected layers — denoted *MLP*. This configuration satisfies the assumptions of Section 4.3.
2. A mix of convolutional and fully connected layers — denoted *ConvMLP*. We expect this configuration to work better than an MLP for images.
3. A variant of a RefineNet (Lin et al., 2017), following Song and Ermon (2019), which implements skip connections to help low level information reach the top layers — denoted for simplicity *Unet* (RefineNets are modern variants of U-net architectures). This configuration is very advanced and complicated, and serves to test if identifiable representations can be learnt for modern architectures.

The detailed architectures are in Table [4.3].

After choosing one of the configurations, we can further chose to reduce the dimensionality of the features ($d_z < d_x$), to use it in conjunction with positive features (assumption 3 of Theorem 4.7) or with augmented features (assumption 4 of Theorem 4.7). This results in the following nomenclature, where we will take as an example a *ConvMLP* network:

- If we reduce the dimension of the latent space ($d_z < d_x$)—for example $d_z = 50$, we denote the configuration by *ConvMLP-50*.
- If we used positive features, we denote the configuration by *ConvMLP-p*.
- If we used augmented features, we denote the configuration by *ConvMLP-a*.
- We can also have a mix of the above, for examples *ConvMLP-50p*.
- We can also have non of the above, in which case we simply write *ConvMLP*—implying that $d_z = d_x$.

We summarize the configurations used for the different experiments of Section 4.5.1 in Table [4.4].

For all the experiments, we used the Adam optimizer (Kingma and Ba, 2014) to update the parameters of the networks. We used a learning rate of 0.001, and $(\beta_1, \beta_2) = (0.9, 0.999)$; `amsgrad` was turned off, as well as weight decay. Data was fed to the networks in mini-batches of size 63, and the training was done for 5000 iterations (no visible improvements in the results were observed after this many iterations). For CIFAR10 and CIFAR100 experiments, we introduced a random horizontal flip to the data, with probability 0.5.

We used conditional denoising score matching (CDSM, Appendix 4.B.1) to train the energy models. The noise parameter used is $\sigma = 0.01$.

4.A.2 The MCC metric

To quantify identifiability, we use the mean correlation coefficient (MCC) developed in Section 2.5.1. Ideally, we should combine the MCC with a measure of independence that is invariant to nonlinear transformations, like the randomized dependence coefficient (Lopez-Paz et al., 2013, RDC).

Configuration	Architecture	Comment
	Input: $d_x = w \times w \times n_c$	n_c: channels, w: width/height
		MNIST: $n_c = 1$, $w = 28$
		FashionMNIST: $n_c = 1$, $w = 28$
		CIFAR10: $n_c = 3$, $w = 32$
	Output: d_z	CIFAR100: $n_c = 3$, $w = 32$
MLP	Input: d_x	
	FC 512, LeakyReLU(0.1)	
	FC 384, LeakyReLU(0.1)	
	Dropout(0.1)	
	FC 256, LeakyReLU(0.1)	
	FC 256, LeakyReLU(0.1)	
	FC d_z	
ConvMLP	Input: $d_x = w \times w \times n_c$	stride 1 for all conv. layers
	Conv $w \times w \times 32$	padding 1, filter size 3
	BatchNorm, ReLU	
	Conv $w \times w \times 64$	padding 1, filter size 3
	BatchNorm, ReLU	
	MaxPool $\frac{w}{2} \times \frac{w}{2} \times 64$	
	Conv $\frac{w}{2} \times \frac{w}{2} \times 128$	padding 1, filter size 3
	BatchNorm, ReLU	
	Conv $\frac{w}{2} \times \frac{w}{2} \times 256$	padding 1, filter size 3
	BatchNorm, ReLU	
	MaxPool $\frac{w}{4} \times \frac{w}{4} \times 256$	
	Conv $1 \times 1 \times 256$	padding 0, filter size $\frac{w}{4}$
	Dropout(0.1)	
	FC 256, LeakyReLU(0.1)	
	FC d_z	
Unet	Input: $d_x = w \times w \times n_c$	stride 1 for all conv. layers
	Conv $w \times w \times 64$	padding 1, filter size 3
	4-cascaded RefineNet	see Song and Ermon (2019)
	| activation: ELU	exponential LU
	| norm.: InstanceNorm+	see Song and Ermon (2019)
	InstanceNorm+, ELU	
	Conv $w \times w \times n_c$	padding 1, filter size 3
	FC d_z	only if $d_z < d_x$

Table 4.3: Architecture detail

Fig./Tab.	Dataset	Description	Configuration
Figure [4.1a]	MNIST	Iden. of representations	*Unet-a*
Figure [4.1b]	CIFAR10	Iden. of representations	*Unet*
Figure [4.1b]	CIFAR100	Iden. of representations	*Unet*
Figure [4.2a]	MNIST	Transfer learning	*ConvMLP-50*
Figure [4.2b]	CIFAR10	Transfer learning	*ConvMLP-90*
Table [4.1]	MNIST	Transfer learning	*ConvMLP-50*
Table [4.1]	CIFAR10	Transfer learning	*ConvMLP-90*
Table [4.2]	FashionMNIST	Semi-supervised learning	*ConvMLP-50*
Table [4.2]	CIFAR10	Semi-supervised learning	*ConvMLP-50p*

Table 4.4: Architectures used in the experiments

When the ground truth is unknown (Section 4.5.1—real image datasets), we compare pairs of learnt representations, each from a different random initialisation. A consistently high MCC means that changing the random state of the model doesn't drastically change the learnt representations.

When the latent ground truth is known (Section 4.5.2—IMCA and nonlinear ICA simulations, for instance), we can test for identifiability of the components by comparing the recovered latents to this ground truth. A high MCC means that we recovered the true latents.

4.A.3 Further experiments

4.A.3.1 Quality of representations

We argued that conditioning enables EBMs to learn identifiable representations. The results in Section 4.5.1 validate this. The plots presented in Figures [4.1a] and [4.1b] were produced using the *Unet* configuration, described in Table [4.4]. This architecture is complex and deep, and involves multiple layers for which a thorough theoretical analysis is very difficult, unlike MLPs for instance. In addition, the dimension of the latent space was chosen to be equal to that of the input space. Intuitively, we would expect that the chance of learning arbitrary representations increases as we increase the number of features because this increases the entropy of the system.

This allows us to challenge the capabilities of ICE-BeeM, and test its limits. We concluded from the results that the theory presented here does benefit

modern deep learning architectures. This experiment serves to empirically validate our theoretical result, and is the first of its kind in recent identifiability literature, which focused on validating the theory on simulated data with well know ground truth.

The matrix $\mathbf{A}$ in equation (4.3) and the permutation σ in equation (4.4) were learnt from the first half of the test partition for each dataset. The evaluation of the MCCs was done on the remaining half of the test dataset.

We present further plots detailing the quality of the learnt representations on MNIST, FashionMNIST, CIFAR10 and CIFAR100 for a variety of different configurations in Figures [4.4] to [4.6].

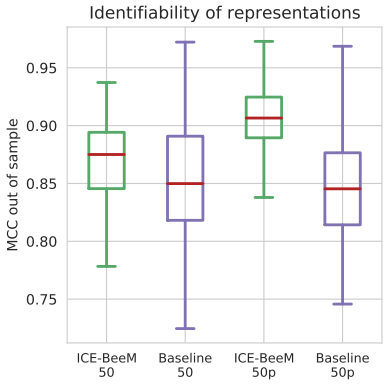
4.A.3.2 Transfer learning experiments

The pre-training was done on labels 0-7 from the train partition for MNIST, FashionMNIST and CIFAR10, and on labels 0-84 from the train partition for CIFAR100. The second (transfer) step was done on labels 8-9 from the train partition for MNIST, FashionMNIST and CIFAR10, and on and labels 85-99 the train partition for CIFAR100.

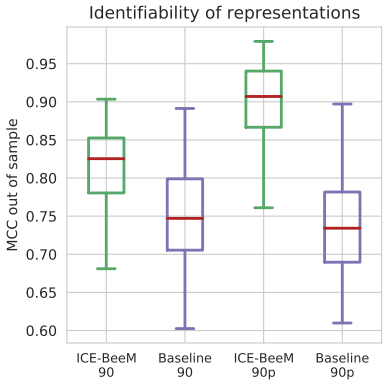
We considered a subset of size 6000 to produce the values in Table [4.1]. This table should be read in conjunction with Figures [4.2a] and [4.2b] for a proper evaluation of performance.

We present further plots and results of transfer learning experiments in Figures [4.8] to [4.10] and Tables [4.5] to [4.8] ran on MNIST, FashionMNIST, CIFAR10 and CIFAR100 respectively, for a variety of different configurations. for different configurations and datasets. We considered a subset of size 6000 to produce the values in these Tables. We expect the baseline where we don't perform transfer learning to perform comparatively for such a subset size: transfer learning is mostly important when data is scarce. For the complete picture, this table should be read in conjunction with Figures [4.8] to [4.10].

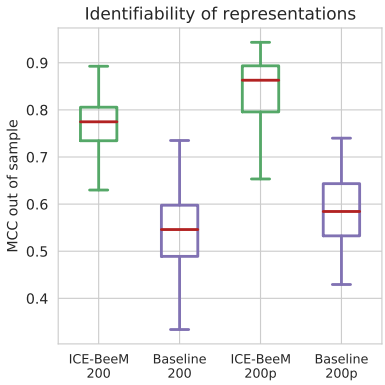
As an additional way to visualize the results, Figure [4.7a] shows unseen MNIST samples (taken across all possible classes) which are assigned high confidence of belonging to the "new" class 8 after transfer learning, indicating that the ICE-BeeM model has learnt a reasonable distribution over unseen classes. By comparison the case where no transfer learning is employed Figure [4.7b]), incorrectly assigns high confidences to other digits.



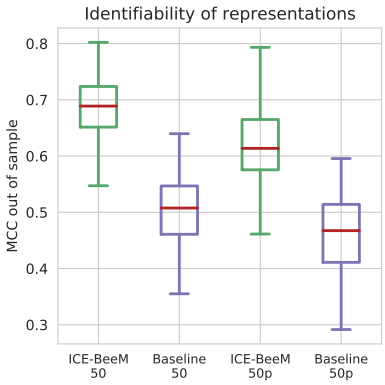
(a) MNIST - *ConvMLP-50/50p*



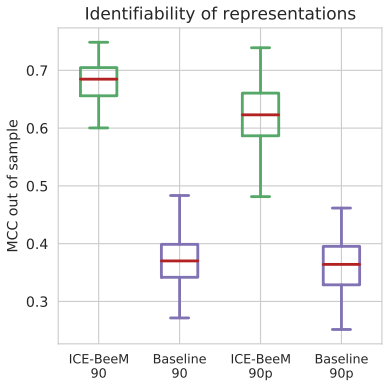
(b) MNIST - *ConvMLP-90/90p*



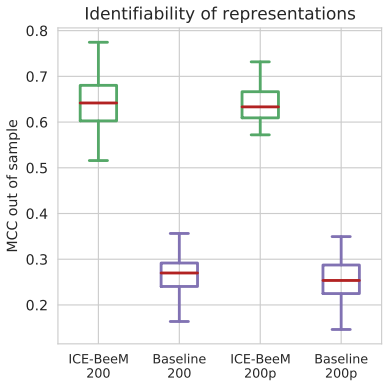
(c) MNIST - *ConvMLP-200/200p*



(d) FMNIST - *ConvMLP-50/50p*

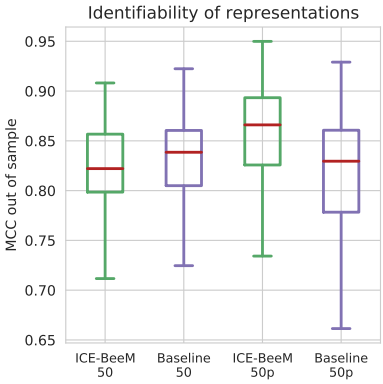


(e) FMNIST - *ConvMLP-90/90p*

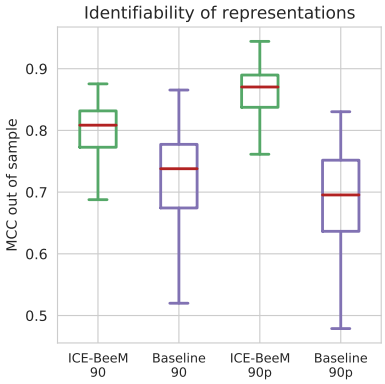


(f) FMNIST - *ConvMLP-200/200p*

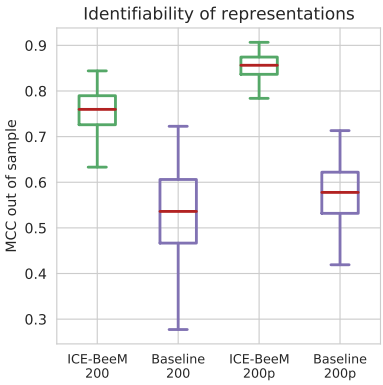
Figure 4.4: Further experiments on the *strong* identifiability of learnt representations using the *ConvMLP* architecture on MNIST and FashionMNIST.



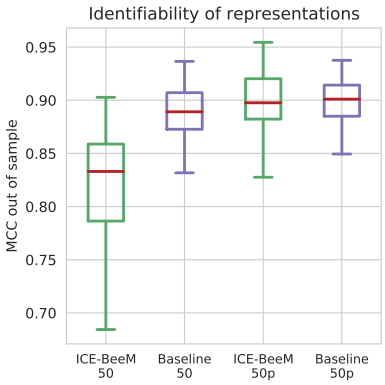
(a) C10 - *ConvMLP-50/50p*



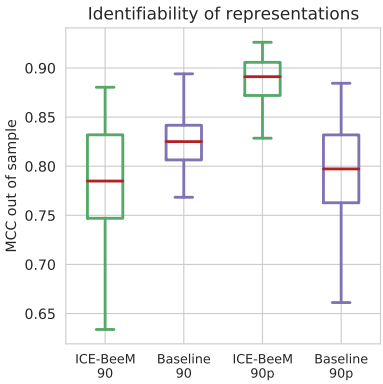
(b) C10 - *ConvMLP-90/90p*



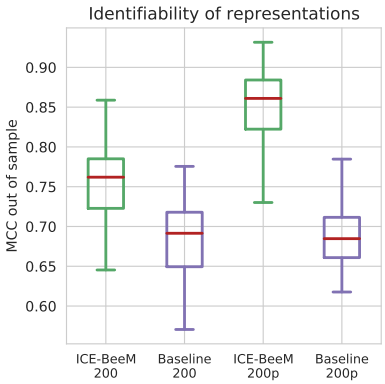
(c) C10 - *ConvMLP-200/200p*



(d) C100 - *ConvMLP-50/50p*



(e) C100 - *ConvMLP-90/90p*



(f) C100 - *ConvMLP-200/200p*

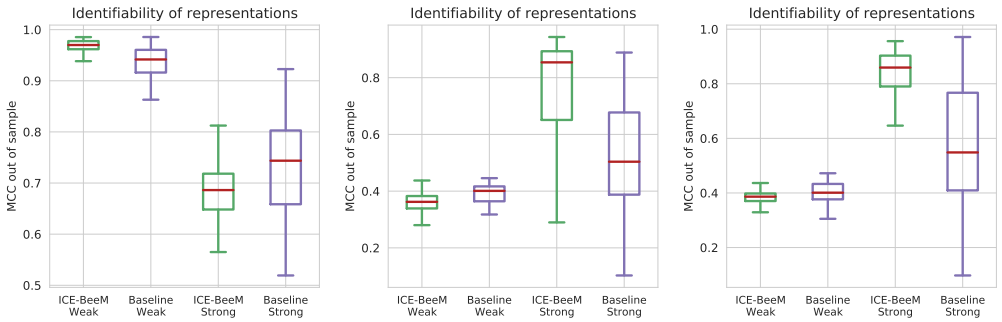
Figure 4.5: Further experiments on the *strong* identifiability of learnt representations using the *ConvMLP* architecture on CIFAR10 and CIFAR100.

Configuration	$\mathbf{f} \cdot \mathbf{g}_{\boldsymbol{\theta}}$	$\mathbf{f} \cdot \mathbf{1}$	$\mathbf{f}_{\boldsymbol{\theta}} \cdot \mathbf{g}_{\boldsymbol{\theta}}$	$\mathbf{f}_{\boldsymbol{\theta}} \cdot \mathbf{1}$
ConvMLP-50	2.95 ± 0.02	23.43 ± 0.04	4.22 ± 0.15	3.64 ± 0.10
ConvMLP-50p	2.79 ± 0.00	796.99 ± 0.86	10.13 ± 4.74	3.63 ± 0.09
ConvMLP-90	2.94 ± 0.01	12.18 ± 0.03	4.29 ± 0.13	3.67 ± 0.12
ConvMLP-90p	3.03 ± 0.01	694.94 ± 1.03	10.22 ± 4.63	3.70 ± 0.12
ConvMLP-200	2.91 ± 0.01	27.70 ± 0.02	4.29 ± 0.12	3.74 ± 0.09
ConvMLP-200p	2.95 ± 0.01	805.45 ± 3.56	12.08 ± 3.79	3.71 ± 0.13
Unet	2.23 ± 0.01	10.04 ± 0.01	3.44 ± 0.03	2.97 ± 0.25
Unet-a	2.29 ± 0.01	6.18 ± 0.00	3.44 ± 0.02	6.27 ± 4.21
Unet-p	14.00 ± 0.01	14.08 ± 0.00	11.97 ± 4.01	6.14 ± 4.17
Unet-50a	2.61 ± 0.02	14.24 ± 0.01	3.79 ± 0.56	2.92 ± 0.20
MLP-50	13.99 ± 0.01	13.99 ± 0.01	14.00 ± 0.01	14.00 ± 0.01
MLP-50p	13.99 ± 0.01	14.00 ± 0.01	14.00 ± 0.01	14.00 ± 0.01
MLP-90	14.00 ± 0.01	14.00 ± 0.01	14.00 ± 0.01	13.99 ± 0.01
MLP-90p	13.99 ± 0.01	14.00 ± 0.01	14.00 ± 0.01	14.00 ± 0.01
MLP-200	13.99 ± 0.01	14.00 ± 0.01	14.00 ± 0.01	14.00 ± 0.01
MLP-200p	13.99 ± 0.01	13.99 ± 0.01	14.00 ± 0.01	14.00 ± 0.01

Table 4.5: Transfer learning — CDSM score on MNIST

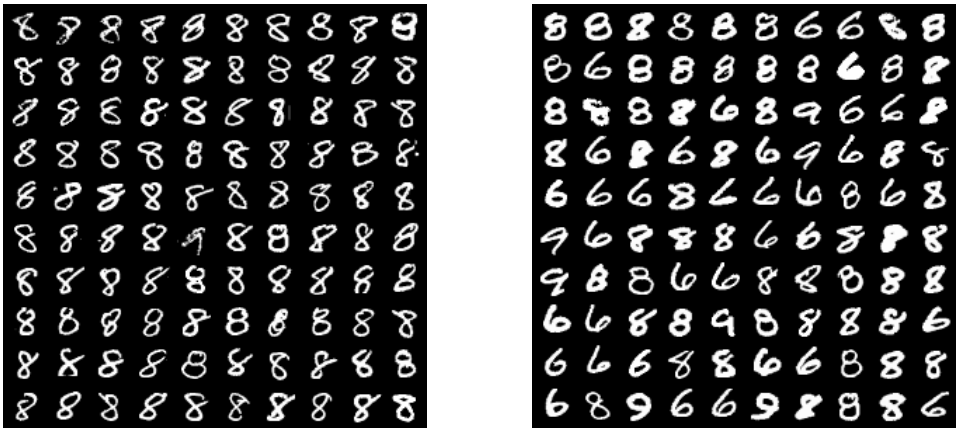
Configuration	$\mathbf{f} \cdot \mathbf{g}_{\boldsymbol{\theta}}$	$\mathbf{f} \cdot \mathbf{1}$	$\mathbf{f}_{\boldsymbol{\theta}} \cdot \mathbf{g}_{\boldsymbol{\theta}}$	$\mathbf{f}_{\boldsymbol{\theta}} \cdot \mathbf{1}$
ConvMLP-50	7.88 ± 0.01	9.82 ± 0.03	7.88 ± 0.07	7.18 ± 0.25
ConvMLP-50p	8.00 ± 0.02	197.84 ± 2.27	7.92 ± 0.18	7.10 ± 0.24
ConvMLP-90	8.09 ± 0.02	10.86 ± 0.04	7.88 ± 0.05	7.14 ± 0.24
ConvMLP-90p	7.94 ± 0.01	197.93 ± 2.33	7.87 ± 0.13	7.13 ± 0.20
ConvMLP-200	7.98 ± 0.00	15.86 ± 0.01	7.91 ± 0.16	7.17 ± 0.21
ConvMLP-200p	7.86 ± 0.01	196.14 ± 2.07	7.81 ± 0.15	7.11 ± 0.15
Unet	6.47 ± 0.02	277.56 ± 1.06	6.52 ± 0.03	6.46 ± 0.07
Unet-a	6.60 ± 0.02	24.62 ± 0.02	6.52 ± 0.02	6.41 ± 0.01
MLP-50	13.99 ± 0.01	14.00 ± 0.01	13.99 ± 0.01	14.00 ± 0.01
MLP-200	13.99 ± 0.01	14.00 ± 0.01	13.99 ± 0.01	14.00 ± 0.01

Table 4.6: Transfer learning — CDSM score on FashionMNIST



(a) MNIST - *Unet* (b) CIFAR10 - *Unet* (c) CIFAR100 - *Unet*

Figure 4.6: Further experiments on the identifiability of representations using the *Unet* architecture on image datasets.

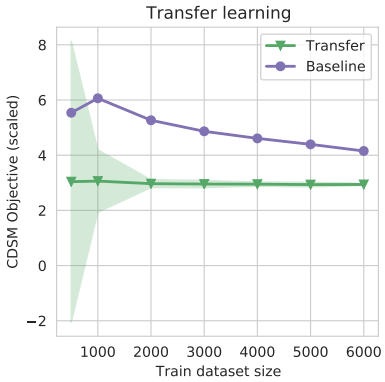


(a) Transfer learning, $\mathbf{f}_{\boldsymbol{\theta}}$ fixed (b) Baseline, both $\mathbf{f}_{\boldsymbol{\theta}}$ and $\mathbf{g}_{\boldsymbol{\theta}}$ estimated

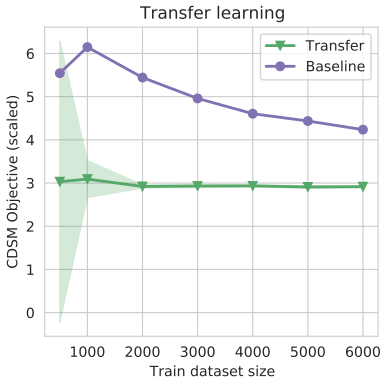
Figure 4.7: Further results for transfer learning experiments on MNIST. In the case of transfer learning 99 out of a hundred returned digits are class 8 compared to only 58 in the baseline.

4.A.3.3 Semi-supervised learning

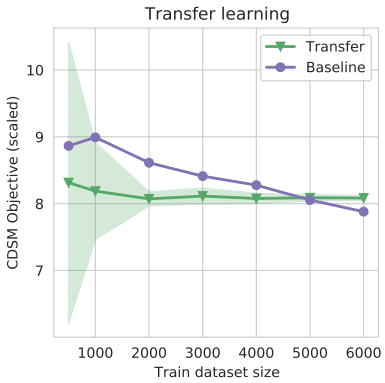
In this experiment, we train both an identifiable ICE-BeeM model and an unconditional (non-identifiable) EBM on classes 0-7. The purpose of this step is to learn a feature extractor $\mathbf{f}_{\boldsymbol{\theta}}$ that is able of learning meaningful features from the images. To test the quality of the features learnt by both models (the ICE-BeeM, and the unconditional EBM), we use the feature map $\mathbf{f}_{\boldsymbol{\theta}}$ to classify unseen samples from classes 8-9. Results show that ICE-BeeM outperforms the unconditional baseline in this classification task. We attribute this to the identifiability of ICE-BeeM: our model seems to be performing a principled



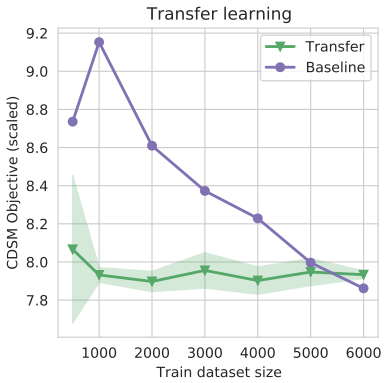
(a) MNIST - *ConvMLP-50*



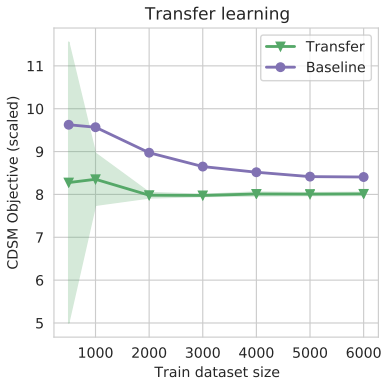
(b) MNIST - *ConvMLP-200*



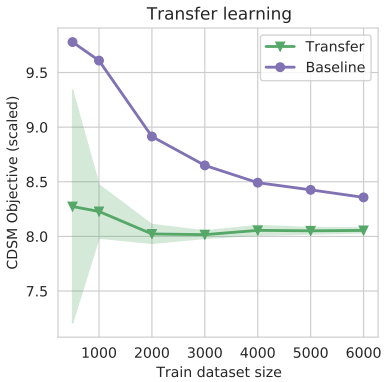
(c) FMNIST - *ConvMLP-90*



(d) FMNIST - *ConvMLP-90p*

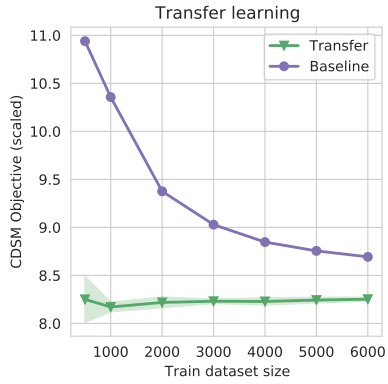


(e) CIFAR10 - *ConvMLP-200*

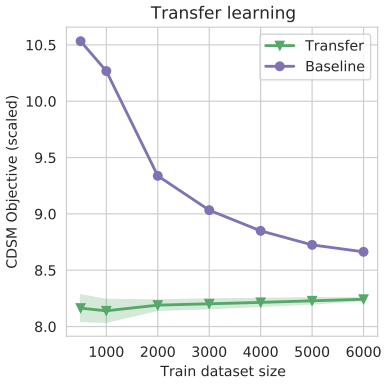


(f) CIFAR10 - *ConvMLP-200p*

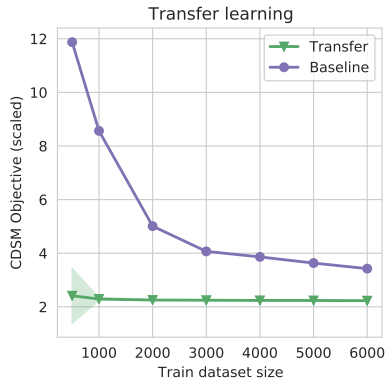
Figure 4.8: Further transfer learning — the dataset/configuration combo are reported in the captions.



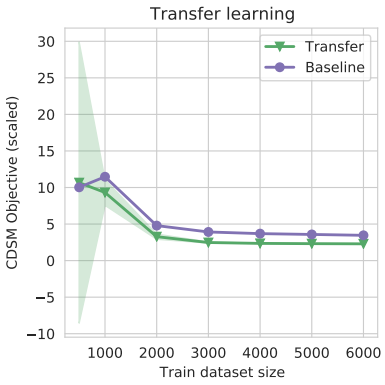
(a) CIFAR100 - *ConvMLP-50*



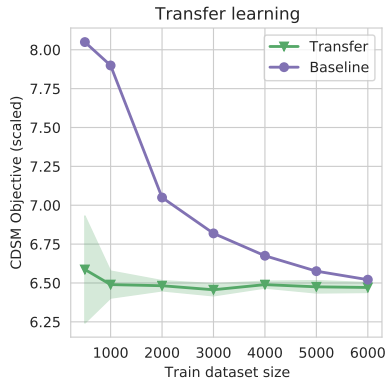
(b) CIFAR100 - *ConvMLP-50p*



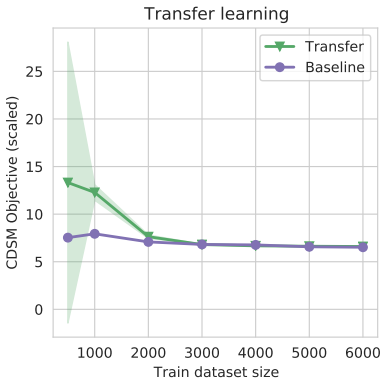
(c) MNIST - *Unet*



(d) MNIST - *Unet-a*

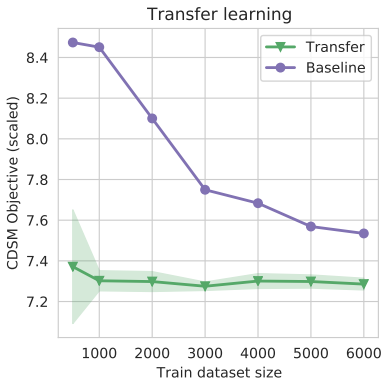


(e) FMNIST - *Unet*

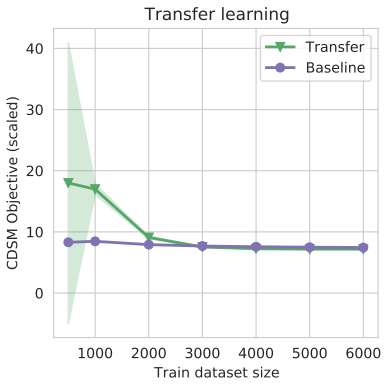


(f) FMNIST - *Unet-a*

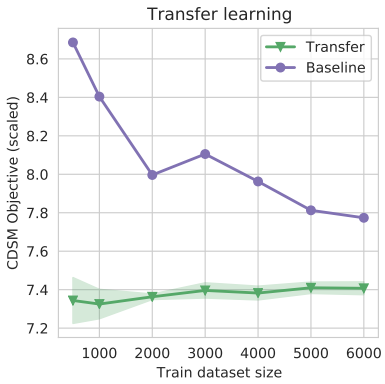
Figure 4.9: Further transfer learning — the dataset/configuration combo are reported in the captions.



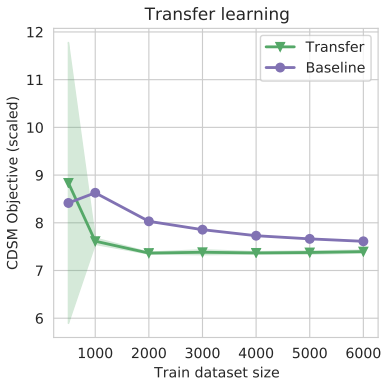
(a) CIFAR10 - *Unet*



(b) CIFAR10 - *Unet-a*



(c) CIFAR100 - *Unet*



(d) CIFAR100 - *Unet-a*

Figure 4.10: Further transfer learning — the dataset/configuration combo are reported in the captions.

Configuration	$\mathbf{f} \cdot \mathbf{g}_{\theta}$	$\mathbf{f} \cdot \mathbf{1}$	$\mathbf{f}_{\theta} \cdot \mathbf{g}_{\theta}$	$\mathbf{f}_{\theta} \cdot \mathbf{1}$
ConvMLP-50	8.02 ± 0.01	32.09 ± 0.07	8.36 ± 0.03	8.15 ± 0.03
ConvMLP-50p	8.04 ± 0.02	412.15 ± 2.54	8.35 ± 0.04	8.17 ± 0.01
ConvMLP-90	8.03 ± 0.01	23.08 ± 0.04	8.37 ± 0.02	8.16 ± 0.05
ConvMLP-90p	8.05 ± 0.01	408.51 ± 2.30	8.37 ± 0.04	8.16 ± 0.01
ConvMLP-200	8.02 ± 0.02	13.35 ± 0.01	8.41 ± 0.07	8.13 ± 0.03
ConvMLP-200p	8.06 ± 0.01	509.09 ± 2.31	8.35 ± 0.02	8.11 ± 0.03
Unet	7.29 ± 0.01	118.93 ± 0.34	7.51 ± 0.05	9.21 ± 3.43
Unet-a	7.18 ± 0.01	18.73 ± 0.01	7.48 ± 0.09	7.47 ± 0.13
Unet-50a	7.30 ± 0.05	16.41 ± 0.00	7.64 ± 0.26	7.27 ± 0.03
MLP-50	16.00 ± 0.00	16.00 ± 0.00	16.00 ± 0.00	16.00 ± 0.00
MLP-200	16.00 ± 0.01	16.00 ± 0.00	16.00 ± 0.01	16.00 ± 0.00

Table 4.7: Transfer learning — CDSM score on CIFAR10

Configuration	$\mathbf{f} \cdot \mathbf{g}_{\boldsymbol{\theta}}$	$\mathbf{f} \cdot \mathbf{1}$	$\mathbf{f}_{\boldsymbol{\theta}} \cdot \mathbf{g}_{\boldsymbol{\theta}}$	$\mathbf{f}_{\boldsymbol{\theta}} \cdot \mathbf{1}$
ConvMLP-50	8.25 ± 0.01	45.19 ± 0.15	8.69 ± 0.04	8.59 ± 0.02
ConvMLP-50p	8.24 ± 0.01	2560.77 ± 7.15	8.68 ± 0.04	8.61 ± 0.04
ConvMLP-90	8.23 ± 0.01	8.74 ± 0.01	8.68 ± 0.05	8.61 ± 0.03
ConvMLP-90p	8.25 ± 0.01	3018.50 ± 7.27	8.65 ± 0.02	8.58 ± 0.03
ConvMLP-200	8.26 ± 0.01	42.80 ± 0.09	8.69 ± 0.06	8.59 ± 0.03
ConvMLP-200p	8.18 ± 0.01	3827.36 ± 16.14	8.65 ± 0.07	8.63 ± 0.05
Unet	7.41 ± 0.02	106.28 ± 0.75	7.77 ± 0.05	8.38 ± 0.55
Unet-a	7.39 ± 0.02	11.15 ± 0.01	7.82 ± 0.42	9.35 ± 3.33
Unet-50a	7.54 ± 0.01	15.95 ± 0.00	7.97 ± 0.13	7.60 ± 0.05
MLP-50p	16.00 ± 0.01	16.00 ± 0.00	16.00 ± 0.00	16.00 ± 0.00
MLP-200p	16.00 ± 0.01	16.00 ± 0.00	16.00 ± 0.00	16.00 ± 0.00

Table 4.8: Transfer learning — CDSM score on CIFAR100

form of disentanglement by learning features that are faithful to the unknown factors of variation in the data.

Training was done on labels 0-7, using the train partition for MNIST, FashionMNIST and CIFAR10. Evaluation was done on labels 8-9, using the test partition for all three datasets. This data was in turn partitioned for the classification into a train and test split. The split proportion is 15% for MNIST and FashionMNIST, and 33% for CIFAR10 and CIFAR100.

We present further results for the semi-supervised learning experiments in Tables [4.9] to [4.11], ran on MNIST, FashionMNIST, CIFAR10 respectively, for a variety of different configurations.

4.A.3.4 IMCA and nonlinear ICA simulations

We give here more detail on the data generation process for the simulations in Section 4.5.2, as well as the architectures used.

Data generation. We generate 5-dimensional synthetic datasets following the nonlinear ICA model which is a special case of equation (4.7) where the base measure, $\mu(\mathbf{z})$, is factorial. In particular, we set it to $\mu(\mathbf{z}) = 1$. As such, latent variables are conditionally independent given segment labels. The sources are divided into $M = 8$ segments, and the conditioning variable $\mathbf{y}$ is defined to be the segment index, uniformly drawn from the integer set $[\![1, M]\!]$. Following Hyvärinen and Morioka (2016), the $\mathbf{z}$ are generated according to

Configuration	ICE-BeeM	Unconditional EBM
ConvMLP-50	76.98 ± 1.61	62.82 ± 1.48
ConvMLP-50p	88.46 ± 1.14	66.58 ± 2.64
ConvMLP-90	78.93 ± 1.51	71.61 ± 1.71
ConvMLP-90p	78.66 ± 1.91	69.13 ± 1.49
ConvMLP-200	81.21 ± 2.6	71.48 ± 2.23
ConvMLP-200p	77.38 ± 1.32	68.99 ± 1.68
MLP-50	91.74 ± 1.72	85.77 ± 1.14
MLP-50p	92.21 ± 1.74	84.56 ± 1.1
MLP-90	95.17 ± 0.46	85.91 ± 2.07
MLP-90p	94.97 ± 0.7	85.97 ± 1.61
MLP-200	94.36 ± 1.28	89.26 ± 1.7
MLP-200p	91.81 ± 2.33	90.87 ± 1.05
Unet	97.79 ± 0.34	98.39 ± 0.68
Unet-a	97.18 ± 0.5	97.79 ± 0.78
Unet-50a	97.52 ± 0.4	97.92 ± 0.49
Unet-20a	95.64 ± 0.7	92.08 ± 1.71

Table 4.9: Semi-supervised learning — classification accuracy on MNIST

Configuration	ICE-BeeM	Unconditional EBM
ConvMLP-50	77.07 ± 1.39	56.33 ± 3.18
ConvMLP-50p	71.67 ± 1.85	57.6 ± 2.24
ConvMLP-90	74.13 ± 1.86	57.73 ± 3.12
ConvMLP-90p	70.87 ± 1.13	60.07 ± 2.9
ConvMLP-200	81.4 ± 1.93	68.27 ± 2.78
ConvMLP-200p	78.47 ± 0.96	57.47 ± 2.62
MLP-50	98.07 ± 1.06	90.47 ± 1.56
MLP-50p	97.6 ± 0.53	90.47 ± 1.56
MLP-90	97.8 ± 0.34	94.4 ± 0.53
MLP-90p	97.8 ± 0.34	94.4 ± 0.53
MLP-200	98.6 ± 0.49	94.87 ± 0.96
MLP-200p	98.6 ± 0.65	95.33 ± 1.05
Unet	99.67 ± 0.3	99.93 ± 0.13
Unet-a	99.53 ± 0.16	99.87 ± 0.16

Table 4.10: Semi-spervised learning — classification accuracy on FashionMNIST

Configuration	ICE-BeeM	Unconditional EBM
ConvMLP-50	69.36 ± 2.23	56.39 ± 1.0
ConvMLP-50p	64.42 ± 1.09	51.88 ± 1.33
ConvMLP-90	68.24 ± 2.0	52.82 ± 0.95
ConvMLP-90p	66.18 ± 1.01	52.33 ± 1.73
ConvMLP-200	64.73 ± 1.36	54.18 ± 1.09
ConvMLP-200p	66.3 ± 0.99	54.48 ± 1.28
MLP-50	68.73 ± 1.35	70.27 ± 2.67
MLP-50p	69.82 ± 1.78	69.36 ± 2.3
MLP-90	71.58 ± 1.21	72.85 ± 1.16
MLP-90p	71.12 ± 1.64	72.85 ± 1.16
MLP-200	72.39 ± 1.92	72.97 ± 1.75
MLP-200p	70.94 ± 1.25	71.97 ± 2.29
Unet	80.27 ± 4.0	80.58 ± 0.9
Unet-a	80.48 ± 1.45	80.48 ± 1.45
Unet-50a	77.64 ± 1.02	73.79 ± 0.81
Unet-20a	74.21 ± 0.73	68.82 ± 0.67

Table 4.11: Semi-supervised learning — classification accuracy on CIFAR10

isotropic Gaussian distributions with distinct precisions $\boldsymbol{\lambda}(\mathbf{y})$ determined by the segment index. Second, we perform the same experiment but on data generated from an IMCA model where the base measure $\mu(\mathbf{z})$ is *not factorial*. More specifically, we randomly generate an invertible and symmetric matrix $\boldsymbol{\Sigma}_0 \in \mathbb{R}^{d \times d}$, such that $\mu(\mathbf{z}) \propto e^{-0.5\mathbf{z}^\top \Sigma_0^{-1}\mathbf{z}}$. As before, we define $\boldsymbol{\lambda}(\mathbf{y})$ to be the distinct conditional precisions. The precision matrix of each segment is now equal to $\Sigma(\mathbf{y})^{-1} = \Sigma_0^{-1} + \operatorname{diag}(\boldsymbol{\lambda}(\mathbf{y}))^{-1}$, meaning the latent variables are no longer conditionally independent.

For both nonlinear ICA and IMCA data, a randomly initialized neural network with varying number of layers, $L \in \{2, 4\}$, was employed to generate the nonlinear mixing function $\mathbf{h}$. Leaky ReLU with negative slope equal to 0.1 was employed as the activation function in order to ensure the network was invertible. The hidden dimensions of the mixing network are equal to the latent dimension d_x, and the output dimension is $d_x = d_z$.

Baseline methods. The first baseline we compare to is TCL (Hyvärinen and Morioka, 2016), which is a self-supervised method for nonlinear ICA based on the non-stationarity of the sources. TCL learns to invert the mixing

function **h**, by performing a surrogate classification task, where the goal is to classify original observations against their segment indices in a multinomial classification task. Its theory is premised on the fact that the feature extractor used for the classification has to extract meaningful latents in order to perform well in the classification task.

The second baseline is iVAE (Khemakhem et al., 2020a), a nonlinear ICA method which uses an identifiable VAE to recover the independent sources. Its theory is premised on the consistency of maximum likelihood training, and on the flexibility of VAEs in approximating densities. They show that given enough data, the variational posterior learns to approximate the true posterior distribution, and can thus be used to invert the mixing function. The iVAE, like a regular VAE, is trained by maximizing the ELBO (Kingma and Welling, 2014).

Training of ICE-BeeM via flow contrastive estimation. To demonstrate that ICE-BeeM can be trained by any method for training EBMs, we switched from denoising score matching to flow contrastive estimation (FCE, Appendix 4.B.2). As a contrastive flow, we used a normalizing flow model (Rezende and Mohamed, 2015), with an isotropic and tractable base distribution. It is then transformed by a 10-layer flow, where each layer is made of a succession of a neural spline flow (Durkan et al., 2019a), an invertible 1×1 convolution (Kingma and Dhariwal, 2018), and an ActNorm layer (Kingma and Dhariwal, 2018). The flow parameters are updated by and Adam optimizer, with a learning rate of 10^{-5}.

Used architectures. The architectures used to produce Figures [4.3a] and [4.3b] are summarized by Table [4.12].

4.B Estimation algorithms

It is important to note that the identifiability results presented above apply to conditional EBMs in general. As such, we may employ any of the wide variety of methods which have been proposed for the estimation of unnormalized EBMs. In this chapter, we used two different options with good results for both: flow contrastive estimation (Gao et al., 2020) and denoising score matching (Vincent,

Model	Optimizer	Architecture	
		Input	$d_x = 5$
		Condition	one hot enc. $d_y = M = 8$
		Latent	$d_z = d_x = 5$
		Num. layers	$L \in \{2, 4\}$
ICE-BeeM	Adam lr 3.10^{-4}	$\mathbf{f}_{\boldsymbol{\theta}}$	$(L+1)$-layer MLP batch norm after each FC hidden dim 32 LeakyReLU(0.1) act
		$\mathbf{g}_{\boldsymbol{\theta}}$	$(d_z \times d_y)$ learnable matrix
iVAE	Adam lr 10^{-3}	Encoder	$p(\mathbf{z}\|\mathbf{x})$ Normal 3-layer MLP hidden dim $2d_x$ LeakyReLU(0.1) act
		Decoder	$p(\mathbf{x}\|\mathbf{z}, \mathbf{y})$ Normal 3-layer MLP hidden dim $2d_x$ LeakyReLU(0.1) act
		Prior	$p(\mathbf{z}\|\mathbf{y})$ Normal 3-layer MLP hidden dim $2d_x$ LeakyReLU(0.1) act
TCL	Momentum 0.9 lr 0.01 exp decay 0.1		L-layer MLP FC $2d_x$, maxout(2) $(L-2)\times$ [FC d_x, maxout(2)] FC d_x, absolute value

Table 4.12: Architectures used in nonlinear ICA and IMCA simulations

2011). We can extend both techniques to the conditional case straightforwardly, as we discuss below.

4.B.1 Conditional denoising score matching

Score matching is a well-known method for learning unnormalized models (Hyvärinen, 2005). However, a well-known problem with the original score matching objective is that it is difficult to use with neural networks because of the many differentiations involved. To solve this issue, Vincent (2011) proposed a stochastic approximation which can be interpreted as denoising the data, and

which works efficiently in deep networks (Saremi et al., 2018; Song and Ermon, 2019).

The original score matching objective can be extended to the conditional setting in a natural way: for a fixed $\mathbf{y}$, we compute the conditional score matching objective: $J(\boldsymbol{\theta}, \mathbf{y}) = \mathbb{E}_{p(\mathbf{x}|\mathbf{y})} \|\nabla_{\mathbf{x}} \log p_{\boldsymbol{\theta}}(\mathbf{x}|\mathbf{y}) - \nabla_{\mathbf{x}} \log p(\mathbf{x}|\mathbf{y})\|^2$, and then average over all values of $\mathbf{y}$ (Arbel and Gretton, 2018). The expression of the conditional score matching objective is then:

$$\mathcal{J}_{\mathrm{CSM}}(\boldsymbol{\theta}) = \mathbb{E}_{p(\mathbf{x},\mathbf{y})} \|\nabla_{\mathbf{x}} \log p_{\boldsymbol{\theta}}(\mathbf{x}|\mathbf{y}) - \nabla_{\mathbf{x}} \log p(\mathbf{x}|\mathbf{y})\|^2 . \tag{4.13}$$

We build on the work of Vincent (2011) and introduce a conditional denoising score matching objective by replacing the unknown density by a kernel density estimator. Formally, given a dataset of observations

$$\mathcal{D} = \left\{ \left(\mathbf{x}^{(1)}, \mathbf{y}^{(1)}\right), \ldots, \left(\mathbf{x}^{(N)}, \mathbf{y}^{(N)}\right) \right\},$$

we first derive nonparametric kernel density estimates of $p(\mathbf{x}, \mathbf{y})$ and $p(\mathbf{y})$, which we then use to derive the estimate for $p(\mathbf{x}|\mathbf{y})$ using the product rule. These estimates have the forms:

$$q_b(\mathbf{y}) = \mathbb{E}_{\mathbf{y}' \sim q_{\mathcal{D}}} \left[l_b(\mathbf{y}|\mathbf{y}') \right], \tag{4.14}$$

$$q_{ab}(\mathbf{x}, \mathbf{y}) = \mathbb{E}_{(\mathbf{x}', \mathbf{y}') \sim q_{\mathcal{D}}} \left[k_a(\mathbf{x}|\mathbf{x}') l_b(\mathbf{y}|\mathbf{y}') \right], \tag{4.15}$$

$$q_{ab}(\mathbf{x}|\mathbf{y}) = \frac{q_{ab}(\mathbf{x}, \mathbf{y})}{q_b(\mathbf{y})}, \tag{4.16}$$

where k_a and l_b are bounded kernel functions defined on $\mathcal{X}$ and $\mathcal{Y}$ and with bandwidths a and b, respectively. The bandwidths satisfy $a = a_N$ and $b = b_N$, and are positive bandwidth sequences which decay to 0 as $N \to +\infty$, where N is the size of the dataset $\mathcal{D}$. In the following, we assume that the bandwidth sequences are equal ($a = b = \sigma$).

We replace $p(\mathbf{x}, \mathbf{y})$ and $p(\mathbf{x}|\mathbf{y})$ in (4.13) by their estimates $q_\sigma(\mathbf{x}, \mathbf{y})$ and $q_\sigma(\mathbf{x}|\mathbf{y})$, to arrive at the new objective

$$\mathcal{J}_{\mathrm{CSM}_\sigma}(\boldsymbol{\theta}) = \mathbb{E}_{q_\sigma(\mathbf{x},\mathbf{y})} \|\nabla_{\mathbf{x}} \log p_{\boldsymbol{\theta}}(\mathbf{x}|\mathbf{y}) - \nabla_{\mathbf{x}} \log q_\sigma(\mathbf{x}|\mathbf{y})\|^2 , \tag{4.17}$$

which is the conditional score matching objective when applied to the nonparametric estimates of the unknown target density. We will show below that it is

equivalent to a simpler objective, in which we only need to compute gradients of the conditioning kernel $k_\sigma(\mathbf{x}|\mathbf{x}')$:

$$\mathcal{J}_{\mathrm{CDSM}_\sigma}(\boldsymbol{\theta}) = \mathbb{E}\|\nabla_{\mathbf{x}} \log p_{\boldsymbol{\theta}}(\mathbf{x}|\mathbf{y}) - \nabla_{\mathbf{x}} \log k_\sigma(\mathbf{x}|\mathbf{x}')\|^2, \tag{4.18}$$

where the expectation is taken with respect to $p_{\mathcal{D}}(\mathbf{x}', \mathbf{y}')k_\sigma(\mathbf{x}|\mathbf{x}')l_\sigma(\mathbf{y}|\mathbf{y}')$. We call this objective conditional denoising score matching. Its extrema landscape is the same as $\mathcal{J}_{\mathrm{CSM}_\sigma}$, but it has the advantage of being simpler to evaluate and interpret.

From CSM to CDSM. We will show here that the stochastic approximation used in denoising score matching can also be used for the conditional case to get to the CDSM objective (4.18) from the CSM objective (4.17):

$$\mathcal{J}_{\mathrm{CSM}_\sigma}(\boldsymbol{\theta}) = \mathbb{E}_{q_\sigma(\mathbf{x},\mathbf{y})} \left\| \nabla_{\mathbf{x}} \log \frac{p_{\boldsymbol{\theta}}(\mathbf{x}|\mathbf{y})}{q_\sigma(\mathbf{x}|\mathbf{y})} \right\|^2 \tag{4.19}$$

$$= \mathbb{E}_{q_\sigma(\mathbf{x},\mathbf{y})} \|\nabla_{\mathbf{x}} \log p_{\boldsymbol{\theta}}(\mathbf{x}|\mathbf{y})\|^2 - S(\boldsymbol{\theta}) + C_1, \tag{4.20}$$

where C_1 is a constant term that only depends on $q_\sigma(\mathbf{x}|\mathbf{y})$, and

$$\begin{aligned}
S(\boldsymbol{\theta}) &= \mathbb{E}_{q_\sigma(\mathbf{x},\mathbf{y})} \langle \nabla_{\mathbf{x}} \log p_{\boldsymbol{\theta}}(\mathbf{x}|\mathbf{y}), \nabla_{\mathbf{x}} \log q_\sigma(\mathbf{x}|\mathbf{y}) \rangle \\
&= \int q_\sigma(\mathbf{x}, \mathbf{y}) \langle \nabla_{\mathbf{x}} \log p_{\boldsymbol{\theta}}(\mathbf{x}|\mathbf{y}), \frac{\nabla_{\mathbf{x}} q_\sigma(\mathbf{x}|\mathbf{y})}{q_\sigma(\mathbf{x}|\mathbf{y})} \rangle \mathrm{d}\mathbf{x}\mathrm{d}\mathbf{y} \\
&= \int q_\sigma(\mathbf{y}) \langle \nabla_{\mathbf{x}} \log p_{\boldsymbol{\theta}}(\mathbf{x}|\mathbf{y}), \nabla_{\mathbf{x}} q_\sigma(\mathbf{x}|\mathbf{y}) \rangle \mathrm{d}\mathbf{x}\mathrm{d}\mathbf{y} \\
&= \int q_\sigma(\mathbf{y}) \langle \nabla_{\mathbf{x}} \log p_{\boldsymbol{\theta}}(\mathbf{x}|\mathbf{y}), \nabla_{\mathbf{x}} \frac{\int p_{\mathcal{D}}(\mathbf{x}', \mathbf{y}') k_\sigma(\mathbf{x}|\mathbf{x}') l_\sigma(\mathbf{y}|\mathbf{y}') \mathrm{d}\mathbf{x}'\mathrm{d}\mathbf{y}'}{q_\sigma(\mathbf{y})} \rangle \mathrm{d}\mathbf{x}\mathrm{d}\mathbf{y} \\
&= \int p_{\mathcal{D}}(\mathbf{x}', \mathbf{y}') l(\mathbf{y}|\mathbf{y}') k(\mathbf{x}|\mathbf{x}') \langle \nabla_{\mathbf{x}} \log p_{\boldsymbol{\theta}}(\mathbf{x}|\mathbf{y}), \nabla_{\mathbf{x}} \log k_\sigma(\mathbf{x}|\mathbf{x}') \rangle \mathrm{d}\mathbf{x}'\mathrm{d}\mathbf{y}'\mathrm{d}\mathbf{x}\mathrm{d}\mathbf{y} \qquad (4.21) \\
&= \mathbb{E}_{p_{\mathcal{D}}(\mathbf{x}',\mathbf{y}')k_\sigma(\mathbf{x}|\mathbf{x}')l_\sigma(\mathbf{y}|\mathbf{y}')} \langle \nabla_{\mathbf{x}} \log p_{\boldsymbol{\theta}}(\mathbf{x}|\mathbf{y}), \nabla_{\mathbf{x}} \log k_\sigma(\mathbf{x}|\mathbf{x}') \rangle,
\end{aligned}$$

where $k = k_\sigma$ and $l = l_\sigma$ in equation (4.21). Plugging this back into equation (4.20), we find that

$$\begin{aligned}
\mathcal{J}_{\mathrm{CSM}_\sigma}(\boldsymbol{\theta}) &= \mathbb{E}\|\nabla_{\mathbf{x}} \log p_{\boldsymbol{\theta}}(\mathbf{x}|\mathbf{y}) - \nabla_{\mathbf{x}} \log k_\sigma(\mathbf{x}|\mathbf{x}')\|^2 + C_1 - C_2 \\
&= \mathcal{J}_{\mathrm{CDSM}_\sigma}(\boldsymbol{\theta}) + C_1 - C_2,
\end{aligned}$$

where the expectation is with respect to $p_{\mathcal{D}}(\mathbf{x}', \mathbf{y}')k_\sigma(\mathbf{x}|\mathbf{x}')l_\sigma(\mathbf{y}|\mathbf{y}')$ and C_2 is another constant that is only a function of $k_\sigma(\mathbf{x}|\mathbf{x}')$. □

Choice of kernels in practice. The CDSM loss developed above works with any choice of kernels that satisfy the aforementioned constrains on their bandwidths. We give now two examples of kernels we can use in practice.

First, we need to select a kernel k_σ for the observation $\mathbf{x}$. A good choice is the Gaussian kernel

$$k_\sigma(\mathbf{x}|\mathbf{x}') = \frac{1}{\sqrt{2\pi\sigma^{2d}}} e^{-\frac{\|\mathbf{x}-\mathbf{x}'\|^2}{2\sigma^2}}. \tag{4.22}$$

Sampling from the Gaussian kernel is fairly straightforward, and the gradient of its logarithm has a simple expression:

$$\nabla_{\mathbf{x}} \log k_\sigma(\mathbf{x}|\mathbf{x}') = \frac{\mathbf{x}' - \mathbf{x}}{\sigma^2}. \tag{4.23}$$

Second, we need to sample noisy observation from the auxiliary variable $\mathbf{y}$. If it is also continuous, then we can chose l_σ to be a Gaussian kernel as well. When $\mathbf{y}$ is discrete (such as a class label), however, the Gaussian kernel can't be used. Instead, we chose to use discrete kernels (Kokonendji and Kiesse, 2011). An example of such kernel when $\mathbf{y}$ can take c different values is

$$l_\sigma(\mathbf{y}|\mathbf{y}') = (1-\sigma)\mathbb{1}_{\mathbf{y}=\mathbf{y}'} + \frac{\sigma}{c-1}\mathbb{1}_{\mathbf{y}\neq\mathbf{y}'}. \tag{4.24}$$

In particular, we will use the identity kernel $l_\sigma(\mathbf{y}|\mathbf{y}') = \mathbb{1}_{\mathbf{y}=\mathbf{y}'}$ in practice, which is a special case of the discrete kernel (4.24) where $\sigma = 0$.

With this choice of kernels, the expression of the loss becomes

$$\mathcal{J}_{\mathrm{CDSM}_\sigma}(\boldsymbol{\theta}) = \mathbb{E}_{p_{\mathcal{D}}(\mathbf{x}',\mathbf{y})k_\sigma(\mathbf{x}|\mathbf{x}')} \left\| \nabla_{\mathbf{x}} \log p_{\boldsymbol{\theta}}(\mathbf{x}|\mathbf{y}) + \frac{\mathbf{x}-\mathbf{x}'}{\sigma^2} \right\|^2. \tag{4.25}$$

4.B.2 Conditional flow contrastive estimation

Flow-contrastive estimation (FCE) can be seen as an extension of noise-contrastive estimation (Gutmann and Hyvärinen, 2012, NCE), which seeks to learn unnormalized EBMs by solving a surrogate classification task. The proposed classification task seeks to discriminate between the true data and some synthetic noise data based on the log-odds ratio of the EBM and the noise distribution. However, a limitation of NCE is the need to specify a noise distribution which can be sampled from and whose log-density can be evaluated pointwise but which also shares some of the empirical properties of the observed data. To address this concern Gao et al. (2020) propose to employ a flow model

as the contrast noise distribution. FCE seeks to simultaneously learn both an unnormalized EBM as well as a flow model for the contrast noise in an alternating fashion.

FCE learns the parameter for the density $p_{\boldsymbol{\theta}}$ of an EBM by performing a surrogate classification task: noise is generated from a noise distribution $q_{\boldsymbol{\alpha}}$ which is parametrized as a flow model, and a logistic regression is performed to classify observation into real data samples or noise samples. The objective function is simply the log-odds:

$$\mathcal{J}_{\mathrm{FCE}}(\boldsymbol{\theta}, \boldsymbol{\alpha}) = \mathbb{E}_{p_{\mathrm{data}(\mathbf{x})}} \log \frac{p_{\boldsymbol{\theta}}(\mathbf{x})}{q_{\boldsymbol{\alpha}}(\mathbf{x}) + p_{\boldsymbol{\theta}}(\mathbf{x})} + \mathbb{E}_{q_{\boldsymbol{\alpha}}(\mathbf{x})} \log \frac{q_{\boldsymbol{\alpha}}(\mathbf{x})}{q_{\boldsymbol{\alpha}}(\mathbf{x}) + p_{\boldsymbol{\theta}}(\mathbf{x})}. \quad (4.26)$$

This objective is minimized with respect to $\boldsymbol{\theta}$ and maximized with respect to $\boldsymbol{\alpha}$: the EBM and the flow model are playing a min-max game. This objective can be extended to the conditional case naturally: we replace the model density by the conditional density $p_{\boldsymbol{\theta}}(\mathbf{x}|\mathbf{y})$.

We naturally get a conditional version for FCE by learning a conditional EBM (see Gao et al., 2020, eq. 12). In this case, it follows that noise samples should also be associated with a conditioning variable, $\mathbf{y}$. One way we can achieve this is by considering a conditional flow. This also has the additional benefit that an improved flow should lead to better estimation of EBM. Alternatively, a standard (non-conditional) flow could be employed, but this would require marginalizing over the conditioning variable, $\mathbf{y}$. The objective simply becomes:

$$\begin{aligned}&\mathcal{J}_{\mathrm{CFCE}}(\boldsymbol{\theta}, \boldsymbol{\alpha}) \\ &\quad = \mathbb{E}_{p_{\mathrm{data}(\mathbf{x},\mathbf{y})}} \log \frac{p_{\boldsymbol{\theta}}(\mathbf{x}|\mathbf{y})}{q_{\boldsymbol{\alpha}}(\mathbf{x}, \mathbf{y}) + p_{\boldsymbol{\theta}}(\mathbf{x}|\mathbf{y})} + \mathbb{E}_{q_{\boldsymbol{\alpha}}(\mathbf{x},\mathbf{y})} \log \frac{q_{\boldsymbol{\alpha}}(\mathbf{x}, \mathbf{y})}{q_{\boldsymbol{\alpha}}(\mathbf{x}, \mathbf{y}) + p_{\boldsymbol{\theta}}(\mathbf{x}|\mathbf{y})}. \quad (4.27)\end{aligned}$$

We can write the flow density as $q_{\boldsymbol{\alpha}}(\mathbf{x}, \mathbf{y}) = p(\mathbf{y}) q_{\boldsymbol{\alpha}}(\mathbf{x}|\mathbf{y})$. This is particularly useful when the conditioning variable $\mathbf{y}$ is discrete, like for instance the index of a dataset or a segment, as we can draw a index from a uniform distribution, and use the conditional flow to sample an observation.

4.C Identifiability of the conditional energy-based model

Recall the form of our conditional energy model

$$p_{\boldsymbol{\theta}}(\mathbf{x}|\mathbf{y}) = Z(\mathbf{y};\boldsymbol{\theta})^{-1} \exp\left(-\mathbf{f}_{\boldsymbol{\theta}}(\mathbf{x})^\top \mathbf{g}_{\boldsymbol{\theta}}(\mathbf{y})\right). \tag{4.28}$$

We present in this section the proofs for the different forms of identifiability that is guaranteed for the feature extractors $\mathbf{f}$ and $\mathbf{g}$. We will focus on the proofs for the feature extractor $\mathbf{f}$, as the proofs for the feature extractor $\mathbf{g}$ are very similar.

For the rest of the Appendix, we will denote by $d = d_x$, $m = d_y$ and $n = d_z$.

4.C.1 Weak identifiability

Proof of Theorem 4.4. We will only prove this theorem for the feature extractor $\mathbf{f}$. The proof for $\mathbf{g}$ is very similar. Suppose assumptions 1 and 2 hold.

Consider two parameters $\boldsymbol{\theta}$ and $\tilde{\boldsymbol{\theta}}$ such that

$$p_{\boldsymbol{\theta}}(\mathbf{x}|\mathbf{y}) = p_{\tilde{\boldsymbol{\theta}}}(\mathbf{x}|\mathbf{y}). \tag{4.29}$$

Then, by applying the logarithm to both sides, we get:

$$\log Z(\mathbf{y};\boldsymbol{\theta}) - \mathbf{f}_{\boldsymbol{\theta}}(\mathbf{x})^\top \mathbf{g}_{\boldsymbol{\theta}}(\mathbf{y}) = \log Z(\mathbf{y};\tilde{\boldsymbol{\theta}}) - \mathbf{f}_{\tilde{\boldsymbol{\theta}}}(\mathbf{x})^\top \mathbf{g}_{\tilde{\boldsymbol{\theta}}}(\mathbf{y}). \tag{4.30}$$

Consider the points $\mathbf{y}^0, \dots, \mathbf{y}^n$ provided by assumption 2 for $\mathbf{g}_{\boldsymbol{\theta}}$. We plug each of these points in equation (4.30) to obtain $n+1$ such equations. We subtract the first equation for $\mathbf{y}^0$ from the remaining n equations, and write the resulting equations in matrix form:

$$\mathbf{R}\mathbf{f}_{\boldsymbol{\theta}}(\mathbf{x}) = \tilde{\mathbf{R}}\mathbf{f}_{\tilde{\boldsymbol{\theta}}}(\mathbf{x}) + \mathbf{b}, \tag{4.31}$$

where $\mathbf{R} = (\dots, \mathbf{g}_{\boldsymbol{\theta}}(\mathbf{y}^l) - \mathbf{g}_{\boldsymbol{\theta}}(\mathbf{y}^0), \dots)$, $\tilde{\mathbf{R}} = (\dots, \mathbf{g}_{\tilde{\boldsymbol{\theta}}}(\mathbf{y}^l) - \mathbf{g}_{\tilde{\boldsymbol{\theta}}}(\mathbf{y}^0), \dots)$, and $\mathbf{b} = (\dots, \log \frac{Z(\mathbf{y}^l;\boldsymbol{\theta})}{Z(\mathbf{y}^l;\tilde{\boldsymbol{\theta}})} - \log \frac{Z(\mathbf{y}^0;\boldsymbol{\theta})}{Z(\mathbf{y}^0;\tilde{\boldsymbol{\theta}})}, \dots)$. Since $\mathbf{R}$ is invertible (by assumption 2), we multiply by its inverse from the left to get:

$$\mathbf{f}_{\boldsymbol{\theta}}(\mathbf{x}) = \mathbf{A}\mathbf{f}_{\tilde{\boldsymbol{\theta}}}(\mathbf{x}) + \mathbf{c}, \tag{4.32}$$

where $\mathbf{A} = \mathbf{R}^{-1}\tilde{\mathbf{R}}$ and $\mathbf{c} = \mathbf{R}^{-1}\mathbf{b}$. Now since $\mathbf{f}_{\boldsymbol{\theta}}$ is differentiable and its Jacobian is full rank assumption 1), by differentiating the last equation we deduce that $\operatorname{rank}(\mathbf{A}) \geq \min(n, d)$, which in turn proves that $\boldsymbol{\theta} \sim_w^{\mathbf{f}} \tilde{\boldsymbol{\theta}}$.

Finally, suppose that in addition, assumption 2 holds for the feature extractor $\mathbf{f}_{\boldsymbol{\theta}}$. Then there exists $\mathbf{x}^0, \dots \mathbf{x}^n$ such that $\mathbf{Q} \coloneqq (\dots, \mathbf{f}_{\boldsymbol{\theta}}(\mathbf{x}^i) - \mathbf{f}_{\boldsymbol{\theta}}(\mathbf{x}^0), \dots)$. Plugging these $n+1$ points into equation (4.32), and subtracting the first equation for $\mathbf{x}^0$ from the remaining n equations, we get

$$\mathbf{Q} = \mathbf{A}(\dots, \mathbf{f}_{\tilde{\boldsymbol{\theta}}}(\mathbf{x}^i) - \mathbf{f}_{\tilde{\boldsymbol{\theta}}}(\mathbf{x}^0), \dots). \tag{4.33}$$

Since $\mathbf{Q}$ is an $n \times n$ invertible matrix, we conclude that $\mathbf{A}$ is also invertible, which concludes the proof. □

4.C.2 Strong identifiability

Proof of Theorem 4.7. We will prove here a more general version where we assume that each component f_i of the feature extractor $\mathbf{f}$ has a global minimum that is reached, instead of being necessarily non-negative.

Consider two different parameters $\boldsymbol{\theta}$ and $\tilde{\boldsymbol{\theta}}$ such that:

$$p_{\boldsymbol{\theta}}(\mathbf{x}|\mathbf{y}) = p_{\tilde{\boldsymbol{\theta}}}(\mathbf{x}|\mathbf{y}). \tag{4.34}$$

To simplify notations, denote by $\mathbf{f} = \mathbf{f}_{\boldsymbol{\theta}}$ and $\tilde{\mathbf{f}} = \mathbf{f}_{\tilde{\boldsymbol{\theta}}}$. We start the proof from the conclusion of Theorem 4.4, since its assumptions hold:

$$\mathbf{f}(\mathbf{x}) = \mathbf{A}\tilde{\mathbf{f}}(\mathbf{x}) + \mathbf{c}, \tag{4.35}$$

where $\mathbf{A}$ is an invertible $n \times n$ matrix and $\mathbf{c}$ a constant vector. Without loss of generality, we can suppose that f_i has an infimum equal to zero, simply by subtracting $\inf f_i$, and including in $\mathbf{c}$, and similarly for $\tilde{\mathbf{f}}$. We will also suppose that the infima are reached, as the next argument would hold if we change exact minima by limits.

Now since $\mathbf{f} \geq 0$ and is surjective, then there exists $\mathbf{x}_0 \in \mathbb{R}^d$ such that $\mathbf{f}(\mathbf{x}_0) = 0$. This implies that $\mathbf{c} = -\mathbf{A}\tilde{\mathbf{f}}(\mathbf{x}_0)$, and that $\mathbf{f}(\mathbf{x}) = \mathbf{A}(\tilde{\mathbf{f}}(\mathbf{x}) - \tilde{\mathbf{f}}(\mathbf{x}_0))$. Define $\mathbf{h}(\mathbf{x}) = \tilde{\mathbf{f}}(\mathbf{x}) - \tilde{\mathbf{f}}(\mathbf{x}_0)$. We know that $\tilde{\mathbf{f}} \geq 0$ and is surjective, and so $\mathbf{h}$ is also surjective, and its image includes $\mathbb{R}^n_+$. Let $\mathbf{I} = (\mathbf{e}_1, \dots, \mathbf{e}_n)$ be the matrix of canonical basis vectors, or positive scalar multiples of the canonical basis

vectors $\mathbf{e}_i$. These must be mapped to the non-negative quadrant, so $\mathbf{AI}$ must be non-negative, which implies that $\mathbf{A}$ must be non-negative.

Denote by $\mathbf{B} = \mathbf{A}^{-1}$. $\mathbf{B}$ is also non-negative for the same reasons described above. Denote the **rows** of $\mathbf{A}$ by $\mathbf{a}_i$ and the **columns** of $\mathbf{B}$ by $\mathbf{b}_j$. We have by definition of inverse:

$$\mathbf{a}_i^\top \mathbf{b}_j = \delta_{ij}, \tag{4.36}$$

where if $i = j$ then $\delta_{ij} = 1$, else $\delta_{ij} = 0$. Now, assume there is a row $\mathbf{a}_k$ which has at least two nonzero entries. By the property above, $d-1$ of the vectors $\mathbf{b}_j$ must have zero dot-product with that vector. By non-negativity of $\mathbf{B}$ and $\mathbf{A}$, those $d-1$ vectors must have zeros in the at least two indices corresponding to the nonzeros of $\mathbf{a}_k$. But that means they can only span a $d-2$-dimensional subspace, and all the $\mathbf{b}_j$ together can only span a $d-1$-dimensional subspace. This is in contradiction of the invertibility of $\mathbf{B}$. Thus, each $\mathbf{a}_i$ can have only one nonzero entry, which, together with the invertibility of $\mathbf{A}$, proves it is a scaled permutation matrix.

Thus, there exists a permutation σ of $[\![1, n]\!]$, such that

$$f_i(\mathbf{x}) = a_{i,\sigma(i)} \tilde{f}_{\sigma(i)}(\mathbf{x}) + c_i, \tag{4.37}$$

which concludes the proof. □

Proof of Theorem 4.8. Similarly to the proof of Theorem 4.7, we pass the features f_i through the nonlinear function $\mathbf{H}_i(f_i) = (f_i, f_i^2)$ which produces the augmented features $\tilde{\mathbf{f}}$ introduced in Section 4.2.2.2.

Consider two different parameters $\boldsymbol{\theta}$ and $\tilde{\boldsymbol{\theta}}$ such that

$$p_{\boldsymbol{\theta}}(\mathbf{x}|\mathbf{y}) = p_{\tilde{\boldsymbol{\theta}}}(\mathbf{x}|\mathbf{y}). \tag{4.38}$$

Since we have similar assumptions to Theorem 4.4, we will skip the first part of the proof and make the same conclusion, where the equivalence up to linear transformation here applies to $\mathbf{H}(\mathbf{f}_{\boldsymbol{\theta}})$ and $\mathbf{H}(\mathbf{f}_{\tilde{\boldsymbol{\theta}}})$:

$$\mathbf{H}(\mathbf{f}_{\boldsymbol{\theta}}(\mathbf{x})) = \mathbf{A}\mathbf{H}(\mathbf{f}_{\tilde{\boldsymbol{\theta}}}(\mathbf{x})) + \mathbf{c}, \tag{4.39}$$

where $\mathbf{A}$ is a $2n \times 2n$ matrix of rank at least n because $\mathbf{J}_{\mathbf{f}}$ and $\mathbf{J}_{\mathbf{H}}$ are full rank ($\mathbf{A}$ is not necessarily invertible yet, but this will be proven later) and $\mathbf{c}$ a constant vector. By replacing $\mathbf{H}$ by its expression, we get

$$\begin{pmatrix} \mathbf{f}_{\boldsymbol{\theta}}(\mathbf{x}) \\ \mathbf{f}^2_{\boldsymbol{\theta}}(\mathbf{x}) \end{pmatrix} = \begin{pmatrix} \mathbf{A}^{(1)} & \mathbf{A}^{(2)} \\ \mathbf{A}^{(3)} & \mathbf{A}^{(4)} \end{pmatrix} \begin{pmatrix} \mathbf{f}_{\tilde{\boldsymbol{\theta}}}(\mathbf{x}) \\ \mathbf{f}^2_{\tilde{\boldsymbol{\theta}}}(\mathbf{x}) \end{pmatrix} + \begin{pmatrix} \boldsymbol{\alpha} \\ \boldsymbol{\beta} \end{pmatrix}, \tag{4.40}$$

where each $\mathbf{A}^{(i)}$ is an $n \times n$ matrix, and $\mathbf{c} = (\boldsymbol{\alpha}, \boldsymbol{\beta})$. To simplify notations, denote by $\mathbf{h} = \mathbf{f}_{\tilde{\boldsymbol{\theta}}}$. We will also drop reference to $\boldsymbol{\theta}$ and $\tilde{\boldsymbol{\theta}}$. The first n lines in the previous equation are

$$f_i(\mathbf{x}) = \sum_{j=1}^{n} A_{ij}^{(1)} h_j(\mathbf{x}) + A_{ij}^{(2)} h_j^2(\mathbf{x}) + \alpha_i, \tag{4.41}$$

and the last n lines are

$$f_i^2(\mathbf{x}) = \sum_{j=1}^{n} A_{ij}^{(3)} h_j(\mathbf{x}) + A_{ij}^{(4)} h_j^2(\mathbf{x}) + \beta_i. \tag{4.42}$$

Fix an index i in equations (4.41) and (4.42). To alleviate notations and reduce the number of subscripts and superscripts, we introduce $a_j = A_{ij}^{(1)}$, $b_j = A_{ij}^{(2)}$, $c_j = A_{ij}^{(3)}$, $d_j = A_{ij}^{(4)}$, $\alpha = \alpha_i$ and $\beta = \beta_i$. This proof is done in 5 steps. Note that the surjectivity assumption is key for the rest of the proof, and it requires that we set the dimension of the feature extractor to be lower than the dimension of the observations.

By equating equations (4.41) and (4.42) after squaring, we get, using our new notations:

$$\left(\sum_j a_j h_j(\mathbf{x}) + b_j h_j^2(\mathbf{x}) + \alpha \right)^2 = \sum_j c_j h_j(\mathbf{x}) + d_j h_j^2(\mathbf{x}) + \beta. \tag{4.43}$$

Step 1. First, since $\mathbf{h}$ is surjective, there exists a point where it is equal to zero. Evaluating equation (4.43) at this point shows that $\beta = \alpha^2$.

Step 2. Second, the left hand side of equation (4.43) has terms raised to the power 4. These terms grow to infinity much faster than the rest of the terms of the rhs and the lhs. It is thus equal to zero. More rigorously, consider the vectors $\mathbf{e}_l(y) = (0, \dots, y, \dots, 0) \in \mathbb{R}^n$ where the only non zero entry is y at the l-th position. Each of these vectors has a preimage by $\mathbf{h}$ (since it is surjective), which we denote by $\mathbf{x}_l(y)$. By evaluating equation (4.43) at each of these points, we get

$$(a_l y + b_l y^2 + \alpha)^2 = c_l y + d_l y^2 + \beta. \tag{4.44}$$

Divide both sides of this equation by y^4, then take the limit $y \to \infty$. The right hand side will converge to 0, while the left hand side will converge to b_l, which shows that $b_l = 0$. By doing this process for all $l \in [\![1, n]\!]$, we can show that $\mathbf{b} = 0$.

Step 3. So far, we've shown that equation (4.43) becomes, after expanding the square in the lhs, and writing $\sum_j a_j h_j(\mathbf{x}) = \mathbf{a}^\top \mathbf{h}(\mathbf{x})$:

$$(\mathbf{a}^\top \mathbf{h}(\mathbf{x}))^2 + 2\alpha \mathbf{a}^\top \mathbf{h}(\mathbf{x}) + \alpha^2 = \sum_j c_j h_j(\mathbf{x}) + d_j h_j^2(\mathbf{x}) + \alpha^2. \tag{4.45}$$

Let's again consider the vectors $\mathbf{e}_l(y)$ from earlier, and their preimages $\mathbf{x}_l(y)$. By evaluating equation (4.45) at the points $\mathbf{x}_l(y)$, we get

$$a_l^2 y^2 + 2\alpha a_l y + \alpha^2 = c_l y + d_l y^2 + \alpha^2. \tag{4.46}$$

Divide both sides by y, and take the limit $y \to 0$. The lhs converges to $2\alpha a_l$, while the rhs converges to c_l. Since this is valid for all $l \in [\![1, n]\!]$, we conclude that $\mathbf{c} = 2\alpha \mathbf{a}$. It also follows that $\mathbf{d} = \mathbf{a}^2$.

Step 4. Injecting this back into equation (4.45), and writing $\sum_j d_j h_j^2(\mathbf{x}) = \mathbf{h}(\mathbf{x})^\top \operatorname{diag}(\mathbf{d}) \mathbf{h}(\mathbf{x})$, we are left with

$$(\mathbf{a}^\top \mathbf{h}(\mathbf{x}))^2 = \mathbf{h}(\mathbf{x})^\top \operatorname{diag}(\mathbf{d}) \mathbf{h}(\mathbf{x}). \tag{4.47}$$

By applying the trace operator to both sides of this equation, and rearranging terms, we get

$$\operatorname{trace}\left(\left(\mathbf{a}\mathbf{a}^\top - \operatorname{diag}(\mathbf{d})\right) \mathbf{h}(\mathbf{x}) \mathbf{h}(\mathbf{x})^\top\right) = 0, \tag{4.48}$$

which is of the form $\operatorname{trace}(\mathbf{C}^\top \mathbf{B}(\mathbf{x})) = 0$. This is a dot product on the space $\mathcal{S}_n$ of $n \times n$ symmetric matrices (both $\mathbf{C}$ and $\mathbf{B}(\mathbf{x})$ are symmetric!), which is a vector space of dimension $\frac{n(n+1)}{2}$. If we can show that the matrix $\mathbf{C}$ is orthogonal to a basis of $\mathcal{S}_n$, then we can conclude that $\mathbf{C} = 0$.

For this, let $(\mathbf{e}_j)_{1 \leq j \leq n}$ be the Euclidean basis of $\mathbb{R}^n$, where each vector $\mathbf{e}_j$ has one nonzero entry equal to 1 at index j, and let $(\mathbf{E}_{ij})_{1 \leq i \leq n, 1 \leq j \leq n}$ be the Euclidean basis of $\mathbb{R}^{n \times n}$, where each matrix $\mathbf{E}_{ij}$ has only one nonzero entry equal to 1 at row i and column j.

Now since $\mathbf{h}$ is surjective, there exists $\mathbf{x}_j$ such that

$$\mathbf{h}(\mathbf{x}_j) = \mathbf{e}_j, \tag{4.49}$$

$$\mathbf{h}(\mathbf{x}_j) \mathbf{h}(\mathbf{x}_j)^\top = \mathbf{e}_j \mathbf{e}_j^\top = \mathbf{E}_{jj}. \tag{4.50}$$

The n different $\mathbf{x}_j$ give us our first n matrices we will use to construct a basis of $\mathcal{S}_n$. We now need to find $\frac{n(n-1)}{2}$ remaining basis matrices. For this, consider

the sums $(\mathbf{e}_j + \mathbf{e}_l)_{1 \leq j < l \leq n}$, of which there is exactly $\frac{n(n-1)}{2}$. Each of these sums of vectors have a preimage $\mathbf{x}_{j,l}$ by $\mathbf{h}$, and $\mathbf{h}(\mathbf{x}_{j,l})\mathbf{h}(\mathbf{x}_{j,l})^\top = (\mathbf{e}_j + \mathbf{e}_l)(\mathbf{e}_j + \mathbf{e}_l)^\top = \mathbf{E}_{jj} + \mathbf{E}_{ll} + (\mathbf{E}_{il} + \mathbf{E}_{li})$, which is a matrix in $\mathcal{S}_n$ that is linearly independent of all $\mathbf{E}_{jj}$, and all other $(\mathbf{e}_s + \mathbf{e}_t)(\mathbf{e}_s + \mathbf{e}_t)^\top$ where $(s,t) \neq (j,l)$ because they have nonzero entries at different rows and columns.

We have then found $\frac{n(n+1)}{2}$ distinct vectors $(\mathbf{x}_1, \ldots, \mathbf{x}_n, \mathbf{x}_{1,2}, \ldots, \mathbf{x}_{n-1,n})$ such that their images by $\mathbf{h}\mathbf{h}^\top$ form a basis of $\mathcal{S}_n$. If we now evaluate equation (4.48) at each of these points, we find that the matrix $\mathbf{a}\mathbf{a}^\top - \operatorname{diag}(\mathbf{d})$ is orthogonal to a basis of $\mathcal{S}_n$, which implies that it is necessarily equal to 0. This in turn implies that $\mathbf{a}\mathbf{a}^\top$ is a diagonal matrix, and that $a_j a_l = 0$ for all $j \neq l$, which implies that at most one a_j is nonzero.

Step 5. So far, we have proven that, among other things, $A^{(2)}_{i,j} = 0$ for all i, j. We now go back to equation (4.41), which we can write as

$$\mathbf{f}(\mathbf{x}) = \mathbf{A}^{(1)}\mathbf{h}(\mathbf{x}) + \boldsymbol{\alpha}. \tag{4.51}$$

Both $\mathbf{f}$ and $\mathbf{h}$ are differentiable, and according to assumption 4, $J_{\mathbf{f}}$ has rank n (it is full rank and $n \leq d$). Thus, by differentiating the last equation, we conclude that $\mathbf{A}^{(1)}$ has rank n, and is thus invertible.

Conclusion. We've shown that $f_i(\mathbf{x}) = a_j h_j(\mathbf{x}) + \alpha_i$, where $a_j = A^{(1)}_{ij}$. This is valid for all $i \in [\![1, n]\!]$. Now since $\mathbf{A}^{(1)}$ is invertible, the nonzero entry $A^{(1)}_{ij}$ has to be in a different column for each row, otherwise some rows will be linearly dependent. Thus, there exists a permutation σ of $[\![1, n]\!]$, such that $A^{(1)}_{i\sigma(i)} \neq 0$, and we deduce that

$$f_i(\mathbf{x}) = a_{\sigma(i)} h_{\sigma(i)}(\mathbf{x}) + \alpha_i, \tag{4.52}$$

which concludes the proof.

From the second conclusion of step 3, we have that $\mathbf{d} = \mathbf{a}^2$. Combined with the fact that exactly one element of $\mathbf{a}$ is nonzero such that $\mathbf{A}^{(1)}$ is full rank, this implies that $\mathbf{A}^{(4)}$ is also full rank, which in turn means that $\mathbf{A}$ is full rank. □

4.C.3 Universal approximation capability

Proof of Theorem 4.9. We consider here two cases.

Continuous auxiliary variable. Recall the form of our model:

$$\log p_{\boldsymbol{\theta}}(\mathbf{x}|\mathbf{y}) = -\log Z(\mathbf{y}) - \mathbf{f}(\mathbf{x})^\top \mathbf{g}(\mathbf{y}). \tag{4.53}$$

By parametrizing each of f_i, g_i as neural networks, these functions can approximate continuous function on their respective domains arbitrarily well (Hornik, 1991). According to Lemma 4.19, this implies that any continuous function on $\mathcal{X} \times \mathcal{Y}$ can be approximated arbitrarily well by a term of the form $-\mathbf{f}(\mathbf{x})^\top \mathbf{g}(\mathbf{y})$. In other words, any continuous function can be approximated by $\log p_{\boldsymbol{\theta}}(\mathbf{x}|\mathbf{y}) + \log Z(\mathbf{y})$ for some $\boldsymbol{\theta}$, where $Z(\mathbf{y})$ captures the difference in scale between the function in question and the normalized density $p_{\boldsymbol{\theta}}(\mathbf{x}|\mathbf{y})$. We apply this result to $\log p(\mathbf{x}|\mathbf{y})$: for any $\varepsilon > 0$, there exists $(\boldsymbol{\theta}, n) \in \Theta \times \mathbb{N}$ such that:

$$\sup_{\mathbf{x},\mathbf{y}} \left| \log p(\mathbf{x}|\mathbf{y}) + \sum_{i=1}^{n} f_i(\mathbf{x};\boldsymbol{\theta}) g_i(\mathbf{y};\boldsymbol{\theta}) \right| < \varepsilon. \tag{4.54}$$

Since $p(\mathbf{x}|\mathbf{y}) > 0$ a.s. on $\mathcal{X} \times \mathcal{Y}$, $\log p(\mathbf{x}|\mathbf{y})$ is finite and bounded. So is the term $-\sum_{i=1}^{n} f_i(\mathbf{x};\boldsymbol{\theta}) g_i(\mathbf{y};\boldsymbol{\theta})$. We can then use the fact that exp is Lipschitz on compacts to conclude for $p(\mathbf{x}|\mathbf{y})$, to conclude that:

$$\sup_{\mathbf{x},\mathbf{y}} |p(\mathbf{x}|\mathbf{y}) - p_{\boldsymbol{\theta}}(\mathbf{x}|\mathbf{y})| < K\varepsilon, \tag{4.55}$$

where K is the Lipschitz constant of exp, which concludes the proof.

Discrete auxiliary variable. If $\mathbf{y}$ is discrete and $\mathcal{Y}$ is compact, then $\mathbf{y}$ only takes finitely many values. In this case, we do not need Lemma 4.19 for the proof. $\mathbf{g}(\mathbf{y})$ can simply be a lookup table, and we learn different approximations for each fixed value of $\mathbf{y}$, since $\mathbf{f}$ has the universal approximation capability, which concludes the proof. □

We introduce few definitions as well as a lemma that will be useful for the proof of Theorem 4.9.

Definition 4.15 (Hausdorff space)**.** *A topological space $\mathcal{X}$ is said to be a* Hausdorff space *if for any pair of disjoint points $(u, v) \in \mathcal{X}^2$, there exists two disjoint open subsets $\mathcal{U}, \mathcal{V} \subseteq \mathcal{X}$ such that $u \in \mathcal{U}$ and $v \in \mathcal{V}$.*

Definition 4.16 (Compact space)**.** *A topological space $\mathcal{X}$ is said to be a* compact space *if any cover of $\mathcal{X}$ has a finite subcover.*

In other words, $\mathcal{X}$ is compact if for every collection C of open subsets of $\mathcal{X}$ such that $\mathcal{X} = \cup_{c \in C} c$, there exist a finite *subset $F \subset C$ such that $\mathcal{X} = \cup_{c \in F} c$.*

In particular, if $\mathcal{X}$ is a subset of an Euclidean space, it is compact if and only if it is closed and bounded.

Definition 4.17 (Unital sub-algebra and point separation)**.** *Let K be a compact Hausdorff space. Consider the Banach algebra $\mathcal{C}(K)$ equipped with the supremum norm $\|f\|_\infty = \sup_{t \in K} |f(t)|$. Then:*

1. *$\mathcal{A} \in \mathcal{C}(K)$ is a unital sub-algebra if:*

   (i) $1 \subset \mathcal{A}$.

   (ii) for all $f, g \in \mathcal{A}$ and $\alpha, \beta \in \mathbb{R}$, we have $\alpha f + \beta g \in \mathcal{A}$ and $fg \in \mathcal{A}$.

2. *$\mathcal{A} \subset \mathcal{C}(K)$ separates points of K if $\forall s, t \in K$ such that $s \neq t$, $\exists f \in \mathcal{A}$ s.t. $f(s) \neq f(t)$.*

Theorem 4.18 (Stone-Weirstrass, Brosowski and Deutsch (1981))**.** *Let K be a compact Hausdorff space, and $\mathcal{A}$ a unital sub-algebra of $\mathcal{C}(K)$ which separates points of K. Then $\mathcal{A}$ is dense in $\mathcal{C}(K)$.*

Denote by $\mathcal{C}(\mathcal{X})$ (respectively $\mathcal{C}(\mathcal{Y})$ and $\mathcal{C}(\mathcal{X} \times \mathcal{Y})$) the Banach algebra of continuous functions from $\mathcal{X}$ (respectively $\mathcal{Y}$ and $\mathcal{X} \times \mathcal{Y}$) to $\mathbb{R}$. For any subsets of functions $\mathcal{F}_\mathcal{X} \subset \mathcal{C}(\mathcal{X})$ and $\mathcal{F}_\mathcal{Y} \subset \mathcal{C}(\mathcal{Y})$, let $\mathcal{F}_\mathcal{X} \otimes \mathcal{F}_\mathcal{Y} \coloneqq \{\sum_{i=1}^n f_i g_i | n \in \mathbb{N}, f_i \in \mathcal{F}_\mathcal{X}, g_i \in \mathcal{F}_\mathcal{Y}\}$ be the set of *all linear combinations* of products of functions from $\mathcal{F}_\mathcal{X}$ and $\mathcal{F}_\mathcal{Y}$ to $\mathbb{R}$. Universal approximation is expressed in terms of set density: for instance, the set of functions $\mathcal{F}_\mathcal{X}$ have universal approximation of $\mathcal{C}(\mathcal{X})$ if it is dense in it, *i.e.* for any function f in $\mathcal{C}(\mathcal{X})$, we can always find a limit of a sequence of functions f_n of $\mathcal{F}_\mathcal{X}$ that converges to it. We mathematically express density by writing $\overline{\mathcal{F}_\mathcal{X}} = \mathcal{C}(\mathcal{X})$.

We have the following universal density result on the density of a cartesian product of dense sets:

Lemma 4.19 (Universal approximation capability)**.** *Suppose the following:*

(i) $\mathcal{X}$ and $\mathcal{Y}$ are compact Hausdorff spaces.

(ii) $\overline{\mathcal{F}_\mathcal{X}} = \mathcal{C}(\mathcal{X})$ and $\overline{\mathcal{F}_\mathcal{Y}} = \mathcal{C}(\mathcal{Y})$

then $\overline{\mathcal{F}_{\mathcal{X}} \otimes \mathcal{F}_{\mathcal{Y}}} = \mathcal{C}(\mathcal{X} \times \mathcal{Y})$. *All completions here are with respect to the infinity norm.*

Proof. We prove this theorem in two steps:

1. We first prove that $\mathcal{F}_{\mathcal{X}} \otimes \mathcal{F}_{\mathcal{Y}}$ is dense in $\mathcal{C}(\mathcal{X}) \otimes \mathcal{C}(\mathcal{Y})$ using the hypotheses the lemma.

2. we prove that $\mathcal{C}(\mathcal{X}) \otimes \mathcal{C}(\mathcal{Y})$ is dense in $\mathcal{C}(\mathcal{X} \times \mathcal{Y})$ using Theorem 4.18.

Step 1. Let $\varepsilon > 0$. Let $h \in \mathcal{C}(\mathcal{X}) \otimes \mathcal{C}(\mathcal{Y})$. Then there exists $k \in \mathbb{N}$ and functions $f_i \in \mathcal{C}(\mathcal{X})$ and $g_i \in \mathcal{C}(\mathcal{Y})$ such that $h = \sum_{i=1}^{k} f_i g_i$. For each i, since $\mathcal{F}_{\mathcal{Y}}$ dense in $\mathcal{C}(\mathcal{Y})$, there exists $\tilde{g}_i \in \mathcal{F}_{\mathcal{Y}}$ such that $\|g_i - \tilde{g}_i\|_\infty < \frac{\varepsilon}{2k\|f_i\|_\infty}$. From $\mathcal{F}_{\mathcal{X}}$ dense in $\mathcal{C}(\mathcal{X})$, there exists $\tilde{f}_i \in \mathcal{F}_{\mathcal{X}}$ such that $\|f_i - \tilde{f}_i\|_\infty < \frac{\varepsilon}{2k\|\tilde{g}_i\|_\infty}$. We then have

$$\begin{aligned} \|f_i g_i - \tilde{f}_i \tilde{g}_i\|_\infty &= \|f_i g_i - f_i \tilde{g}_i + f_i \tilde{g}_i - \tilde{f}_i \tilde{g}_i\|_\infty && (4.56) \\ &\leq \|f_i\|_\infty \|g_i - \tilde{g}_i\|_\infty + \|\tilde{g}_i\|_\infty \|f_i - \tilde{f}_i\|_\infty && (4.57) \\ &< \frac{\varepsilon}{k}. && (4.58) \end{aligned}$$

Using this, we conclude that

$$\|h - \sum_{i=1}^{k} \tilde{f}_i \tilde{g}_i\|_\infty \leq \sum_{i=1}^{k} \|f_i g_i - \tilde{f}_i \tilde{g}_i\|_\infty < \varepsilon, \tag{4.59}$$

which proves that $\mathcal{F}_{\mathcal{X}} \otimes \mathcal{F}_{\mathcal{Y}}$ is dense in $\mathcal{C}(\mathcal{X}) \otimes \mathcal{C}(\mathcal{Y})$.

Step 2. We will use the Stone-Weirstrass theorem for this step. It is enough to show that:

(i) $\mathcal{X} \times \mathcal{Y}$ is a compact Hausdorff space.

(ii) $\mathcal{C}(\mathcal{X}) \otimes \mathcal{C}(\mathcal{Y}) \subset \mathcal{C}(\mathcal{X} \times \mathcal{Y})$.

(iii) $\mathcal{C}(\mathcal{X}) \otimes \mathcal{C}(\mathcal{Y})$ is a unital sub-algebra of $\mathcal{C}(\mathcal{X} \times \mathcal{Y})$ (see Definition 4.17).

(iv) $\mathcal{C}(\mathcal{X}) \otimes \mathcal{C}(\mathcal{Y})$ separates points in $\mathcal{X} \times \mathcal{Y}$ (see Definition 4.17).

To prove (i), we use the fact that every finite product of compact spaces is a compact space, and every finite product of Hausdorff spaces is a Hausdorff

space. Points (ii) and (iii) are easy to verify. To prove (iv), let $(\mathbf{x}, \mathbf{y})$ and $(\mathbf{x}', \mathbf{y}')$ be distinct points in $\mathcal{X} \times \mathcal{Y}$. Assume that $\mathbf{x} \neq \mathbf{x}'$ (we proceed similarly if $\mathbf{y} \neq \mathbf{y}'$). Define the continuous function $f \in \mathcal{C}(\mathcal{X})$ such that $f(\mathbf{x}) \neq 0$ and $f(\mathbf{x}') = 0$. Then for $g = 1 \in \mathcal{C}(\mathcal{Y})$, we have $f(\mathbf{x})g(\mathbf{y}) = f(\mathbf{x}) \neq 0 = f(\mathbf{x}')g(\mathbf{y}')$.

All the conditions required to use the Stone-Weirestrass Theorem are verified, and we can conclude that $\mathcal{C}(\mathcal{X}) \otimes \mathcal{C}(\mathcal{Y})$ is dense in $\mathcal{C}(\mathcal{X} \times \mathcal{Y})$

Conclusion. Combining the results of steps 1 and 2, we conclude that $\mathcal{F}_{\mathcal{X}} \otimes \mathcal{F}_{\mathcal{Y}}$ is dense in $\mathcal{C}(\mathcal{X} \times \mathcal{Y})$. □

4.D An identifiable architecture

Proof of Proposition 4.10. Let $\mathbf{f}$ be an MLP that satisfies assumptions (A) to (C). Using Lemma 4.20, we conclude that $\mathbf{f}$ satisfies assumptions 1 and 4. Applying a ReLU to the output of $\mathbf{f}$ constraints its image to $\mathbb{R}^n$, which makes it satisfy assumption 3. □

Proof of Proposition 4.11. Let $\mathbf{g}$ be an MLP that satisfies assumptions (A), (B) and (D). Using Lemma 4.21, we conclude that $\mathbf{g}$ satisfies assumptions 2 and 5. □

Lemma 4.20. *Consider an MLP with L layers, where each layer consists of a linear mapping with weight matrix $\mathbf{W}_l \in \mathbb{R}^{d_l \times d_{l-1}}$ and bias $\mathbf{b}_l$, followed by an activation function. Assume*

a. All activation functions are LeakyReLUs.

b. All weight matrices $\mathbf{W}_l$ are full rank.

c. The row dimension of the weight matrices are either monotonically increasing or decreasing: $d_l \geq d_{l+1}, \forall l \in [\![0, L-1]\!]$ or $d_l \leq d_{l+1}, \forall l \in [\![0, L-1]\!]$.

Then the MLP has a full rank Jacobian almost everywhere. If in addition, $d_L \leq d_0$, then the MLP is surjective.

Proof. Denote by $\mathbf{x}$ the input to the MLP, and by $\mathbf{x}^l$ the output of layer l:

$$\mathbf{x}^0 = \mathbf{x}, \tag{4.60}$$

$$\overline{\mathbf{x}}^l = \mathbf{W}_l \mathbf{x}^{l-1} + \mathbf{b}_l, \tag{4.61}$$

$$\mathbf{x}^l = h(\mathbf{W}_l \mathbf{x}^{l-1} + \mathbf{b}_l) = h(\overline{\mathbf{x}}^l), \tag{4.62}$$

$$h(y) = \alpha y \mathbf{1}_{y<0} + y \mathbf{1}_{y>0}, \tag{4.63}$$

with h in equation (4.62) is an activation function applied to each element of its input, and $\alpha \in (0, 1)$.

Denote by $\mathbf{v}^l \in \mathbb{R}^{d_l}$ the vector whose elements are

$$v_k^l = h'(\overline{x}_k^l) = \begin{cases} 1 \text{ if } \overline{x}_k^l > 0 \\ \alpha \text{ if } \overline{x}_k^l < 0 \end{cases}, \tag{4.64}$$

which is undefined if $\overline{x}_k^l = 0$, and by $\mathbf{V}_l = \text{diag}(\mathbf{v}^l)$. Note that $\mathbf{V}_l$ is a function of its input, and thus of $\mathbf{x}$, but we keep this implicit for simplicity. Using these notations, and the fact that h is piece-wise linear, we can write,

$$\mathbf{x}^L = h(\overline{\mathbf{x}}^L) = \mathbf{V}_L \overline{\mathbf{x}}^L = \mathbf{V}_L \mathbf{W}_L \mathbf{x}^{L-1} + \mathbf{V}_L \mathbf{b}_{L-1} = \cdots = \overline{\mathbf{V}}^L \mathbf{x} + \overline{\mathbf{b}}^L, \tag{4.65}$$

where $\overline{\mathbf{V}}^l = \mathbf{V}_l \mathbf{W}_l \mathbf{V}_{l-1} \mathbf{W}_{l-1} \dots \mathbf{V}_1 \mathbf{W}_1$, $\overline{\mathbf{b}}^0 = 0$ and $\overline{\mathbf{b}}^l = \mathbf{V}_l \mathbf{b}_l + \mathbf{V}_l \mathbf{W}_l \overline{\mathbf{b}}^{l-1}$. This is of course only possible if $\overline{x}_k^l \neq 0$ for all $l \in [\![1, L]\!]$ and all $k \in [\![1, d_l]\!]$. As such, define the set

$$\mathcal{N} = \bigcup_{l=1}^{L} \bigcup_{k=1}^{d_l} \left\{ \mathbf{x} \in \mathbb{R}^d | \overline{x}_k^l = 0 \right\} = \bigcup_{l=1}^{L} \bigcup_{k=1}^{d_l} \left\{ \mathbf{x} \in \mathbb{R}^d | (\overline{\mathbf{v}}_k^l)^\top \mathbf{x} + \overline{b}_k^l = 0 \right\}, \tag{4.66}$$

where $\overline{\mathbf{v}}_k^l$ is the k-th row of $\overline{\mathbf{V}}^l$. For each $\mathbf{x} \notin \mathcal{N}$, we have that $\mathbf{V}_l$ is full rank, and, using Lemma 4.23, $\overline{\mathbf{V}}^l$ is also a full rank matrix.

While it is true that $\overline{b}_k^l$ and $\overline{\mathbf{v}}_k^l$ are functions of $\mathbf{x}$, yet they only take a finite number of values. Thus, the set $\left\{ \mathbf{x} \in \mathbb{R}^d | (\overline{\mathbf{v}}_k^l)^\top \mathbf{x} + \overline{b}_k^l = 0 \right\}$ is included in the union over all the values taken by $\overline{b}_k^j$ and $\overline{\mathbf{v}}_k^j$ up to layer l. For each of these values, the set becomes a dot product between a row of $\overline{\mathbf{V}}^j$ which is independent of the input $\mathbf{x}$, and is nonzero because $\overline{\mathbf{V}}^j$ is full rank; such set has measure zero in $\mathbb{R}^d$. Thus, $\mathcal{N}$ is included in a finite union of sets of measure zero, which implies that it also has measure zero.

Now, for all $\mathbf{x} \notin \mathcal{N}$, $\frac{\partial \mathbf{x}^L}{\partial \mathbf{x}}$ exists, and can be computed using the chain rule:

$$\frac{\partial \mathbf{x}^L}{\partial \mathbf{x}} = \prod_{l=L}^{1} \frac{\partial \mathbf{x}^l}{\partial \mathbf{x}^{l-1}} = \prod_{l=L}^{1} \frac{\partial \mathbf{x}^l}{\partial \overline{\mathbf{x}}^l} \frac{\partial \overline{\mathbf{x}}^l}{\partial \mathbf{x}^{l-1}} = \prod_{l=L}^{1} \mathbf{V}_l \mathbf{W}_l = \overline{\mathbf{V}}^L \tag{4.67}$$

which is full rank. Thus, the MLP has a full rank Jacobian almost everywhere.

The surjectivity is easy to prove since h is surjective and so is $\overline{\mathbf{x}}^l$ as a function of $\mathbf{x}^{l-1}$ if $d_{l-1} \geq d_l$ and $\operatorname{rank}(\mathbf{W}_l) = d_l$. □

Lemma 4.21. *Consider an MLP* $\mathbf{g}$ *with* L *layers, where each layer consists of a linear mapping with weight matrix* $\mathbf{W}_l \in \mathbb{R}^{d_l \times d_{l-1}}$ *and bias* $\mathbf{b}_l$*, followed by an activation function. Assume*

a. All activation functions are LeakyReLUs.

b. All weight matrices $\mathbf{W}_l$ *are full rank.*

c. All submatrices of $\mathbf{W}_l$ *of size* $d_l \times d_l$ *are invertible if* $d_l < d_{l+1}$*.*

Then there exist $d_L + 1$ *points* $\mathbf{y}^0, \ldots, \mathbf{y}^{d_L}$ *such that the matrix*

$$\mathbf{R} = \Big(\mathbf{g}(\mathbf{y}^1) - \mathbf{g}(\mathbf{y}^0), \ldots, \mathbf{g}(\mathbf{y}^{d_L}) - \mathbf{g}(\mathbf{y}^0)\Big)$$

is invertible.

Proof. Let $\mathbf{y}^0$ be an arbitrary point in $\mathbb{R}^{d_0}$. Without loss of generality, suppose that $\mathbf{g}(\mathbf{y}^0) = 0$. This is because $\mathbf{y} \mapsto \mathbf{g}(\mathbf{y}) - \mathbf{g}(\mathbf{y}^0)$ is still an MLP that satisfies all the assumptions above. If for any choice of points $\mathbf{y}^1$ to $\mathbf{y}^{d_L}$, the matrix $\mathbf{R}$ defined above isn't invertible, then this means that $\mathbf{g}(\mathbb{R}^{d_0})$ is necessarily included in a subspace of $\mathbb{R}^{d_L}$ of dimension at most $d_L - 1$. In other words, this would imply that the functions $g_1, \ldots, g_{d_L}$ are not linearly independent. However, this is in contradiction with the result of Lemma 4.29, which stipulates that $g_1, \ldots, g_{d_L}$ are linearly independent, provided all weight matrices satisfy the assumptions of the lemma (which are the same as the assumptions made in this proposition).

Thus, we can conclude that there exist $d_L + 1$ points $\mathbf{y}^0, \ldots, \mathbf{y}^{d_L}$ such that the matrix $\mathbf{R} = \Big(\mathbf{g}(\mathbf{y}^1) - \mathbf{g}(\mathbf{y}^0), \ldots, \mathbf{g}(\mathbf{y}^{d_L}) - \mathbf{g}(\mathbf{y}^0)\Big)$ is invertible. □

Lemma 4.22. *Denote by* $\sigma_{min}(\mathbf{A})$ *the smallest singular value of a matrix* $\mathbf{A}$*. Let* $\mathbf{M}$ *be an* $m \times n$ *matrix, and* $\mathbf{N}$ *be an* $n \times p$ *matrix, such that* $m \leq n \leq p$ or $m \geq n \geq p$*. Then* $\sigma_{min}(\mathbf{MN}) \geq \sigma_{min}(\mathbf{M})\sigma_{min}(\mathbf{N})$*.*

Proof. The proof in the case $m \geq n \geq p$ can be found in Arbel et al. (2018, Lemma 10), but we provide a proof here for completeness, and for the other case $m \leq n \leq p$.

Let $\mathbb{R}^n_* \coloneqq \mathbb{R}^n \setminus \{0\}$, and $\lambda_{\min}(\mathbf{A})$ the smallest eigenvalue of $\mathbf{A}$. Recall that for a matrix $\mathbf{A} \in \mathbb{R}^{n\times m}$, with $m \geq n$,

$$\sigma_{\min}(\mathbf{A}) = \sqrt{\lambda_{\min}(\mathbf{A}^\top \mathbf{A})} = \sqrt{\inf_{\mathbf{x}\in\mathbb{R}^n_*} \frac{\mathbf{x}^\top \mathbf{A}^\top \mathbf{A}\mathbf{x}}{\mathbf{x}^\top \mathbf{x}}} = \inf_{\mathbf{x}\in\mathbb{R}^n_*} \frac{\|\mathbf{A}\mathbf{x}\|}{\|\mathbf{x}\|}. \tag{4.68}$$

Thus, if the null space of $\mathbf{N}$ is non trivial, then $\sigma_{\min}(\mathbf{N}) = 0$, and the inequality is satisfied. Otherwise, we have $\mathbf{N}\mathbf{x} \neq 0$, $\forall \mathbf{x} \in \mathbb{R}^n_*$,

$$\begin{aligned}
\sigma_{\min}(\mathbf{M}\mathbf{N}) &= \inf_{\mathbf{x}\in\mathbb{R}^p_*} \frac{\|\mathbf{M}\mathbf{N}\mathbf{x}\|}{\|\mathbf{x}\|} \\
&= \inf_{\mathbf{x}\in\mathbb{R}^p_*} \frac{\|\mathbf{M}\mathbf{N}\mathbf{x}\|\,\|\mathbf{N}\mathbf{x}\|}{\|\mathbf{N}\mathbf{x}\|\,\|\mathbf{x}\|} \\
&\geq \left(\inf_{\mathbf{x}\in\mathbb{R}^p_*} \frac{\|\mathbf{M}\mathbf{N}\mathbf{x}\|}{\|\mathbf{N}\mathbf{x}\|}\right)\left(\inf_{\mathbf{x}\in\mathbb{R}^p_*} \frac{\|\mathbf{N}\mathbf{x}\|}{\|\mathbf{x}\|}\right) \\
&\geq \left(\inf_{\mathbf{x}\in\mathbb{R}^n_*} \frac{\|\mathbf{M}\mathbf{x}\|}{\|\mathbf{x}\|}\right)\left(\inf_{\mathbf{x}\in\mathbb{R}^p_*} \frac{\|\mathbf{N}\mathbf{x}\|}{\|\mathbf{x}\|}\right) \\
&= \sigma_{\min}(\mathbf{M})\sigma_{\min}(\mathbf{N}).
\end{aligned}$$

If, instead, $\mathbf{A} \in \mathbb{R}^{m\times n}$ with $m \leq n$, then

$$\sigma_{\min}(\mathbf{A}) = \sqrt{\lambda_{\min}(\mathbf{A}\mathbf{A}^\top)} = \sqrt{\inf_{\mathbf{x}\in\mathbb{R}^m_*} \frac{\mathbf{x}^\top \mathbf{A}\mathbf{A}^\top \mathbf{x}}{\mathbf{x}^\top \mathbf{x}}} = \inf_{\mathbf{x}\in\mathbb{R}^m_*} \frac{\left\|\mathbf{A}^\top \mathbf{x}\right\|}{\|\mathbf{x}\|}. \tag{4.69}$$

Similarly, if the null space of $\mathbf{M}^\top$ is non trivial, then $\sigma_{\min}(\mathbf{M}^\top) = \sigma_{\min}(\mathbf{M}) = 0$, and the inequality holds. Otherwise, we have $\mathbf{M}^\top\mathbf{x} \neq 0$, $\forall \mathbf{x} \in \mathbb{R}^m_*$,

$$\begin{aligned}
\sigma_{\min}(\mathbf{M}\mathbf{N}) &= \inf_{\mathbf{x}\in\mathbb{R}^m_*} \frac{\left\|\mathbf{N}^\top\mathbf{M}^\top\mathbf{x}\right\|}{\|\mathbf{x}\|} \\
&= \inf_{\mathbf{x}\in\mathbb{R}^m_*} \frac{\left\|\mathbf{N}^\top\mathbf{M}^\top\mathbf{x}\right\|\left\|\mathbf{M}^\top\mathbf{x}\right\|}{\|\mathbf{M}^\top\mathbf{x}\|\,\|\mathbf{x}\|} \\
&\geq \left(\inf_{\mathbf{x}\in\mathbb{R}^m_*} \frac{\left\|\mathbf{N}^\top\mathbf{M}^\top\mathbf{x}\right\|}{\|\mathbf{M}^\top\mathbf{x}\|}\right)\left(\inf_{\mathbf{x}\in\mathbb{R}^m_*} \frac{\left\|\mathbf{M}^\top\mathbf{x}\right\|}{\|\mathbf{x}\|}\right) \\
&\geq \left(\inf_{\mathbf{x}\in\mathbb{R}^n_*} \frac{\left\|\mathbf{N}^\top\mathbf{x}\right\|}{\|\mathbf{x}\|}\right)\left(\inf_{\mathbf{x}\in\mathbb{R}^m_*} \frac{\left\|\mathbf{M}^\top\mathbf{x}\right\|}{\|\mathbf{x}\|}\right) \\
&= \sigma_{\min}(\mathbf{N})\sigma_{\min}(\mathbf{M}),
\end{aligned}$$

which concludes the proof. □

Lemma 4.23. *Consider a finite sequence of matrices $(\mathbf{M}_i)_{1\leq i\leq p}$, with $\mathbf{M}_i \in \mathbb{R}^{n_{i-1}\times n_i}$. If $\mathbf{M}_i$ is full rank for all $i \in [\![1, p]\!]$, and either $n_0 \leq n_1 \leq \ldots \leq n_p$ or $n_0 \geq n_1 \geq \ldots \geq n_p$, then the product $\mathbf{M}_1\mathbf{M}_2 \ldots \mathbf{M}_p$ is also full rank.*

Proof. If two matrices $\mathbf{M}_1$ and $\mathbf{M}_2$ with ordered dimensions are full rank, then $\sigma_{\min}(\mathbf{M}_1) > 0$ and $\sigma_{\min}(\mathbf{M}_2) > 0$. According to Lemma 4.22, this implies that $\sigma_{\min}(\mathbf{M}_1\mathbf{M}_2) > 0$, and that $\mathbf{M}_1\mathbf{M}_2$ is full rank. The proof for $p \geq 3$ is done by induction on p. □

Lemma 4.24. *Let $\mathbf{A}$ be an $n \times n$ invertible matrix. Denote by $\mathbf{a}_n$ the n-th row of $\mathbf{A}$. Then the matrix $\mathbf{B} \in \mathbb{R}^{n+1,n+1}$ such that*

$$\mathbf{B} = \left(\begin{array}{c|c} & \gamma_1 \\ \mathbf{A} & \vdots \\ & \gamma_{n-1} \\ & \lambda \\ \hline \mathbf{a}_n & 1 \end{array}\right) \tag{4.70}$$

is invertible for any choice of $\gamma_1, \ldots, \gamma_{n-1}$, and for $\lambda \neq 1$.

Proof. Denote by $\mathbf{b}_i$ the i-th row of $\mathbf{B}$. Let $\alpha_1, \ldots, \alpha_{n+1}$ such that

$$\sum_{i=1}^{n+1} \alpha_i \mathbf{b}_i = 0. \tag{4.71}$$

Then in particular, by looking at the first n lines of this vectorial equation, we have that $\sum_{i=1}^{n-1} \alpha_i \mathbf{a}_i + (\alpha_n + \alpha_{n+1})\mathbf{a}_n = 0$. Since $\mathbf{A}$ is invertible, its rows are linearly independent, and thus $\alpha_n = -\alpha_{n+1}$ and $\alpha_i = 0$, $\forall i < n$. Plugging this back into equation (4.71), and looking closely at the last equation, we have that $(1 - \lambda)\alpha_n = 0$, and we conclude that $\alpha_{n+1} = \alpha_n = 0$ (because $\lambda \neq 1$), and that $\mathbf{B}$ is invertible. □

Lemma 4.25. *Consider n affine functions $f_i : \mathbf{x} \in \mathbb{R}^d \mapsto \mathbf{a}_i^\top \mathbf{x} + b_i$, such that the matrix $\mathbf{A} \in \mathbb{R}^{n\times d}$ whose rows are the $\mathbf{a}_i$ is full column rank, and all its submatrices of size $d \times d$ are invertible if $d < n$. Then there exist n non-empty regions $\mathcal{H}_1, \ldots, \mathcal{H}_n$ of $\mathbb{R}^d$ defined by the signs of the functions f_i (for instance, $\mathcal{H} = \{\mathbf{x} \in \mathbb{R}^n | \forall i, f_i(\mathbf{x}) > 0\}$) such that the matrix $\mathbf{s}^n \in \mathbb{R}^{n\times n}$ defined as $S^n_{i,j} = \operatorname{sign}_{\mathbf{x}\in\mathcal{H}_i}(f_j(\mathbf{x}))$ is invertible.*

Proof. We will prove this Lemma by induction on n the number of functions f_i. Denote by $V_i = \{\mathbf{x} \in \mathbb{R}^d | f_i(\mathbf{x}) = 0\}$. The sign of f_i changes if we cross the hyperplan V_i.

First, suppose that $n = 2$. By assumption, we now that $\mathbf{a}_1 \not\propto \mathbf{a}_2$, and thus the hyperplans V_1 and V_2 are not parallel and divide $\mathbb{R}^d$ into 4 regions. This implies that the regions $\mathcal{H}_1 = \{\mathbf{x} \in \mathbb{R}^d | \mathbf{a}_1^\top \mathbf{x} + b_1 > 0, \mathbf{a}_2^\top \mathbf{x} + b_2 > 0\}$ and $\mathcal{H}_2 = \{\mathbf{x} \in \mathbb{R}^d | \mathbf{a}_1^\top \mathbf{x} + b_1 > 0, \mathbf{a}_2^\top \mathbf{x} + b_2 < 0\}$ are not empty.

Second, suppose that there exists n regions $\mathcal{H}_1, \dots \mathcal{H}_n$ such that the matrix $\mathbf{s}^n$ is invertible. Consider the affine function $f_{n+1} = \mathbf{a}_{n+1}^\top \mathbf{x} + b_{n+1}$. The hyperplan $V_{n+1} = \{\mathbf{x} \in \mathbb{R}^d | f_{n+1}(\mathbf{x}) = 0\}$ intersects at least one of the regions $\mathcal{H}_1, \dots \mathcal{H}_n$. This is because $(\dots, \mathbf{a}_i, \dots)_{i \in J})$ are linearly independent for any J of size $\min(d, n+1)$ such that $n+1 \in J$, and thus there exists i_0 such that $\mathbf{a}_{n+1} \not\propto \mathbf{a}_{i_0}$. Suppose without loss of generality that this region is $\mathcal{H}_n$. Denote by $\tilde{\mathcal{H}}_n = \{\mathbf{x} \in \mathbb{R}^n | \mathbf{x} \in \mathcal{H}_n, f_{n+1}(\mathbf{x}) < 0\} \subset \mathcal{H}_n$. Now consider the matrix $\tilde{\mathbf{s}}^n$ such that $\tilde{S}^n_{n,j} = \operatorname{sign}_{\mathbf{x} \in \tilde{\mathcal{H}}_n}(f_j(\mathbf{x}))$ and $\tilde{S}^n_{i,j} = S^n_{i,j}$. Because $\tilde{\mathcal{H}}_n \subset \mathcal{H}_n$, we have that $\operatorname{sign}_{\mathbf{x} \in \mathcal{H}_n}(f_j(\mathbf{x})) = \operatorname{sign}_{\mathbf{x} \in \tilde{\mathcal{H}}_n}(f_j(\mathbf{x}))$ and thus $\tilde{\mathbf{s}}^n = \mathbf{s}^n$, which implies that $\tilde{\mathbf{s}}^n$ is also invertible. Now define $\mathcal{H}_{n+1} = \{\mathbf{x} \in \mathbb{R}^n | \mathbf{x} \in \mathcal{H}_n, f_{n+1}(\mathbf{x}) > 0\} \subset \mathcal{H}_n$. Again, the inclusion implies that $\operatorname{sign}_{\mathbf{x} \in \mathcal{H}_n}(f_j(\mathbf{x})) = \operatorname{sign}_{\mathbf{x} \in \tilde{\mathcal{H}}_n}(f_j(\mathbf{x}))$. Finally, consider the regions $\mathcal{H}_1, \dots, \mathcal{H}_{n-1}, \tilde{\mathcal{H}}_n, \mathcal{H}_{n+1}$, and the matrix $\mathbf{s}^{n+1}$ defined on those regions. Then

$$\mathbf{s}^{n+1} = \left(\begin{array}{cc|c} & & u_1 \\ & \mathbf{s}^n & \vdots \\ & & u_{n-1} \\ & & -1 \\ \hline & \mathbf{s}^n_n & 1 \end{array}\right), \tag{4.72}$$

where $u_i = \operatorname{sign}_{\mathbf{x} \in \mathcal{H}_i} f_{n+1}(\mathbf{x})$ and $\mathbf{s}^n_n$ is the n-th line of $\mathbf{s}^n$. According to Lemma 4.24, $\mathbf{s}^{n+1}$ is invertible, which achieves the proof. □

Lemma 4.26. *Let h denote a LeakyReLU activation function with slope $\lambda \in [0, 1)$ (if $\lambda = 0$, then h is simply a ReLU). Consider n piece-wise affine functions $g_i : \mathbf{x} \in \mathbb{R}^d \mapsto h(\mathbf{a}_i^\top \mathbf{x} + b_i)$, such that the matrix $\mathbf{A} \in \mathbb{R}^{n \times d}$ whose rows are the $\mathbf{a}_i$ is full column rank, and all its submatrices of size $d \times d$ are invertible if $d < n$. Then the functions $g_1, \dots, g_n$ are linearly independent, and their generalized slopes (as piece-wise affine functions) are also linearly independent.*

Proof. Let $f_i = \mathbf{a}_i^\top \mathbf{x} + b_i$ such that $g_i = h(f_i) = \mathbf{1}_{f_i \geq 0} f_i + \mathbf{1}_{f_i < 0} \lambda f_i$.

The assumptions of Lemma 4.25 are met for the function $f_1, \dots, f_n$, and we conclude that there exists n regions $\mathcal{H}_1, \dots, \mathcal{H}_n$ such that the matrix

$$\mathbf{s}^n = \left(\operatorname*{sign}_{\mathbf{x} \in \mathcal{H}_i} (f_j(\mathbf{x})) \right)_{i,j} \tag{4.73}$$

is invertible. Define the matrix $\tilde{\mathbf{s}}$ where we replace all entries of $\mathbf{s}^n$ by λ if they are equal to -1. Then $\tilde{\mathbf{s}}$ is invertible (in fact, to see this, consider the proof of the previous lemma with the slightly unconventional choice of sign function $\operatorname{sign}(x) = \lambda$ if $x < 0$).

Now consider $\alpha_1, \dots, \alpha_n$ such that

$$\sum_{i=1}^{n} \alpha_i g_i = 0. \tag{4.74}$$

Let $k \in [\![1, n]\!]$, and evaluate this equation at $\mathbf{x} \in \mathcal{H}_k$. After taking the gradient with respect to $\mathbf{x}$, we get

$$\sum_i (\mathbf{1}_{\mathbf{x} \in \mathcal{H}_k, f_i(\mathbf{x}) \geq 0} + \lambda \mathbf{1}_{\mathbf{x} \in \mathcal{H}_k, f_i(\mathbf{x}) < 0}) \alpha_i \mathbf{a}_i = 0. \tag{4.75}$$

Denote by $\tilde{\mathbf{s}}_k$ the k-th line of the matrix $\tilde{\mathbf{s}}$, and define $\mathbf{e}_l = (\alpha_1 a_{1,l}, \dots, \alpha_n a_{n,l}) \in \mathbb{R}^n$. We can write the l-th line of equation (4.75) as:

$$\tilde{\mathbf{s}}_k^\top \mathbf{e}_l = 0. \tag{4.76}$$

Collating these equations for a fixed l and $k \in [\![1, n]\!]$, we get

$$\tilde{\mathbf{s}} \mathbf{e}_l = 0, \tag{4.77}$$

which implies that $\mathbf{e}_l = 0$ because $\mathbf{s}$ is invertible. In particular, $\alpha_i a_{i,l} = 0$ for all $i \in [\![1, n]\!]$ and $l \in [\![1, d]\!]$. This implies that $\mathbf{A}_J^\top \boldsymbol{\alpha}_J = 0$, where $J \subset [\![1, n]\!]$ of size $\min(n, d)$, $\mathbf{A}_J = (a_{i,l})_{i \in J, l \in [\![1,d]\!]} \in \mathbb{R}^{d \times d}$ is a submatrix of $\mathbf{A}$ and $\boldsymbol{\alpha}_J = (\alpha_i)_{i \in J} \in \mathbb{R}^d$. Since we know, by assumption, that $\mathbf{A}_J$ is invertible for any choice of set of indices J (relevant when $n > d$), we conclude that $\boldsymbol{\alpha} = 0$ and that the functions $g_1, \dots, g_n$ are linearly independent.

Each function g_i is a piece-wise affine function, with a "generalized slope" equal to $\tilde{\mathbf{a}}_i(\mathbf{x}) = (\mathbf{1}_{f_i \geq 0}(\mathbf{x}) + \lambda \mathbf{1}_{f_i < 0}(\mathbf{x})) \mathbf{a}_i$. As a corollary of the independence of $g_1, \dots g_n$, we can conclude that the slopes $\tilde{\mathbf{a}}_1(\mathbf{x}), \dots, \tilde{\mathbf{a}}_n(\mathbf{x})$ are also independent.

□

Lemma 4.27. *Let $\mathbf{f} = (f_1, \dots, f_n)$ be a vector-valued function defined on $\mathbb{R}^d$. We suppose that $f_1, \dots, f_n$ are linearly independent piece-wise affine functions, and that their generalized slopes $\mathbf{a}_1(\mathbf{x}), \dots, \mathbf{a}_n(\mathbf{x})$ are also linearly independent. Consider m piece-wise affine functions $g_i : \mathbf{x} \in \mathbb{R}^d \mapsto \mathbf{c}_i^\top \mathbf{f}(\mathbf{x}) + d_i$, such that the matrix $\mathbf{C} \in \mathbb{R}^{m \times n}$ whose rows are the $\mathbf{c}_i$ is full column rank, and all its submatrices of size $n \times n$ are invertible if $n < m$. Then there exist m non-empty regions $\mathcal{K}_1, \dots, \mathcal{K}_m$ of $\mathbb{R}^d$ defined by the signs of the functions g_i such that the matrix $\mathbf{T}^m \in \mathbb{R}^{m \times m}$ defined as $T^m_{i,j} = \operatorname{sign}_{\mathbf{x} \in \mathcal{K}_i}(g_j(\mathbf{x}))$ is invertible.*

Proof. Denote by $\tilde{\mathbf{c}}_i(\mathbf{x})$ the generalized slope of the piece-wise affine function g_i: $\tilde{\mathbf{c}}_i(\mathbf{x}) = \sum_j c_{i,j} \mathbf{a}_j(\mathbf{x})$. The key is to show than under the assumptions made here, the slopes $(\dots, \tilde{\mathbf{c}}_i(\mathbf{x}), \dots)_{i \in J}$ are linearly independent for any choice of subset $J \subset [\![1, m]\!]$ of size $\min(m, n)$.

If $m > n$, chose a subset $J \in [\![1, m]\!]$ of size n, and let $(\alpha_i)_{i \in J}$ such that $\sum_{i \in J} \alpha_i \tilde{\mathbf{c}}_i(\mathbf{x}) = 0$. By replacing $\tilde{\mathbf{c}}_i$ by its expression, we get:

$$\sum_j (\sum_i \alpha_i c_{i,j}) \mathbf{a}_j(\mathbf{x}) = 0. \tag{4.78}$$

Since $\mathbf{a}_1, \dots, \mathbf{a}_n$ are linearly independent, we conclude that $\sum_{i \in J} \alpha_i c_{i,j} = 0$ for all $j \in [\![1, n]\!]$. This, along with the full rank assumption on $\mathbf{C}$ prove that $(\alpha_i)_{i \in J} = 0$ and that $(\dots, \tilde{\mathbf{c}}_i(\mathbf{x}), \dots)_{i \in J}$ are linearly independent. We can use the same argument if, instead, $m \leq n$, where $J = [\![1, m]\!]$, and conclude.

The rest of the proof follows the same argument of the proof of Lemma 4.25: we proceed by induction on m. For $m = 2$, we know that $\tilde{\mathbf{c}}_1 \not\propto \tilde{\mathbf{c}}_2$, and so the "generalized hyperplans" defined by these two vectors divide $\mathbb{R}^d$ into at least 3 different regions, 2 of which yield a matrix $\mathbf{T}^2$ that is invertible. Then, if the result hold for m, then the hyperplan defined by the generalized slope of the $(m+1)$-th piece-wise affine function g_{m+1} necessarily intersects one of the regions $\mathcal{K}_1, \dots, \mathcal{K}_m$ since for any subset J of size $\min(m+1, n)$ s.t. $(m+1) \in J$, the generalized slopes $(\dots, \tilde{\mathbf{c}}_i(\mathbf{x}), \dots)_{i \in J}$ are linearly independent. The rest is identical to Lemma 4.25. □

Lemma 4.28. *Let h denote a LeakyReLU activation function with slope $\lambda \in [0, 1)$ (if $\lambda = 0$, then h is simply a ReLU), and $\mathbf{f} = (f_1, \dots, f_n)$ be a vector-valued function defined on $\mathbb{R}^d$. We suppose that $f_1, \dots, f_n$ are linearly independent piece-wise affine functions, and that their generalized slopes $\mathbf{a}_1(\mathbf{x}), \dots, \mathbf{a}_n(\mathbf{x})$*

are also linearly independent. Consider m piece-wise affine functions $g_i : \mathbf{x} \in \mathbb{R}^d \mapsto h(\mathbf{c}_i^\top \mathbf{f}(\mathbf{x}) + d_i)$, such that the matrix $\mathbf{C} \in \mathbb{R}^{m \times n}$ whose rows are the $\mathbf{c}_i$ is full column rank, and all its submatrices of size $n \times n$ are invertible if $n < m$. Then the functions $g_1, \dots, g_m$ are linearly independent, and their generalized slopes are also linearly independent.

Proof. Let $\tilde{g}_i = \mathbf{c}_i^\top \mathbf{f} + d_i$ such that $g_i = h(\tilde{g}_i)$. The assumptions of Lemma 4.27 are met for the functions $\tilde{g}_1, \dots, \tilde{g}_m$, and we conclude that there exists m regions $\mathcal{K}_1, \dots, \mathcal{K}_m$ such that $\mathbf{T}^m = \left(\operatorname{sign}_{\mathbf{x} \in \mathcal{K}_i}(\tilde{g}_j(\mathbf{x}))\right)_{i,j}$ is invertible. Let $\tilde{\mathbf{T}}$ the invertible matrix equal to $\mathbf{T}^m$ after substituting -1 for λ.

Now consider $\alpha_1, \dots, \alpha_m$ such that $\sum_{i=1}^m \alpha_i g_i = 0$ After taking the gradient with respect to $\mathbf{x}$, we get

$$\sum_j (\sum_i \alpha_i (\mathbf{1}_{\tilde{g}_i \geq 0}(\mathbf{x}) + \lambda \mathbf{1}_{\tilde{g}_i < 0}(\mathbf{x})) c_{i,j}) \mathbf{a}_j(\mathbf{x}) = 0. \tag{4.79}$$

Since $\mathbf{a}_1, \dots, \mathbf{a}_n$ are independent, we conclude that

$$\sum_i \alpha_i (\mathbf{1}_{\tilde{g}_i \geq 0}(\mathbf{x}) + \lambda \mathbf{1}_{\tilde{g}_i < 0}(\mathbf{x})) c_{i,j}, \tag{4.80}$$

for all $j \in [\![1, m]\!]$. This in turn implies that

$$\sum_i \alpha_i (\mathbf{1}_{\tilde{g}_i \geq 0}(\mathbf{x}) + \lambda \mathbf{1}_{\tilde{g}_i < 0}(\mathbf{x})) \mathbf{c}_i = 0. \tag{4.81}$$

Let $k \in [\![1, m]\!]$, and evaluate the last equation at $\mathbf{x} \in \mathcal{K}_k$:

$$\sum_i (\mathbf{1}_{\mathbf{x} \in \mathcal{H}_k, f_i(\mathbf{x}) \geq 0} + \lambda \mathbf{1}_{\mathbf{x} \in \mathcal{H}_k, f_i(\mathbf{x}) < 0}) \alpha_i \mathbf{c}_i = 0 \tag{4.82}$$

This last equation is similar to equation (4.75), and we can use the same argument used for the proof of Lemma 4.26 here (using $\tilde{\mathbf{T}}$ instead of $\tilde{\mathbf{s}}$) and deduce that $\alpha_i = 0$ for all i.

We conclude that $g_1, \dots, g_m$ are linearly independent, and so are their generalized slopes as a consequence. □

Lemma 4.29. *Let $\mathbf{f}^L = (f_1^L, \dots, f_{d_L}^L)$ be the output of an L-layer MLP (we assume that $L \geq 2$: there is at least one nonlinearity) that satisfies:*

(a.) All activation functions are LeakyReLUs with slope $\lambda \in [0, 1)$ (if $\lambda = 0$, then the activation function is simply a ReLU).

(b.) All weight matrices $\mathbf{W}_l \in \mathbb{R}^{d_{l+1} \times d_l}$ *are full rank, and all submatrices of* $\mathbf{W}_l$ *of size* $d_l \times d_l$ *are invertible if* $d_l < d_{l+1}$.

Then $f_1^L, \dots, f_{d_L}^L$ *are linearly independent. In addition, all the intermediate features* $(f_1^l, \dots, f_{d_l}^l)$ *are also linearly independent.*

Proof. We prove the Lemma by induction on the number of layers $L \geq 2$. If $L = 2$, then by Lemma 4.26, we conclude that $f_1, \dots, f_n$ are independent. If we suppose the result hold for $L \geq 2$, we can use Lemma 4.28 to prove that it also holds for $L + 1$. Finally, since all layers satisfy the same conditions, the conclusion also applies to intermediate layers. □

4.E Latent variable estimation in generative models

Recall the generative model of IMCA: we observe a random variable $\mathbf{x} \in \mathbb{R}^d$ as a result of a nonlinear transformation $\mathbf{h}$ of a latent variable $\mathbf{z} \in \mathbb{R}^d$ whose distribution is conditioned on an auxiliary variable $\mathbf{y}$ that is also observed:

$$\mathbf{z} \sim p(\mathbf{z}|\mathbf{y}), \tag{4.83}$$

$$\mathbf{x} = \mathbf{h}(\mathbf{z}). \tag{4.84}$$

We assume the latent variable in the IMCA model has a density of the form

$$p(\mathbf{z}|\mathbf{y}) = Q(\mathbf{z}) e^{\sum_i \mathbf{T}_i(z_i)^\top \boldsymbol{\lambda}_i(\mathbf{y}) - \Gamma(\mathbf{y})}, \tag{4.85}$$

where Q is not necessarily factorial.

Further, we will suppose that the density $p(\mathbf{z}|\mathbf{y})$ is strongly exponential (Definition 2.7).

If we suppose that only n out of d components of the latent variable are modulated by the auxiliary variable $\mathbf{y}$ (equivalently, if we suppose that the parameters $\boldsymbol{\lambda}_{n+1:d}(\mathbf{y})$ are constant), then we can write its density as

$$p(\mathbf{z}|\mathbf{y}) = Q(\mathbf{z}) e^{\sum_{i=1}^n \mathbf{T}_i(z_i)^\top \boldsymbol{\lambda}_i(\mathbf{y}) - \Gamma(\mathbf{y})}. \tag{4.86}$$

The term $e^{\sum_{i=n+1}^d \mathbf{T}_i(z_i)^\top \boldsymbol{\lambda}_i}$ is absorbed into $Q(\mathbf{z})$. This last expression will be useful for dimensionality reduction.

To estimate the latent variables of the IMCA model, we fit an augmented version of our energy model

$$p_{\boldsymbol{\theta}}(\mathbf{x}|\mathbf{y}) = Z(\mathbf{y};\boldsymbol{\theta})^{-1} \exp\left(-\mathbf{H}(\mathbf{f}_{\boldsymbol{\theta}}(\mathbf{x}))^\top \mathbf{g}_{\boldsymbol{\theta}}(\mathbf{y})\right), \tag{4.87}$$

where $\mathbf{H}(\mathbf{f}(\mathbf{x})) = (\mathbf{H}_1(f_1(\mathbf{x})), \ldots, \mathbf{H}_d(f_d(\mathbf{x})))$, and each $\mathbf{H}_l$ is a (nonlinear) output activation. An example of such map is $\mathbf{H}_l(x) = (x, x^2)$.

In this section, we present the proofs for the estimation of the Independently Modulated Component Analysis by an identifiable energy model. These proofs are based on similar ideas and techniques to previous proofs, but are different enough that we can't forgo them.

4.E.1 Assumptions

We will decompose Theorem 4.13 into two sub-theorems, which will make the proof easier to understand, but also more adaptable into future work. For the sake of clarity, we will separate its assumptions into smaller assumptions, and refer to them when needed in the proofs. Assumptions (ix) and (x) are only needed for the proof of Theorem 4.14.

(i) The observed data follows the IMCA model of equations (4.83) to (4.85).

(ii) The mixing function $\mathbf{h} : \mathbb{R}^d \to \mathbb{R}^d$ in equation (4.84) is invertible.

(iii) The sufficient statistics $\mathbf{T}_i$ in equation (4.85) are differentiable, and the functions $T_{ij} \in \mathbf{T}_i$ are linearly independent on any subset of $\mathcal{X}$ of measure greater than zero. In other words, $p_{\boldsymbol{\theta}}(\mathbf{z}|\mathbf{u})$ is strongly exponential (Definition 2.7).

(iv) There exist $k+1$ distinct points $\mathbf{y}^0, \ldots, \mathbf{y}^k$ such that the matrix

$$\mathbf{L} = (\boldsymbol{\lambda}(\mathbf{y}_1) - \boldsymbol{\lambda}(\mathbf{y}_0), \ldots, \boldsymbol{\lambda}(\mathbf{y}_k) - \boldsymbol{\lambda}(\mathbf{y}_0))$$

of size $k \times k$ is invertible, where $k = \sum_{i=1}^{d} \dim(\mathbf{T}_i)$.

(v) We fit the model (4.87) to the conditional density $p(\mathbf{x}|\mathbf{y})$, where we assume the feature extractor $\mathbf{f}(\mathbf{x})$ to be differentiable, d-dimensional, and the pointwise nonlinearity $\mathbf{H}$ to be differentiable and k-dimensional, and the dimension of its vector-valued components $\mathbf{H}_l$ to be chosen from $(\dim(\mathbf{T}_1), \ldots, \dim(\mathbf{T}_d))$ without replacement.

(vi) The sufficient statistic in equation (4.85) is twice differentiable and $\dim(\mathbf{T}_l) \geq 2$, $\forall l$.

(vii) The mixing function $\mathbf{h}$ is a $\mathcal{D}^2$-diffeomorphisms.

(viii) The feature extractor $\mathbf{f}$ in equation (4.87) is a $\mathcal{D}^2$-diffeomorphism.

(vi') $\dim(\mathbf{T}_l) = 1$ and $\mathbf{T}_l$ is non-monotonic $\forall l$.

(vii') The mixing function $\mathbf{h}$ is a $\mathcal{C}^1$-diffeomorphism.

(viii') The feature extractor $\mathbf{f}$ in equation (4.87) is a $\mathcal{C}^1$-diffeomorphism, and the nonlinearities $\mathbf{H}_l$ have a unique extremum.

(ix) Only $n \leq d$ components of the latent variable are modulated, and its density has the form (4.86).

(x) The feature extractor $\mathbf{f}$ has the form $\mathbf{f}(\mathbf{x}) = (\mathbf{f}_1(\mathbf{x}), \mathbf{f}_2(\mathbf{x}))$ where $\mathbf{f}_1(\mathbf{x}) \in \mathbb{R}^n$, and the auxiliary feature extractor $\mathbf{g}$ has the form $\mathbf{g}(\mathbf{y}) = (\mathbf{g}_1(\mathbf{y}), \mathbf{g}_2)$ where $\mathbf{g}_1(\mathbf{y}) \in \mathbb{R}^n$ and $\mathbf{g}_2$ is constant.

4.E.2 Proofs

Proof of Theorem 4.13. As mentioned above, we decompose Theorem 4.13 into two smaller results, summarized below by Theorems 4.30 and 4.31. The reader should refer to their proofs for the proof of this Theorem. □

Theorem 4.30. *Assume assumptions (i) to (v) hold. Then, after convergence of our model $p_{\boldsymbol{\theta}}(\mathbf{x}|\mathbf{y})$ to the true density $p(\mathbf{x}|\mathbf{y})$, we can recover the latent variables up to an invertible linear transformation and pointwise nonlinearities, i.e.*

$$\mathbf{H}(\mathbf{f}(\mathbf{x})) = \mathbf{A}\mathbf{T}(\mathbf{z}) + \mathbf{b}, \tag{4.88}$$

where $\mathbf{A}$ is an invertible matrix.

Proof. We fit our density model (4.87) to the conditional density $p(\mathbf{x}|\mathbf{y})$, setting the dimension of the feature extractor $\mathbf{f}$ to be equal to d, and the dimensions of the output nonlinearities $\mathbf{H}_l$ chosen from $(\dim(\mathbf{T}_1), \ldots, \dim(\mathbf{T}_d))$, as per assumption (v):

$$Z(\mathbf{y})^{-1} \exp \mathbf{H}(\mathbf{f}(\mathbf{x}))^\top \mathbf{g}(\mathbf{y}) = p(\mathbf{x}|\mathbf{y}), \tag{4.89}$$

by doing the change of variable $\mathbf{x} = \mathbf{h}(\mathbf{z})$, taking the log on both sides, we get:

$$-\log Z(\mathbf{y}) + \mathbf{H}(\mathbf{f}(\mathbf{x}))^\top \mathbf{g}(\mathbf{y}) = \log p(\mathbf{z}|\mathbf{y}) - \log |\det \mathbf{J}_{\mathbf{h}^{-1}}(\mathbf{x})| \tag{4.90}$$

$$\begin{aligned} &= \log Q(\mathbf{h}^{-1}(\mathbf{x})) + \mathbf{T}(\mathbf{z})^\top \boldsymbol{\lambda}(\mathbf{y}) - \Gamma(\mathbf{y}) \\ &\quad - \log |\det \mathbf{J}_{\mathbf{h}^{-1}}(\mathbf{x})| \,. \end{aligned} \tag{4.91}$$

Let $\mathbf{y}_0, \dots, \mathbf{y}_k$ be the points provided by assumption (iv) of the theorem, where $k = \sum_i k_i$, and $k_i = \dim(\mathbf{T}_i)$. Define $\overline{\boldsymbol{\lambda}}(\mathbf{y}) = \boldsymbol{\lambda}(\mathbf{y}) - \boldsymbol{\lambda}(\mathbf{y}_0)$, $\overline{\Gamma}(\mathbf{y}) = \Gamma(\mathbf{y}) - \Gamma(\mathbf{y}_0)$, $\overline{\mathbf{g}}(\mathbf{y}) = \mathbf{g}(\mathbf{y}) - \mathbf{g}(\mathbf{y}_0)$ and $\overline{Z}(\mathbf{y}) = \log Z(\mathbf{y}) - \log Z(\mathbf{y}_0)$. We plug each of those $\mathbf{y}_l$ in equation (4.91) to obtain $k+1$ such equations. We subtract the first equation for $\mathbf{y}_0$ from the remaining k equations to get for $l = 1, \dots, k$

$$-\overline{Z}(\mathbf{y}_l) + \mathbf{H}(\mathbf{f}(\mathbf{x}))^\top \overline{\mathbf{g}}(\mathbf{y}_l) = \mathbf{T}(\mathbf{z})^\top \overline{\boldsymbol{\lambda}}(\mathbf{y}_l) - \overline{\Gamma}(\mathbf{y}_l). \tag{4.92}$$

The **crucial point** here is that the non factorial terms $Q(\mathbf{g}(\mathbf{x}))$ and $\tilde{Q}(\tilde{\mathbf{g}}(\mathbf{x}))$ cancel out when we take these differences. This is what allows us to generalize the identifiability results of nonlinear ICA to the context of IMCA.

Let $\mathbf{L}$ bet the matrix defined in assumption (iv), and $\tilde{\mathbf{L}} \coloneqq (\dots, \overline{\mathbf{g}}(\mathbf{y}_l), \dots)$. Define $\mathbf{b} = (\dots, \overline{Z}(\mathbf{y}_l) - \overline{\Gamma}(\mathbf{y}_l), \dots)$. Expressing equation (4.92) for all points $\mathbf{y}_l$ in matrix form, we get

$$\tilde{\mathbf{L}}^\top \mathbf{H}(\mathbf{f}(\mathbf{x})) = \mathbf{L}^\top \mathbf{T}(\mathbf{z}) + \mathbf{b}. \tag{4.93}$$

By assumption (iv), $\mathbf{L}$ is invertible, and thus we can write

$$\mathbf{T}(\mathbf{z}) = \mathbf{A}\mathbf{H}(\mathbf{f}(\mathbf{x})) + \mathbf{c}, \tag{4.94}$$

where $\mathbf{c} = \mathbf{L}^{-\top}\mathbf{b}$ and $\mathbf{A} = \mathbf{L}^{-\top}\tilde{\mathbf{L}}^\top$.

To prove that $\mathbf{A}$ is invertible, we first take the gradient of equation (4.94) with respect to $\mathbf{z}$. The Jacobian $\mathbf{J}_\mathbf{T}$ of $\mathbf{T}$ is a matrix of size $k \times d$. Its columns are independent because each $\mathbf{T}_i$ is only a function of z_i, and thus the nonzero entries of each column are in different rows. This means that its rank is d (since $k = \sum_{i=1}^d k_i \geq d$). This is not enough to prove that $\mathbf{A}$ is invertible though. For that, we consider the functions $\mathbf{T}_i$ for which $k_i > 1$: for each of these functions, using Lemma 2.17, there exists points $z_i^{(1)}, \dots, z_i^{(k_i)}$ such that $(\mathbf{T}_i'(z_i^{(1)}), \dots, \mathbf{T}_i'(z_i^{(k_i)}))$ are independent. Collate these points into

$k_{\max} \coloneqq \max_i k_i$ vectors $\mathbf{z}^{(j)} \coloneqq (z_1^{(j)}, \dots z_d^{(j)})$, where for each i, $z_i^{(j)} = z_i^{(1)}$ if $j > k_i$, and $z_i^{(1)}$ is a point such that $T_i(z_i^{(1)}) \neq 0$ if $k_i = 1$. We plug these vectors into equation (4.94) after differentiating it, and collate the $dk_{\max}$ equations in vector form:

$$\mathbf{M} = \mathbf{A}\tilde{\mathbf{M}}, \tag{4.95}$$

where $\mathbf{M} \coloneqq (\dots, \mathbf{J_T}(\mathbf{z}^{(j)}), \dots)$ and $\tilde{\mathbf{M}} \coloneqq (\dots, \mathbf{J_{H \circ f \circ h}}(\mathbf{z}^{(j)}), \dots)$. Now the matrix $\mathbf{M}$ is of size $k \times dk_{\max}$, and it has exactly k independent columns by definition of the points $\mathbf{z}^{(j)}$. This means that $\mathbf{M}$ is of rank k, which in turn implies that $\text{rank}(\mathbf{A}) \geq k$. Since $\mathbf{A}$ is a $k \times k$ matrix, we conclude that $\mathbf{A}$ is invertible. □

The theorem above shows a first step in identifiability which holds up to a linear transformation. This is similar to Hyvärinen et al. (2019), but here we allow for dependencies between components. We can further sharpen the result, in line with Khemakhem et al. (2020a) even in this non-independent case as follows:

Theorem 4.31. *Assume assumptions (i) to (v) hold. Further assume that either assumptions (vi) to (viii)* or *assumptions (vi') to (viii') hold. Then equation (4.88) can be reduced to the component level, i.e. for each $i \in [\![1, d]\!]$:*

$$\mathbf{H}_i(f_i(\mathbf{x})) = A_i \mathbf{T}_{\gamma(i)}(z_{\gamma(i)}) + \mathbf{b}_i, \tag{4.96}$$

where γ is a permutation of $[\![1, d]\!]$ such that $\dim(\mathbf{H}_i) = \dim(\mathbf{T}_{\gamma(i)})$ and A_i a square invertible matrix.

Proof. We prove this theorem separately for both sets of assumptions.

Multi-dimensional sufficient statistics: assumptions (vi) and (viii)
We suppose that $k_i \geq 2$, $\forall i$.
The assumptions of Theorem 4.30 hold, and so we have

$$\mathbf{H}(\mathbf{f}(\mathbf{h}(\mathbf{z}))) = \mathbf{A}\mathbf{T}(\mathbf{z}) + \mathbf{c}, \tag{4.97}$$

for an invertible $\mathbf{A} \in \mathbb{R}^{k \times k}$. We will index $\mathbf{A}$ by four indices (i, l, a, b), where $1 \leq i \leq d, 1 \leq l \leq k_i$ refer to the rows and $1 \leq a \leq d, 1 \leq b \leq k_a$ to the columns.

Let $\mathbf{y} = \mathbf{f} \circ \mathbf{h}(\mathbf{z})$. Since both $\mathbf{f}$ and $\mathbf{h}$ are $\mathcal{D}^2$-diffeomorphisms assumptions (vii) and (viii)), we can invert this relation and write $\mathbf{z} = \mathbf{v}(\mathbf{y})$. We introduce the notations $v_i^s(\mathbf{y}) \coloneqq \frac{\partial v_i}{\partial y_s}(\mathbf{y})$, $v_i^{st}(\mathbf{y}) \coloneqq \frac{\partial^2 v_i}{\partial y_s \partial y_t}(\mathbf{y})$, $T'_{a,b}(z) = \frac{\mathrm{d}T_{a,b}}{\mathrm{d}z}(z)$, $T''_{a,b}(z) = \frac{\mathrm{d}^2 T_{a,b}}{\mathrm{d}z}(z)$ and $H'_{a,b}(y) = \frac{\mathrm{d}H_{a,b}}{\mathrm{d}y}(y)$. Each line of equation (4.97) can be written as

$$H_{i,l}(y_i) = \sum_{a=1}^{d} \sum_{b=1}^{k_i} A_{i,l,a,b} T_{a,b}(v_a(\mathbf{y})) + c_{a,b}, \tag{4.98}$$

for $i \leq d$, $l \leq k_i$. The first step is to show that $v_i(\mathbf{y})$ is a function of only one y_{j_i}, for all $i \leq d$. by differentiating equation (4.98) with respect to y_s, $s \leq d$:

$$\delta_{is} H'_{i,l}(y_i) = \sum_{a=1}^{d} \sum_{b=1}^{k_i} A_{i,l,a,b} T'_{a,b}(v_a(\mathbf{y})) v_a^s(\mathbf{y}), \tag{4.99}$$

and by differentiating equation (4.99) with respect to $y_t, s < t \leq d$:

$$0 = \sum_{a,b} A_{i,l,a,b} \left(T'_{a,b}(v_a(\mathbf{y})) v_a^{s,t}(\mathbf{y}) + T''_{a,b}(v_a(\mathbf{y})) v_a^s(\mathbf{y}) v_a^t(\mathbf{y}) \right). \tag{4.100}$$

This equation is valid for all pairs $(s,t), t > s$. Define

$$\mathbf{B}_a(\mathbf{y}) \coloneqq \left(v_a^{1,2}(\mathbf{y}), \dots, v_a^{d-1,d}(\mathbf{y}) \right) \in \mathbb{R}^{\frac{d(d-1)}{2}}, \tag{4.101}$$

$$\mathbf{C}_a(\mathbf{y}) \coloneqq \left(v_a^1(\mathbf{y}) v_a^2(\mathbf{y}), \dots, v_a^{d-1}(\mathbf{y}) v_a^d(\mathbf{y}) \right) \in \mathbb{R}^{\frac{d(d-1)}{2}}, \tag{4.102}$$

$$\mathbf{M}(\mathbf{y}) \coloneqq (\mathbf{B}_1(\mathbf{y}), \mathbf{C}_1(\mathbf{y}), \dots, \mathbf{B}_d(\mathbf{y}), \mathbf{C}_d(\mathbf{y})) \in \mathbb{R}^{\frac{d(d+1)}{2} \times 2d}, \tag{4.103}$$

$$\mathbf{e}^{(a,b)} \coloneqq (0, \dots, 0, T'_{a,b}, T''_{a,b}, 0, \dots, 0) \in \mathbb{R}^{2d}, \tag{4.104}$$

$$\overline{\mathbf{e}}(\mathbf{y}) \coloneqq (\mathbf{e}^{(1,1)}(y_1), \dots, \mathbf{e}^{(1,k_1)}(y_1), \dots, \mathbf{e}^{(d,1)}(y_d), \dots, \mathbf{e}^{(d,k_d)}(y_d)) \in \mathbb{R}^{2d \times k}, \tag{4.105}$$

such that the nonzero entries of $e^{(\mathbf{a},\mathbf{b})}$ in equation (4.104) are at indices $(2a, 2a+1)$. Then by grouping equation (4.100) for all valid pairs (s,t) and pairs (i,l) and writing it in matrix form, we get

$$\mathbf{M}(\mathbf{y}) \overline{\mathbf{e}}(\mathbf{y}) \mathbf{A} = 0. \tag{4.106}$$

Now by Lemma 2.20, we know that $\overline{\mathbf{e}}(\mathbf{y})$ has rank $2d$ almost surely on $\mathcal{Z}$. Since $\mathbf{A}$ is invertible, it is full rank, and thus $\operatorname{rank}(\overline{\mathbf{e}}(\mathbf{y})\mathbf{A}) = 2d$ almost surely on $\mathcal{Z}$. It suffices then to multiply by its pseudo-inverse from the right to get

$$\mathbf{M}(\mathbf{y}) = 0. \tag{4.107}$$

In particular, $C_a(\mathbf{y}) = 0$ for all $1 \leq a \leq d$. This means that the Jacobian of $\mathbf{v}$ at each $\mathbf{y}$ has at most one nonzero entry in each row. By invertibility and continuity of $J_{\mathbf{v}}$, we deduce that the location of the nonzero entries are fixed and do not change as a function of $\mathbf{y}$. We deduce that there exists a permutation σ of $[\![1, d]\!]$ such that each of the $v_i(\mathbf{y}) = v_i(y_{\sigma(i)})$, and the same would apply to $\mathbf{v}^{-1}$. Without any loss of generality, we assume that σ is the identity.

Now let $\overline{\mathbf{H}}(\mathbf{z}) = \mathbf{H} \circ \mathbf{v}^{-1}(\mathbf{y}) - \mathbf{c}$. This function is a pointwise function because $\mathbf{H}$ and $\mathbf{v}^{-1}$ are such functions. Plugging this back into equation (4.97) yields

$$\overline{\mathbf{H}}(\mathbf{z}) = \mathbf{A}\mathbf{T}(\mathbf{z}). \tag{4.108}$$

The last equation is valid for every component:

$$\overline{H}_{i,l}(z_i) = \sum_{a,b} A_{i,l,a,b} T_{a,b}(z_a). \tag{4.109}$$

By differentiating both sides with respect to z_s where $s \neq i$ we get

$$0 = \sum_{b} A_{i,l,s,b} T'_{s,b}(z_s). \tag{4.110}$$

By Lemma 1.7, we get $A_{i,l,s,b} = 0$ for all $1 \leq b \leq k$. Since equation (4.110) is valid for all l and all $s \neq i$, we deduce that the matrix $\mathbf{A}$ has a block diagonal form:

$$\mathbf{A} = \begin{pmatrix} \mathbf{A}_1 & & \\ & \ddots & \\ & & \mathbf{A}_n \end{pmatrix}, \tag{4.111}$$

which achieves the proof.

One-dimensional sufficient statistics: assumptions (vi') to (viii') We now suppose that $k_i = 1$, $\forall i$.

The proof of Khemakhem et al. (2020a, Theorem 3) can be used here, where we define $\mathbf{v} = (\mathbf{f} \circ \mathbf{h})^{-1}$ and $h_{i,a}(y_a) = D_{i,a} H_a(y_a) - D_{i,a} c_a$, where $\mathbf{D} = \mathbf{A}^{-1}$. We can then rewrite equation (4.98) for every component as

$$T_i(v_i(\mathbf{z})) = \sum_{a=1}^{d} h_{i,a}(z_a), \tag{4.112}$$

which is the same as equation (45) of Khemakhem et al. (2020a). All the assumptions required to prove their theorem are met in our case, and the rest of their proof would simply apply here to prove that $\mathbf{A}$ is a permutation matrix. □

In practice, one could hope to have the feature extractor reduce the dimension of the data, as it is usually very large. This has been achieved in nonlinear ICA before (Hyvärinen and Morioka, 2016; Khemakhem et al., 2020a). It turns out that we can also incorporate dimensionality reduction in IMCA and its estimation by ICE-BeeM, under some assumptions.

Proof of Theorem 4.14. The proof of Theorem 4.30 in this case is unchanged. Simply, we update the total dimension of matrix $\mathbf{L}$ here to $k = \sum_{i=1}^{n} \dim(\mathbf{T}_i)$. When we evaluate equation (4.91) on these points $\mathbf{y}_0, \ldots, \mathbf{y}_k$, the constant term $\mathbf{g}_2$ and the non-modulated components cancel out, and we are left with the equation

$$\tilde{\mathbf{L}}^\top \mathbf{H}_{1:n}(\mathbf{f}_1(\mathbf{x})) = \mathbf{L}^\top \mathbf{T}_{1:n}(\mathbf{z}) + \mathbf{b}. \tag{4.113}$$

We then use similar arguments to the proof of Theorem 4.30 to conclude that

$$\mathbf{H}_{1:n}(\mathbf{f}_1(\mathbf{x})) = \mathbf{A}\mathbf{T}_{1:n}(\mathbf{z}) + \mathbf{c}, \tag{4.114}$$

where $\mathbf{A} \in \mathbb{R}^{n \times n}$ a square invertible matrix. At this point, we can make the same conclusion as Theorem 4.30, while reducing the dimension of the latent space.

We now explain how we can extend Theorem 4.31 to the lower dimensional latent space case. Note that we still assume that $\mathbf{f} = (\mathbf{f}_1, \mathbf{f}_2)$ is a diffeomorphism per assumptions (viii) and (viii'). We can then still define $\mathbf{v} = (\mathbf{f} \circ \mathbf{h})^{-1}$.

We consider now two cases like in the proof of Theorem 4.31.

One-dimensional sufficient statistics Let $\mathbf{D} = \mathbf{A}^{-1}$ and define

$$h_{i,a}(y_a) = D_{i,a} H_a(y_a) - D_{i,a} c_a. \tag{4.115}$$

We can still write equation (4.114) like equation (4.112) as

$$T_i(v_i(\mathbf{y})) = \sum_{a=1}^{n} h_{i,a}(y_a), \tag{4.116}$$

for all $i \leq n$. The assumptions required for the proof are still met, despite reducing the dimension from d to n. This interesting fact is also used for the proof of Theorem 4.8 as well, which achieves this part of the proof.

Multi-dimensional sufficient statistics We rewrite equation (4.114) as

$$H_{i,l}(y_i) = \sum_{a=1}^{n} \sum_{b=1}^{k_i} A_{i,l,a,b} T_{a,b}(v_a(\mathbf{y})) + c_{a,b}, \tag{4.117}$$

for all $i \leq n, l \leq k_i$. We proceed similarly to the proof of Theorem 4.31, replacing all mentions of d by n and keeping all differentiations to indices $t, s \leq n$, up to equation (4.107), after which we can conclude that $v_i^s v_i^t = 0$ for all $i \leq n$, and all $s, t \leq n$. This is not enough to conclude that each of the v_i is only function of one y_{j_i}.

For that, we go back to equation (4.117) and differentiate it with respect to y_s, $s > n$:

$$0 = \sum_{a=1}^{d} \sum_{b=1}^{k_i} A_{i,l,a,b} T'_{a,b}(v_a(\mathbf{y})) v_a^s(\mathbf{y}), \tag{4.118}$$

which is valid for all $i \leq n$, $l \leq k_i$. Since $\mathbf{A}$ is invertible, we can conclude that $T'_{a,b}(v_a(\mathbf{y}))v_a^s(\mathbf{y}) = 0$ for all $a \leq n$ and $s > n$. Since we only consider strongly exponential distributions (assumption (iii)), and using Lemma 2.16, we conclude that $T'_{a,b}(v_a(\mathbf{y})) \neq 0$ almost everywhere, and that $v_a^s(\mathbf{y}) = 0$, for all $s > n$. This, in addition to the fact that $v_i^s v_i^t = 0$ for all $i \leq n$, and all $s, t \leq n$ allows us to conclude that the first n components of $\mathbf{v}$ are each only a function of one different y_j because $\mathbf{v}$ is a diffeomorphism and its Jacobian is continuous. Finally, we can use this fact to deduce that $\mathbf{A}$ is a block permutation matrix, which achieves the proof. □

Chapter 5

Causal autoregressive flows

Normalizing flows and causality, two apparently unrelated fields, have recently received considerable attention in the machine learning community. This chapter highlights an intrinsic correspondence between a simple family of autoregressive normalizing flows and identifiable causal models. We exploit the fact that autoregressive flow architectures define an ordering over variables, analogous to a causal ordering, to show that they are well-suited to performing a range of causal inference tasks, ranging from causal discovery to making interventional and counterfactual predictions. First, we show that causal models derived from both affine and additive autoregressive flows with fixed orderings over variables are identifiable, *i.e.* the true direction of causal influence can be recovered. This provides a generalization of the additive noise model well-known in causal discovery. Second, we derive a bivariate measure of causal direction based on likelihood ratios, leveraging the fact that flow models can estimate normalized log-densities of data. Third, we demonstrate that flows naturally allow for direct evaluation of both interventional and counterfactual queries, the latter being possible due to the invertible nature of normalizing flows. Finally, throughout a series of experiments on synthetic and real data, the proposed method is shown to outperform current approaches for causal discovery and make accurate interventional and counterfactual predictions.

This chapter is based on Khemakhem et al. (2021).

5.1 Introduction

Causal models play a fundamental role in modern scientific endeavour (Spirtes et al., 2000; Pearl, 2009a). Many of the questions which drive scientific research are not associational but rather causal in nature. While randomized controlled studies are the gold standard for understanding the underlying causal mechanisms of a system, such experiments are often unethical, too expensive, or technically impossible. In the absence of randomized controlled trials, the framework of *structural equation models* (SEMs) can be used to encapsulate causal knowledge as well as to answer interventional and counterfactual queries (Pearl, 2009b). At a fundamental level, SEMs define a generative model for data based on causal relationships, and contain strictly more information than their corresponding causal graph and law.

The first step in performing causal inference is to determine the underlying causal graph. Whilst this can be achieved in several ways (e.g., randomized study, expert judgement), data-driven approaches using purely observational data, termed *causal discovery*, are often employed. The challenge for causal discovery algorithms is that given a (typically empirical) data distribution one can write many different SEMs that could generate such distribution (Zhang et al., 2015b; Spirtes and Zhang, 2016). In other words, the causal structure is unidentifiable in the absence of any constraints.

Causal discovery algorithms typically take one of two approaches to achieve identifiability. The first approach is to introduce constraints over the family of functions present in the SEM, for example, assuming all causal dependencies are linear, or that disturbances are additive (Shimizu et al., 2006; Hoyer et al., 2009; Shimizu et al., 2011; Peters et al., 2014; Bloebaum et al., 2018; Zheng et al., 2018). While such approaches have been subsequently extended to allow for bijective transformations (Zhang and Hyvärinen, 2009), they often introduce unverifiable assumptions over the true underlying SEM. An alternative approach is to consider unconstrained causal models whilst introducing further assumptions over the data distribution. These methods often introduce non-stationarity constraints on the distribution of latent variables (Peters et al., 2016; Monti et al., 2019) or assume exogenous variables are present (Zhang et al., 2017).

In the present chapter, we consider the first approach, i.e. constraining the

functions defining the causal relationships, and combine it with the framework of *normalizing flows* recently developed in deep learning literature.

Normalizing flows (Papamakarios et al., 2019; Kobyzev et al., 2020) provide a general way of constructing flexible generative models with tractable distributions, where both sampling and density estimation are efficient and exact. Flows model the data as an invertible transformation of some noise variable, whose distribution is often chosen to be simple, and make use of the change of variable formula in order to express the data density. This formula requires the evaluation of the Jacobian determinant of the transformation.

Autoregressive normalizing flows (Kingma et al., 2016; Papamakarios et al., 2017; Huang et al., 2018) purposefully yield a triangular Jacobian matrix, and the Jacobian determinant can be computed in linear time. Importantly for our purposes, the autoregressive structure in such flows is specified by an ordering on the input variables, and each output variable is only a function of the input variables that precede it in the ordering. Different architectures for autoregressive flows have been proposed, ranging from simple additive and affine transformations (Dinh et al., 2014; Dinh et al., 2016), to more complex cubic and neural spline transformations (Durkan et al., 2019b; Durkan et al., 2019a). Flows have been increasingly popular, with applications in density estimation (Dinh et al., 2016; Papamakarios et al., 2017), variational inference (Rezende and Mohamed, 2015; Kingma et al., 2016) and image generation (Kingma and Dhariwal, 2018; Durkan et al., 2019a), to name a few. Active research is conducted to increase the expressivity and flexibility of flow models while maintaining the invertibility and sampling efficiency.

In this chapter, we consider the ordering of variables in an autoregressive flow model from a causal perspective and highlight the similarities between SEMs and autoregressive flows. We show that under some constraints, autoregressive flow models are well suited to performing a variety of causal inference tasks. As a first contribution, we focus on affine normalizing flows and show that they define an identifiable causal model. In the bivariate data case, the relationship between cause x, independent noise n and effect y in this model can be mathematically expressed by $y = f(x) + v(x)n$, where f denotes the nonlinear effect of the cause, and $v > 0$ is a noise modulation function that depends on the cause. This causal model is a new generalization of the well-known additive noise model, and the proof of its identifiability constitutes the

main theoretical result of this manuscript. We then leverage the properties of flows to perform causal discovery and inference in such models. First, we use the fact that flows can efficiently evaluate exact likelihoods to propose a nonlinear measure of causal direction based on likelihood ratios, with ensuing optimality properties. Second, we show that when autoregressive flow models are conditioned upon the correct causal ordering, they can be used to accurately answer interventional and counterfactual queries. Finally, we show that our method performs favourably on a range of experiments, both on synthetic and real data when compared to previous methods.

5.2 Preliminaries

5.2.1 Structural equation models

Suppose we observe d-dimensional random variables $\mathbf{x} = (x_1, \ldots, x_d)$ with joint distribution $\mathbb{P}_{\mathbf{x}}$. A structural equation model (SEM) is here defined as a tuple $\mathcal{S} = (\mathbf{S}, \mathbb{P}_{\mathbf{n}})$ of a collection $\mathbf{S}$ of d structural equations

$$S_j : \quad x_j = f_j(\mathbf{pa}_j, n_j), \quad j = 1, \ldots, d, \tag{5.1}$$

together with a joint distribution $\mathbb{P}_{\mathbf{n}}$ over latent disturbance (noise) variables n_j, which are assumed to be mutually independent. We write $\mathbf{pa}_j$ to denote the parents of the variable x_j. The SEM defines the *observational* distribution of the random vector $\mathbf{x}$: sampling from $\mathbb{P}_{\mathbf{x}}$ is equivalent to sampling from $\mathbb{P}_{\mathbf{n}}$ and propagating the samples through $\mathbf{S}$. The causal graph $\mathcal{G}$, associated with an SEM (5.1), is a graph consisting of one node corresponding to each variable x_j; throughout this chapter, we assume $\mathcal{G}$ is a directed acyclic graph (DAG).

It is well known that for a DAG, there exists a causal ordering (or permutation) π of the nodes, such that $\pi(i) < \pi(j)$ if the variable x_i precedes the variable x_j in the DAG (but such an ordering is not necessarily unique). Thus, given the causal ordering of the associated DAG we may re-write equation (5.1) as

$$x_j = f_j\left(\mathbf{x}_{<\pi(j)}, n_j\right), \quad j = 1, \ldots, d, \tag{5.2}$$

where $\mathbf{x}_{<\pi(j)} = \{x_i : \pi(i) < \pi(j)\}$ denotes all variables before x_j in the causal ordering. Moreover, in the above definition of SEMs, we allow f_j to be any

(possibly nonlinear) function. Zhang et al. (2015a) proved that the causal direction of the general SEM (5.1) is not identifiable without constraints. To this end, the causal discovery community has focused on specific special cases in order to obtain identifiability results as well as provide practical algorithms. In particular, the additive noise model (Hoyer et al., 2009, ANM), which assumes the noise is additive, is of interest to us in the rest of this manuscript, and its SEM has the form

$$x_j = f_j(\mathbf{x}_{<\pi(j)}) + n_j, \quad j = 1, \ldots, d. \tag{5.3}$$

5.2.2 Autoregressive normalizing flows

Normalizing flow models seek to express the log-density of observations $\mathbf{x} \in \mathbb{R}^d$ as an invertible and differentiable transformation $\mathbf{T}$ of latent variables $\mathbf{z} \in \mathbb{R}^d$, which follow a simple (typically factorial) base distribution that has density $p_{\mathbf{z}}(\mathbf{z})$. The generative model implied under such a framework is:

$$\mathbf{z} \sim p_{\mathbf{z}}(\mathbf{z}), \quad \mathbf{x} = \mathbf{T}(\mathbf{z}).$$

This allows for the density of $\mathbf{x}$ to be obtained via a change of variables as follows:

$$p_{\mathbf{x}}(\mathbf{x}) = p_{\mathbf{z}}(\mathbf{T}^{-1}(\mathbf{x})) \left|\det \mathbf{J}_{\mathbf{T}^{-1}}(\mathbf{x})\right|.$$

Typically, $\mathbf{T}$ or $\mathbf{T}^{-1}$ will be implemented with neural networks. Very often, normalizing flow models are obtained by chaining together different transformations $\mathbf{T}_1, \ldots, \mathbf{T}_k$ from the same family to obtain $\mathbf{T} = \mathbf{T}_1 \circ \cdots \circ \mathbf{T}_k$, while remaining invertible and differentiable. The Jacobian determinant of $\mathbf{T}$ can simply be computed from the Jacobian determinants of the sub-transformations $\mathbf{T}_l$: As such, an important consideration is ensuring the Jacobian determinant of each of the sub-transformations to be efficiently calculated.

Autoregressive flows use transformations that are designed precisely to enable simple Jacobian computation by restricting their Jacobian matrices to be lower triangular (Huang et al., 2018). In this case, the transformation $\mathbf{T}$ has the form

$$x_j = \tau_j(z_j, \mathbf{x}_{<\pi(j)}), \tag{5.4}$$

where π is a permutation that specifies an autoregressive structure on $\mathbf{x}$ and the functions τ_j (called *transformers*) are invertible with respect to their first arguments and are parametrized by their second argument.

5.3 Causal autoregressive flow model

The ideas presented in this manuscript highlight the similarities between equations (5.2) and (5.4). In particular, both models explicitly define an ordering over variables and both models assume the latent variables (denoted by $\mathbf{n}$ or $\mathbf{z}$ respectively) follow simple, factorial distributions. Hereafter, we will use $\mathbf{z}$ to denote both latent disturbances in an SEM and latent variables in an autoregressive flow model. Throughout the remainder of this chapter, we will look to build upon these similarities in order to employ autoregressive flow models for causal inference. First, we explicit in Section 5.3.1 the general conditions under which such correspondence is possible. Then, we consider bivariate *affine* flows in Section 5.3.2, and show that they define a causal model that is identifiable, and which generalizes existing models, particularly additive noise models. In Section 5.3.3, we present our measure of causal direction based on the ratio of the likelihoods under two alternative flow models corresponding to different causal orderings. Finally, Section 5.3.4 presents an extension to the multivariate case. The causal model as well as the flow-based likelihood ratio measure of causal direction constitute the **c**ausal **a**uto**re**gressive **f**low (CAREFL) model.

5.3.1 From autoregressive flow models to SEMs

There are some constraints we need to make on how we define autoregressive normalizing flows so that they remain compatible with causal models:

(**I**) **Fixed ordering:** When chaining together different autoregressive transformations $\mathbf{T}_1, \dots, \mathbf{T}_k$ into $\mathbf{T} = \mathbf{T}_1 \circ \cdots \circ \mathbf{T}_k$, the ordering π of the input variables should be the same for all sub-transformations.

(**II**) **Affine/additive transformations:** The transformers τ_j in (5.4) take what is called an *affine* form:

$$\tau_j(u, \mathbf{v}) = e^{s_j(\mathbf{v})}u + t_j(\mathbf{v}), \tag{5.5}$$

where an *additive* transformation is a special case with $s_j = 0$.

Constraint (**I**) ensures that composing transformations maintains the autoregressive structure of the flow, so as to respect the correspondence with

the SEM (5.2). In fact, if all sub-transformations $\mathbf{T}_l$ are autoregressive and follow the same ordering π, then $\mathbf{T}$ is also autoregressive and follows π (see Appendix 5.A.2 for a proof). We emphasize this point here because it is contrary to the common practice of changing the ordering π throughout the flow to make sure all input variables interact with each other (Germain et al., 2015; Dinh et al., 2016; Kingma and Dhariwal, 2018). This practice aims to improve the model's flexibility, which is the subject of our next point.

Constraint (**II**) ensures that the flow model is not too flexible, and in particular, cannot approximate any density. In fact, the causal ordering of autoregressive flows with universal approximation capability is not identifiable. A proof can be found using the theory of nonlinear ICA (Hyvärinen and Pajunen, 1999): we can autoregressively, and *in any order*, transform any random vector into independent components with simple distributions. Such an autoregressive transformation was described in more detail in equation (1.13) of Section 1.2.2.2. In other words, for any two variables x_1 and x_2, we can construct another variable z_2 such that $z_2 \perp\!\!\!\perp x_1$. Such construction is invertible for x_2, meaning that we can write x_2 as a function of (x_1, z_2). Similarly, the same treatment can be done in the reverse order, to construct a variable z_1 that is independent of x_2, such that x_1 is a function of (x_2, z_1). That is, any two variables would be symmetric according to the SEM. This contradicts the definition of identifiability of a causal model, which states that the transformation $\mathbf{T}$ from the noise $\mathbf{z}$ to the observed variable $\mathbf{x}$ has a unique causal ordering. Since our primary goal in this chapter is to use flows for causal discovery and inference, we shall not use recent methods which emphasize improving the flexibility and expressivity of flow models for density estimation and generative modelling (Durkan et al., 2019b; Durkan et al., 2019a; Müller et al., 2019). Fortunately, flows based on *additive* and *affine* transformations, as defined above (based on the work of Dinh et al. (2016)), are not universal density approximators (see Appendix 5.A.3 for a proof).

Finally, note that constraints (**I**) and (**II**) only limit the expressivity of flows as universal *density* approximators. In contrast, the coefficients s_j and t_j of the affine transformer (5.5), when parametrized as neural networks, can be universal *function* approximators. This property of universal approximation of the functional relationships is desirable in practice, and is preserved when stacking normalizing flows (see Appendix 5.A.4 for a proof).

5.3.2 Model definition and identifiability

Suppose we observe bivariate data $\mathbf{x} = (x_1, x_2) \in \mathbb{R}^2$. Underlying the data, there is a causal ordering described by a permutation π of the set $\{1, 2\}$, where $\pi = (1, 2)$ if $x_1 \to x_2$ and $\pi = (2, 1)$ otherwise.

As per constraints **(I)** and **(II)**, let $\mathbf{T}_1, \ldots, \mathbf{T}_k$ be $k \geq 1$ *affine* autoregressive transformations—*i.e.* of the form (5.4) where the transformers τ_j are affine functions (5.5)—with ordering π, and let $\mathbf{T} = \mathbf{T}_1 \circ \cdots \circ \mathbf{T}_k$. Then $\mathbf{T}$ is also an affine transformation (see Appendix 5.A.2 for a proof). As mentioned earlier, such *composability* is a central and well-known property of affine flows: the ordering stays the same, and the composition is still an affine flow.

The flow $\mathbf{T}$ defines the following SEM on the observations $\mathbf{x}$:

$$x_j = e^{s_j(x_{<\pi(j)})} z_j + t_j(x_{<\pi(j)}), \quad j = 1, 2, \tag{5.6}$$

where z_1, z_2 are statistically independent latent noise variables, and $s_j(x_{<\pi(j)})$ and $t_j(x_{<\pi(j)})$ are defined constant (with respect to $\mathbf{x}$) for $\pi(j) = 1$. Equation (5.6) defines our proposed causal model where the noise is not merely added to some function of the cause (as typical in existing models) but also modulated by the cause.

As a special case, if the transformations $\mathbf{T}_l$, $l = 1, \ldots, k$ are additive (in the sense defined above), then the flow $\mathbf{T}$ is also additive, and $s_1 = s_2 = 0$. In such a special case, equation (5.6) is part of the additive noise model family (5.3), which was proven to be identifiable by Hoyer et al. (2009).

Next, We present a simple and non-technical version of our identifiability result, which states that the more general affine causal model (5.6) is identifiable when the noise variable $\mathbf{z}$ is Gaussian.

Theorem 5.1 (A simplified identifiability result)**.** *Assume* $\mathbf{x} = (x_1, x_2)$ *follows the model described by equation* (5.6)*, with* z_1, z_2 *statistically independent, and the function* t_j *linking cause to effect is nonlinear and invertible. If* z_1 *and* z_2 *are Gaussian, the model is identifiable (i.e.,* π *is uniquely defined). Alternatively (Hoyer et al., 2009), if* $s_1 = s_2 = 0$*, the model is identifiable for any (factorial) distribution of the noise variables* z_1 *and* z_2*.*

More rigorously, Definition 5.2 and Theorem 5.3 below summarize the two scenarios under which the causal model defined by an affine flow is not

identifiable. In particular, if the function t_j in equation (5.6) linking cause to effect is invertible and nonlinear, then none of these scenarios can hold. In addition, the proof in Appendix 5.A.1 only requires one of the noise variables to be Gaussian.

Definition 5.2 (Log-mix-rational-log distributions)**.** *Consider the family of scalars* $(\alpha, \gamma, \delta, \beta, \alpha_0, \beta_0, \gamma_0, \delta_0) \in \mathbb{R}_{\geq 0} \times \mathbb{R}^2_{>0} \times \mathbb{R}^5$ *such that one of the following conditions holds:*

- $\alpha > 0$, $\alpha_0^2 < \alpha\delta$ *and* $\beta^2 < 4\alpha\gamma$.
- $\alpha = \beta = \alpha_0 = 0$ *and* $\beta_0^2 < \delta$.

We say that a density p_x *of a continuous variable* x *is* log-mix-rational-log *if it has the form*

$$\log p_x(x) = -\frac{1}{2}\delta x^2 + \delta_0 x + \frac{1}{2}\frac{(\alpha_0 x^2 + \beta_0 x + \gamma_0)^2}{\alpha x^2 + \beta x + \gamma} - \frac{1}{2}\log(\alpha x^2 + \beta x + \gamma) + const.$$

We say that p_x *is* strictly *log-mix-rational-log if* $\alpha > 0$.

Note that the Gaussian distribution is part of the log-mix-rational-log family, for $\alpha = \beta = \alpha_0 = 0$. If $\alpha \neq 0$, then the log-mix-rational-log family is not part of the exponential family.

Theorem 5.3 (Identifiability of the affine causal model)**.** *Assume the data follows the model*

$$y = f(x) + v(x)n,$$

where n *is a standardized Gaussian independent of* x, f *and* v *are twice-differentiable scalar functions defined on* $\mathbb{R}$ *and* $v > 0$.
If a backward model exists, i.e. the data also follows the same model in the other direction

$$x = g(y) + w(y)m,$$

where m *is a standardized Gaussian independent of* y *and* $w > 0$, *then one of the following scenarios must hold:*

1. $(v, f) = \left(\frac{1}{Q}, \frac{P}{Q}\right)$ *and* $(w, g) = \left(\frac{1}{Q'}, \frac{P'}{Q'}\right)$ *where* Q, Q' *are polynomials of degree two,* $Q, Q' > 0$, P, P' *are polynomials of degree two or less, and* p_x, p_y *are strictly log-mix-rational-log. In particular,* $\lim_{-\infty} v = \lim_{+\infty} v =$

0^+, $\lim_{-\infty} f = \lim_{+\infty} f < \infty$, $\lim_{-\infty} w = \lim_{+\infty} w = 0^+$, $\lim_{-\infty} g = \lim_{+\infty} g < \infty$ *and* f, v, g, w *are not invertible.*

2. v, w *are constant,* f, g *are linear and* p_x, p_y *are Gaussian densities.*

Note that while both Theorems 5.1 and 5.3 assume that at least one noise variable is Gaussian, we believe that the identifiability result also holds for general noise. We show that empirically in Section 5.5.

5.3.3 Choosing causal direction using likelihood ratio

Next, we use our flow-based framework to develop a concrete method for estimating the causal direction, i.e. π. We follow Hyvärinen and Smith (2013) and pose causal discovery as a statistical testing problem which we solve by likelihood ratio testing. We seek to compare two candidate models which can be seen as corresponding to two hypotheses: $x_1 \to x_2$ against $x_1 \leftarrow x_2$. Likelihood ratios are, in general, an attractive way to deciding between alternative hypotheses (models) because they have been proven to be uniformly most powerful, at least when testing "simple" hypotheses (Neyman and Pearson, 1933). However, in our special case, the framework in fact reduces to simply choosing the causal direction that has a higher likelihood.

Normalizing flows allow for easy and exact evaluation of the likelihoods. If we assume the causal ordering $\pi = (1, 2)$, then the likelihood of an affine autoregressive flow is:

$$\log L_{\pi=(1,2)}(\mathbf{x}) = \\ \log p_{z_1}\left(e^{-s_1}(x_1 - t_1)\right) + \log p_{z_2}\left(e^{-s_2(x_1)}(x_2 - t_2(x_1))\right) - s_1 - s_2(x_1).$$

We propose to fit two affine autoregressive flow models (5.6), each conditioned on a distinct causal order over variables: $\pi = (1, 2)$ or $\pi = (2, 1)$. For each candidate model, we train parameters for each flow via maximum likelihood. In order to avoid overfitting, we look to evaluate log-likelihood for each model over a held out testing dataset. As such, the proposed measure of causal direction is defined as:

$$R = \mathbb{E}\left[\log L_{\pi=(1,2)}(\mathbf{x}_{test}; \mathbf{x}_{train})\right] - \mathbb{E}\left[\log L_{\pi=(2,1)}(\mathbf{x}_{test}; \mathbf{x}_{train})\right], \tag{5.7}$$

where $\mathbb{E}\left[\log L_{\pi=(1,2)}(\mathbf{x}_{test}; \mathbf{x}_{train})\right]$ is the empirical expectation of the estimated log-likelihood evaluated on unseen test data $\mathbf{x}_{test}$. If R is positive, we conclude that x_1 is the causal variable, and if R is negative, we conclude that x_2 is the causal variable.

5.3.4 Extension to multivariate data

We can generalize the likelihood ratio measure developed in Section 5.3.3 to the multivariate case by computing the log-likelihood $\log L_\pi$ for each ordering π, and accept the ordering with the highest log-likelihood as the true causal ordering of the data. This procedure is only feasible for small values of d since the number of permutations of $[\![1, d]\!]$ grows exponentially with d. An alternative approach is to employ the bivariate likelihood ratio (5.7) in conjunction with a traditional constraint-based method such as the PC algorithm (Spirtes et al., 2000), similarly to Zhang and Hyvärinen (2009). The PC algorithm is first used to estimate, up to the Markov equivalence class, the skeleton of the DAG $\mathcal{G}$ that describes the causal structure of the data and orient as many edges as possible. Then, the remaining edges are oriented using the bivariate likelihood ratio measure proposed above.

We can also extend the likelihood ratio measure in a different way: we can identify the causal direction between pairs of multivariate variables. More specifically, consider two random vectors $(\mathbf{x}_1, \mathbf{x}_2) \in \mathbb{R}^{2d}$, and suppose that $\mathbf{x}_1 \to \mathbf{x}_2$. Then they can be described by the following SEM:

$$\begin{aligned}\mathbf{x}_1 &= e^{\mathbf{s}_1} \odot \mathbf{z}_1 + \mathbf{t}_1,\\ \mathbf{x}_2 &= e^{\mathbf{s}_2(\mathbf{x}_1)} \odot \mathbf{z}_2 + \mathbf{t}_2(\mathbf{x}_1),\end{aligned}$$

where $(\mathbf{z}_1, \mathbf{z}_2)$ is the vector of latent noise variables that are supposed independent, $\mathbf{s}_i$ and $\mathbf{t}_i$ are vector-valued instead of scalar-valued, and $\odot$ denotes the element-wise product. The likelihood ratio measure (5.7) can be used straightforwardly here to find the correct causal direction between $\mathbf{x}_1$ and $\mathbf{x}_2$, and is implemented in practice using coupling-layer-based normalizing flows (Dinh et al., 2016). Note that while the identifiability theory was developed for the bivariate case, our experiments in Section 5.5 show that it also holds for this case of two multivariate $\mathbf{x}_i$. This is the first model that can readily perform causal discovery over groups of multivariate variables.

5.4 Causal inference using autoregressive flows

This section demonstrates how normalising flow architectures may be employed to perform both interventional and counterfactual queries. We assume that the true causal ordering over variables has been resolved (e.g., as the result of expert judgement or obtained via the method described in Section 5.3).

5.4.1 Interventions

It is possible to manipulate an SEM $\mathcal{S}$ to create interventional distributions over $\mathbf{x}$. As described in Pearl (2009a), intervention on a given variable x_i defines a new *mutilated* generative model where the structural equation associated with variable x_i is replaced by the interventional value, while keeping the rest of the equations (5.4) fixed. Interventions are very useful in understanding causal relationships. If, under the assumption of faithfulness, intervening on a variable x_i changes the marginal distribution of another variable x_j, then it is likely that x_i has some causal effect on x_j. Conversely, if intervening on x_j does not change the marginal distribution of x_i, then the latter is not a descendant of x_j. We follow Pearl (2009a) and denote by $do(x_i = \alpha)$ the interventions that puts a point mass on x_i.

Autoregressive flow modelling allows us to answer interventional queries easily. After fitting a flow model (5.4) conditioned on the right causal ordering (assumed known) to the data, we change the structural equation for variable x_i from $x_i = \tau_i(z_i, \mathbf{x}_{<\pi(i)})$ to $x_i = \alpha$. This breaks the edges from $x_{<\pi(i)}$ to x_i, and puts a point mass on the latent variable z_i. Thereafter, we can directly draw samples from the distribution $\prod_{j\neq i} p_{z_j}$ for all remaining latent variables $z_{j\neq i}$. Finally, we obtain a sample for $\mathbf{x}^{do(x_i=\alpha)}$ by propagating these samples through the flow, which allows us to compute empirical estimates of the interventional distribution. This avoids having to invert the flow and compute $z_i = \tau_i^{-1}(x_i, \mathbf{x}_{<\pi(i)})$. However, in the case of affine autoregressive flows, τ^{-1} is readily available and can be used to parallelise the above algorithm. In fact, we can compute $z_i = \tau_i^{-1}(x_i, \mathbf{x}_{<\pi(i)})$, sample $\mathbf{z}_{j\neq i}$, then propagate the concatenated $\mathbf{z}$ forward through the flow to obtain $\mathbf{x}^{\mathrm{do}(x_i=\alpha)}$. Note that the value of $\mathbf{x}_{<\pi(i)}$ is required to infer z_i, which will break the parallelism. But

since the same value is used to parametrize τ_i and τ_i^{-1}, any value $\mathbf{v}$ can be used as long as $\tau_i(\tau_i^{-1}(\alpha, \mathbf{v}), \mathbf{v}) = \alpha$. In our implementation, we chose $\mathbf{v} = 0$. The sequential and parallel implementation are summarized by Algorithms 1 and 2 respectively.

Algorithm 1 Generate samples from an interventional distribution (sequential)

Input: interventional variable x_i, intervention value α, number of samples S
for $s = 1$ **to** S **do**
  sample $\mathbf{z}(s)$ from flow base distribution (the value of z_i can be discarded)
  set $x_i(s) = \alpha$
  for $j = \pi^{-1}(1)$ to $\pi^{-1}(d)$; $j \neq i$ **do**
    compute observation $x_j(s) = \tau_j(z_j(s), \mathbf{x}_{<\pi(j)}(s))$
  end for
end for
Return: interventional sample $\mathbf{X} = \{\mathbf{x}(s) : s = 1, \ldots, S\}$

Algorithm 2 Generate samples from an interventional distribution (parallel)

Input: interventional variable x_i, intervention value α, number of samples S
for $s = 1$ **to** S **do**
  sample $\mathbf{z}(s)$ from flow base distribution (the value of z_i can be discarded)
  set $z_i(s) = \tau^{-1}(\alpha, \mathbf{0})$
  compute $\mathbf{x}(s) = \mathbf{T}(\mathbf{z}(s))$
end for
Return: interventional sample $\mathbf{X} = \{\mathbf{x}(s) : s = 1, \ldots, S\}$

5.4.2 Counterfactuals

A counterfactual query seeks to quantify statements of the form: what would the value for variable x_i have been if variable x_j had taken value α, *given that we have observed* $\mathbf{x} = \mathbf{x}^{obs}$? The fundamental difference between an interventional and counterfactual query is that the former seeks to marginalise over latent variables, whereas the latter conditions on them.

Given a set of structural equations and an observation $\mathbf{x}^{obs}$, we follow the notation of Pearl (2009a) and write $x_{i,x_j \leftarrow \alpha}(\mathbf{z})$ to denote the value of x_i under the counterfactual that $x_j \leftarrow \alpha$. As detailed by Pearl (2009b), counterfactual inference involves three steps: *abduction, action* and *prediction*:

1. **Abduction**: given an observation $\mathbf{x}^{obs}$, infer the conditional distribution/values over latent variables $\mathbf{z}^{obs}$. This is non-trivial for most causal models. However, since flow models readily give access to both forward and backward transformation between observations and latent variables (Kingma et al., 2016; Papamakarios et al., 2017; Durkan et al., 2019a), this first step can be readily evaluated as $\mathbf{z}^{obs} = \mathbf{T}^{-1}(\mathbf{x}^{obs})$.

2. **Action**: substitute the values of $\mathbf{z}^{obs}$ with the values based on the counterfactual query, $\mathbf{x}_{x_j \leftarrow \alpha}$. More concretely, for a counterfactual, $\mathbf{x}_{x_j \leftarrow \alpha}$, we replace the structural equations for x_j with $x_j = \alpha$ and adjust the inferred value of latent z_j^{obs} accordingly.

3. **Prediction**: compute the implied distribution over $\mathbf{x}$ by propagating the latent variables $\mathbf{z}^{obs}$ through the structural equation models.

The second and third steps mirror those taken when making interventional predictions: the structural equation for the counterfactual variable is fixed at α, and the structural equations are propagated forward. The only difference here is that the latent samples are drawn from their new distribution: in fact, conditioning on $\mathbf{x} = \mathbf{x}^{obs}$ changes the distribution of the latent variables by putting a point mass on $\mathbf{z} = \mathbf{T}^{-1}(\mathbf{x}^{obs})$. This is summarized in Algorithm 3.

Algorithm 3 Answer a counterfactual query

Input: observed data $\mathbf{x}^{obs}$, counterfactual variable x_j and value α
1. **Abduction**: infer $\mathbf{z}^{obs} = \mathbf{T}^{-1}(\mathbf{x}^{obs})$
2. **Action**: (a) set $z^{obs}_{j,x_j \leftarrow \alpha} = \tau_j^{-1}(\alpha, \mathbf{x}^{obs}_{<\pi(j)})$
   (b) set $z^{obs}_{i,x_j \leftarrow \alpha} = z_i^{obs}$ for $i \neq j$
3. **Prediction**: pass $\mathbf{z}^{obs}_{x_j \leftarrow \alpha}$ forward through the flow $\mathbf{T}$

Return: $\mathbf{x}_{x_j \leftarrow \alpha} = \mathbf{T}(\mathbf{z}^{obs}_{x_j \leftarrow \alpha})$

5.5 Experiments[1]

5.5.1 Causal discovery

We compare the performance of CAREFL on a range of synthetic and real world data, against several alternative methods: the linear likelihood ratio

[1]Code to reproduce the experiments is available at https://github.com/ilkhem/carefl.

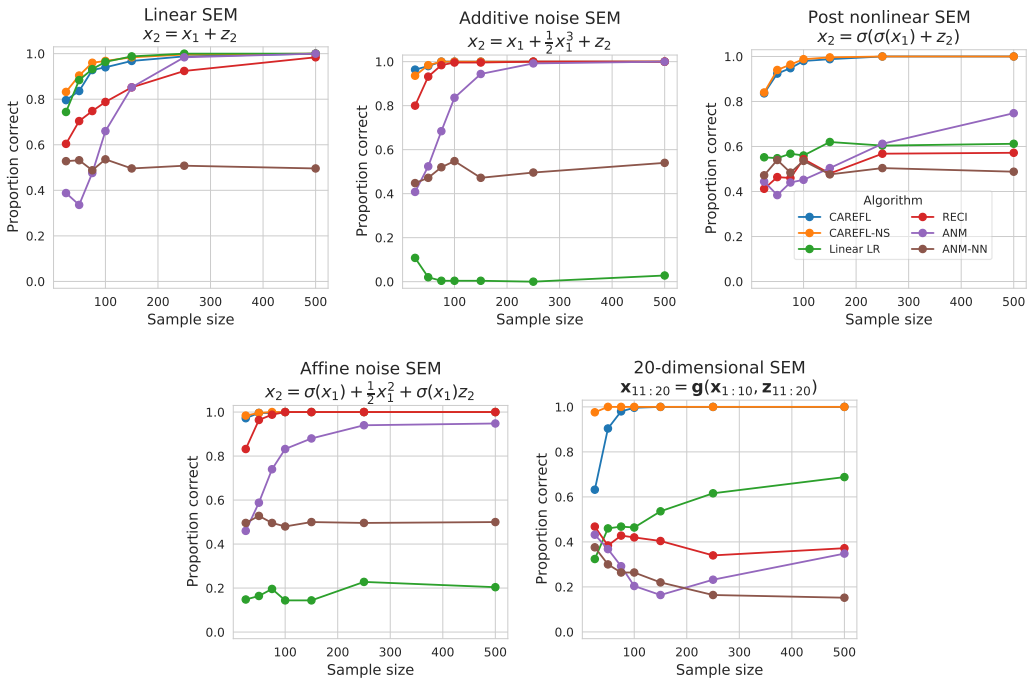


Figure 5.1: Performance on synthetic data generated under distinct SEMs. We note that for all five SEMs CAREFL performs competitively and is able to robustly identify the underlying causal direction.

method of Hyvärinen and Smith (2013), the additive noise model (ANM) method of Hoyer et al. (2009) and Peters et al. (2014), and the Regression Error Causal Inference (RECI) method of Bloebaum et al. (2018). For CAREFL, we considered the more general affine flows, as well as the special case of additive flows (denoted CAREFL-NS, for "non-scaled"), where $s_j = 0$ in (5.6). For ANM, we considered both a Gaussian process and a neural network as the regression class. Experimental details can be found in Appendix 5.B.

5.5.1.1 Synthetic data

We consider a series of synthetic experiments where the underlying causal model is known. Data was generated according to the following SEM:

$$x_1 = z_1,$$
$$x_2 = f(x_1, z_2),$$

where z_1, z_2 follow a standard Laplace distribution. We consider three distinct forms for f: (i) linear, where $f(x_1, z_2) = \alpha x_1 + z_2$; (ii) nonlinear with additive noise, where $f(x_1, z_2) = x_1 + \alpha x_1^3 + z_2$; (iii) nonlinear with modulated noise, where $f(x_1, z_2) = \sigma(x_1) + \frac{1}{2}x_1^2 + \sigma(x_1)z_2$; (iv) nonlinear with nonlinear noise, where $f(x_1, z_2) = \sigma\left(\sigma\left(\alpha x_1\right) + z_2\right)$. We write σ to denote the sigmoid nonlinearity. We also consider a high dimensional SEM:

$$\begin{aligned}\mathbf{x}_1 &= \mathbf{z}_1 \in \mathbb{R}^{10},\\ \mathbf{x}_2 &= \mathbf{g}(\mathbf{x}_1, \mathbf{z}_2) \in \mathbb{R}^{10},\end{aligned}$$

where $\mathbf{z}_1$ and $\mathbf{z}_2$ follow standard Laplace distribution, and for each $i \in [\![1, 10]\!]$, g_i has one of the following forms, picked at random:

(i) a function of all inputs $g_i(\mathbf{x}_1, \mathbf{z}_2) = \sigma(\sigma(\sum_j x_{1,j}) + z_i)$;

(ii) a function of the first half of the input $g_i(\mathbf{x}_1, \mathbf{z}_2) = \sigma(\sigma(\sum_{j\leq 5} x_{1,j}) + z_i)$;

(iii) a function of the second half of the input $g_i(\mathbf{x}_1, \mathbf{z}_2) = \sigma(\sum_{j>5} \sigma(x_{1,j})^{j-5} + z_i)$.

For each distinct class of SEMs, we consider the performance of each algorithm under various distinct sample sizes ranging from $N = 25$ to $N = 500$ samples. Furthermore, each experiment is repeated 250 times. For each repetition, the causal ordering is selected at random. We implemented CAREFL by stacking two affine flows (5.6), where s_j and t_j are feed-forward networks with one hidden layer of dimension 10.

Results are presented in Figure [5.1]. Only CAREFL is able to consistently uncover the true causal direction in all situations. We note that the same architecture and training parameters were employed throughout all experiments, highlighting the fact that the proposed method is agnostic to the nature of the true underlying causal relationship.

We note that while the identifiability results of Theorem 5.1 are premised on Gaussian noise variables, the simulations used a Laplace distribution instead. This proves that the Gaussianity assumption is sufficient but not necessary for identifiability to hold.

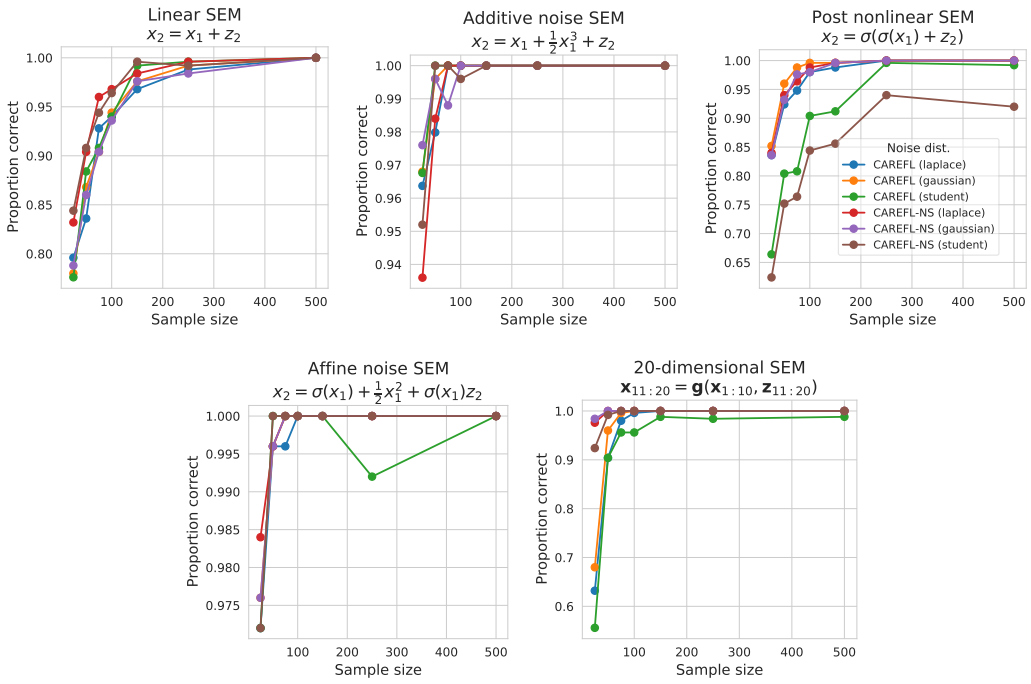


Figure 5.2: Impact of prior mismatch on the performance of CAREFL. The prior of each flow is fixed to a Laplace distribution, while the noise distribution is chosen to be either a Laplace, Student-t or Gaussian distribution.

Robustness to prior misspecification. In the simulations above, the prior distribution of the flow was chosen to be a Laplace distribution, matching the noise distribution. To investigate CAREFL's robustness to prior mismatch, we run additional simulations where the flow prior is still a Laplace distribution, but the noise distribution is changed. The remaining architectural parameters are kept the same as the simulations above. The results are shown in Figure [5.2]. We see that the performance stays the same. We also note that in the following subsection, we will consider real-world datasets where we did not set the underlying (unknown) noise distribution while maintaining better performance when compared to alternative methods.

Exploring flow architectures. As discussed in Section 5.3.1, stacking multiple autoregressive flows on top of each other is equivalent to using a single autoregressive flow with a wide hidden layer. To explore this aspect, we run multiple experiments where each flow is an MLP with one hidden layer and a

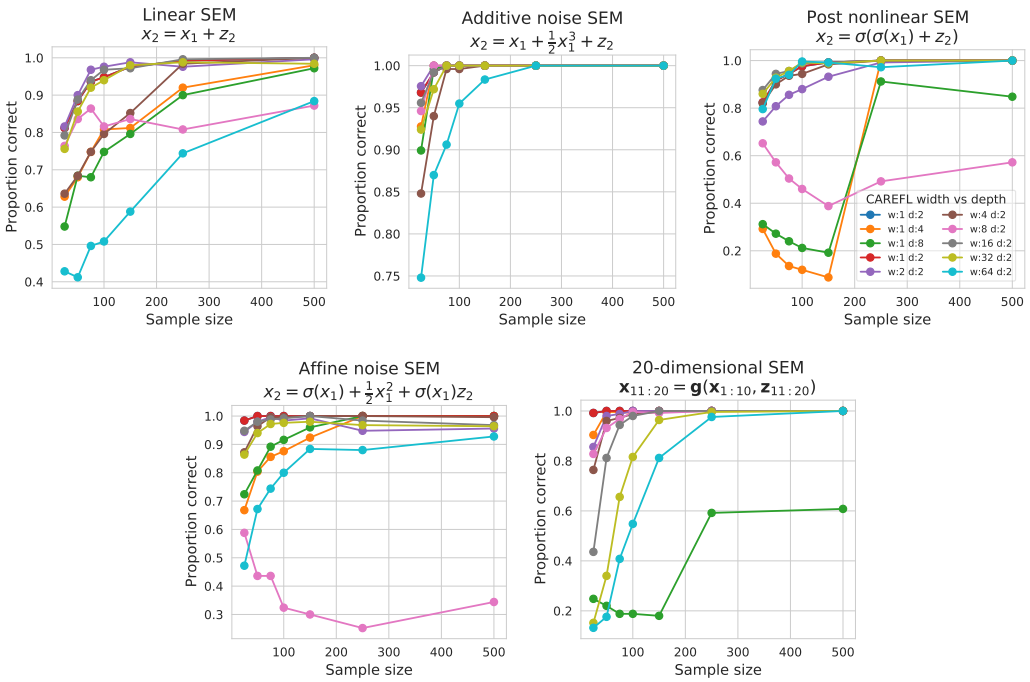


Figure 5.3: Impact of changing the width (w) versus the depth (d) of the normalizing flow in CAREFL.

LeakyReLU activation, in which we vary the width of the hidden layer and the number of stacked flows. We empirically observed that stacking multiple layers in the flows leads to better performance, as reported by Figure [5.3].

5.5.1.2 Real data

Cause effect pairs data. We also consider the performance of the proposed method on the cause-effect pairs benchmark dataset (Mooij et al., 2016). This benchmark consists of 108 distinct bivariate datasets where the objective is to distinguish between cause and effect. For each dataset, two separate autoregressive flow models were trained conditional on $\pi = (1, 2)$ or $\pi = (2, 1)$ and the log-likelihood ratio was evaluated as in equation (5.7) to determine the causal variable. Results are presented in Table [5.1]. We note that the proposed method performs better than alternative algorithms.

Arrow of time on EEG data. Finally, we consider the performance of CAREFL in inferring the arrow of time from open-access electroencephalogram

CAREFL	Linear LR	ANM	RECI
73 %	66%	69 %	69%

Table 5.1: Percentage of correct causal variables identified over 108 pairs from the Cause Effect Pairs benchmark.

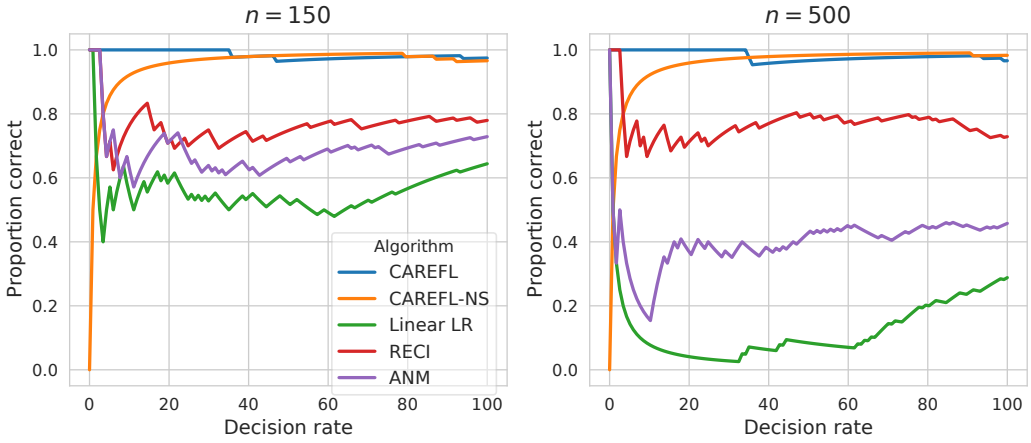


Figure 5.4: Performance on finding the arrow of time of EEG data, as a function of decision rate (percentage of channels — sorted by decreasing confidence — we have to classify).

(EEG) time series (Dornhege et al., 2004). The data consists of 118 EEG channels for one subject. We only consider the first n time points, where $n \in \{150, 500\}$, after which each of the channels is randomly reversed. More details on the preprocessing can be found in Appendix 5.B.2. The goal is to correctly infer whether $x_t \to x_{t+1}$ or $x_{t+1} \to x_t$ for each channel. This is a good test case for causal methods since the true direction is from the past to the future. We report in Figure [5.4] the accuracy as a function of the percentage of channels considered, sorted from highest to lowest confidence (*i.e.* by how high the amplitude of the output of each algorithm is). For average to high confidence, CAREFL is comparable in performance to the baseline methods but performs better in the low confidence regime. We also note that the performance of CAREFL improves by increasing the sample size, which is to be expected from a method based on deep learning.

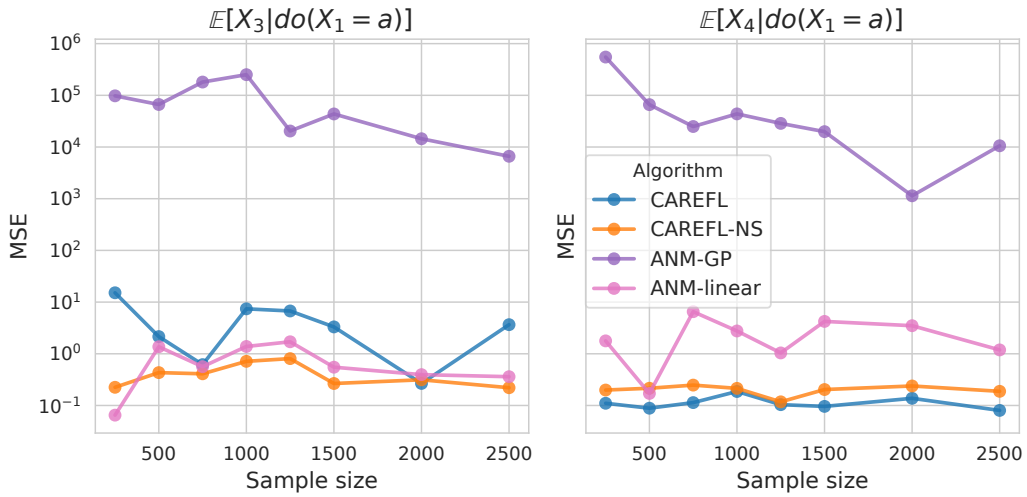


Figure 5.5: Mean square error for interventional predictions on simulated data, generated using equation (5.8). The left and right panels consider linear and nonlinear interventional distributions.

5.5.2 Interventions

To demonstrate that CAREFL can answer interventional queries, we will consider both a synthetic controlled example and real-world fMRI data.

Synthetic data. Consider four-dimensional data generated as

$$\begin{aligned} x_1 &= z_1, & x_3 &= x_1 + c_1 x_2^3 + z_3, \\ x_2 &= z_2, & x_4 &= c_2 x_1^2 - x_2 + z_4, \end{aligned} \tag{5.8}$$

where each z_i is drawn independently from a standard Laplace distribution, and (c_1, c_2) are random coefficients. From the SEM above we can derive the expectations for x_3 and x_4 under an intervention $do(X_1 = \alpha)$ as being α and $c_2\alpha^2$ respectively.

We compare CAREFL against the regression function from an additive noise model (Hoyer et al., 2009, ANM), where the regression is either linear or a Gaussian process. Figure [5.5] visualizes the expected mean squared error between predicted expectations for x_3 and x_4 under the intervention $do(X_1 = \alpha)$ for the proposed method, and the true expectations. We note that CAREFL is able to infer the nature of the true interventional distributions better than the baseline.

Algorithm	Median abs error (std. dev.)
CAREFL	**0.586 (0.048)**
ANM	0.655 (0.057)
Linear SEM	0.643 (0.044)

Table 5.2: Median absolute error for interventional predictions in electrical stimulation fMRI data.

Interventional fMRI data. In order to validate the performance on interventional real-world data, we applied CAREFL to open-access electrical stimulation fMRI (Thompson et al., 2020). Data were collected across 26 patients with medically refractory epilepsy, which required surgically implanting intracranial electrodes in cortical and subcortical locations. FMRI data were then collected during rest as well as while electrodes were being stimulated. Whilst each patient had electrodes implanted in slightly different locations, we identified 16 patients with electrodes in or near the Cingulate Gyrus and studied these patients exclusively. We further restricted ourselves to studying the data from the Cingulate Gyrus (CG) and Heschl's Gyrus (HG), resulting in bivariate time series per patient. Full data preprocessing and preparation is described in Appendix 5.B.3.

We compared CAREFL with both linear and additive noise causal models. Throughout these experiments we assumed the underlying causal structure between regions was known (with CG→HG) and trained each model using the resting-state data. Given the trained model, sessions where the CG was stimulated were treated as interventional sessions, with the task being to predict fMRI activation in HG given CG activity. Whilst the true underlying DAG will certainly be more complex than the simple bivariate structure considered here, these experiments nonetheless serve as a real dataset benchmark through which to compare various causal inference algorithms. The results are provided in Table [5.2], where CAREFL is shown to outperform alternative causal models.

5.5.3 Counterfactuals

We continue with the simple 4 dimensional structural equation model described in equation (5.8). We assume we observe $\mathbf{x}^{obs} = (2.00, 1.50, 0.81, -0.28)$ and consider the counterfactual values under two distinct scenarios: (i) the expected

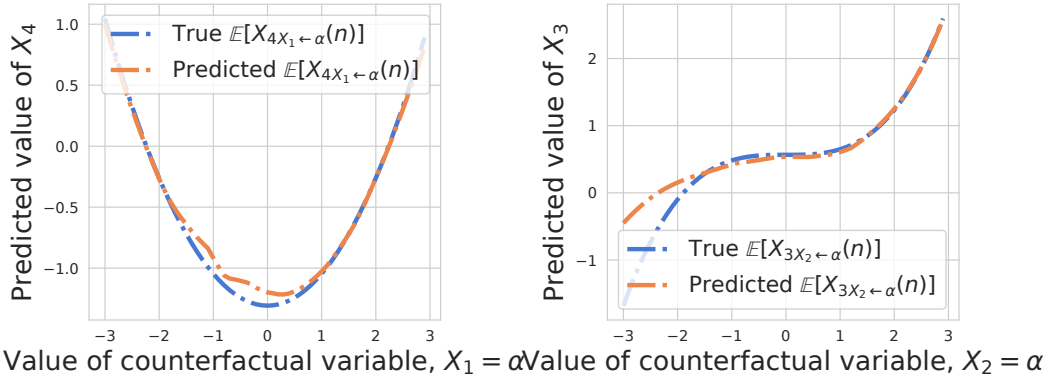


Figure 5.6: Counterfactual predictions for variables x_3 and x_4. Note that flow is able to obtain accurate counterfactual predictions for a range of values of α.

counterfactual value of x_3 if $x_2 = \alpha$ instead of $x_1 = 2$; (ii) the expected counterfactual value of x_4 if $x_1 = \alpha$ instead of $x_1 = 2$. Counterfactual predictions require us to infer the values of latent variables, called *abduction* step by Pearl (2009b). This is non-trivial for most causal models but can be easily achieved with CAREFL due to the invertibility of flow models. Figure [5.6] demonstrates that CAREFL can indeed make accurate counterfactual predictions.

5.6 Related methods

Existing identifiability results on causal models other than additive noise models are limited. To our knowledge, the other notable and identifiable non-additive noise models are the post-nonlinear model (Zhang and Hyvärinen, 2009, PNL) and the non-stationary nonlinear SEM model (Monti et al., 2019, NonSENS). The PNL model assumes that the cause x and the effect y are related through the equation $y = f_2(f_1(x) + n)$, where n is a noise variable independent of x. In contrast to affine flows, the function f_2 is fixed (in the sense of not modulated by the cause x), while being nonlinear as opposed to affine. By applying its inverse f_2^{-1} to y, we actually end up with an additive noise model.

In our model, in stark contrast to the PNL model, it is not possible to apply a fixed (as in not a function of the cause) transformation to the effect to revert back to an additive noise model. This is the main reason why the existing identifiability theory does not cover our causal model (5.6). Theorem 5.3 thus

presents a novel identifiability result in the context of non-additive noise models, and the proposed estimation algorithm benefits from it, as was shown in our experiments.

The NonSENS framework allows for general nonlinear relationships between cause, noise and effect. Assuming access to non-stationary data, it is identifiable even in such a general case by leveraging recent results in the theory of nonlinear ICA (Hyvärinen and Morioka, 2016; Hyvärinen et al., 2019; Khemakhem et al., 2020b; Khemakhem et al., 2020a). In contrast, the proposed model does restrict the nature of nonlinear relationships but places no assumptions of non-stationarity, so our model can be applied in more general scenarios. Our work follows a recent trend of combining flexible generative models (such as autoregressive flows and VAEs) with structural causal models (Louizos et al., 2017; Pawlowski et al., 2020; Wehenkel and Louppe, 2020).

In the context of additive noise models, the estimation methods by Hoyer et al. (2009, ANM) and Bloebaum et al. (2018, RECI) require least-squares regressions in both directions. RECI then compares the magnitudes of the residuals, while ANM depends on independence tests between residuals and causes. Choosing the suitable regression model in both these methods is difficult. As stated by Bloebaum et al. (2018), a very good regression function can reduce the performance of ANM and RECI because it decreases the confidence of the independence tests. We have observed this in our experiments when using neural networks as the regression class, as seen in Figure [5.1]. Importantly, if the additive noise assumption fails to hold, both approaches will fail regardless of the regression class.

CAREFL is specifically leveraging the recent developments in deep learning with the promise of finding computationally efficient methods and improving the statistical efficiency (power) by using likelihood ratios. Furthermore, both ANM and RECI were solely designed for causal discovery, and the invertibility of the system in order to perform interventions and counterfactuals was not discussed. So, it is plausible that our model might be preferable even in the context of additive noise models, in addition to generalizing them.

We note that the likelihood ratio approach by Hyvärinen and Smith (2013) was originally designed for LiNGAM, which is a linear model based on non-Gaussianity (Shimizu et al., 2006). An extension of likelihood ratios to nonlinear ANM was also proposed by Hyvärinen and Smith (2013), together with a

heuristic approximation which roughly amounts to RECI.

5.7 Conclusion

We argue that autoregressive flow models are well-suited to causal inference tasks, ranging from causal discovery to making counterfactual predictions. This is because we can interpret the ordering of variables in an autoregressive flow in the framework of SEMs.

We show that affine flows in particular define a new class of causal models, where the noise is modulated by the cause. For such models, we prove a completely new causal identifiability result which generalizes additive noise models. We show how to efficiently learn causal structure by selecting the ordering with the highest test log-likelihood and thus present a measure of causal direction based on the likelihood ratio for nonlinear SEMs.

Furthermore, by restricting ourselves to autoregressive flow models we are able to easily evaluate interventional queries by fixing the interventional variable whilst sampling from the flow. The invertible property of autoregressive flows further facilitates the evaluation of counterfactual queries.

In experiments on synthetic and real data, our method outperformed alternative methods in causal discovery as well as interventional and counterfactual predictions.

Appendices to Chapter 5

5.A Proofs and additional results

5.A.1 Identifiability of the affine causal model

Recall the form of the SEM that is defined by an autoregressive affine flow:

$$x_j = e^{s_j(x_{<\pi(j)})} z_j + t_j(x_{<\pi(j)}), \quad j = 1, 2, \tag{5.9}$$

where π is a permutation that describes the causal ordering.

The proof for additive flows ($s_1 = s_2 = 0$ in equation (5.9)) and general noise can be found in Hoyer et al. (2009).

Proof of Theorem 5.3. The goal is to prove that the two causal models

$$y = f(x) + v(x)n, \tag{5.10}$$

$$x = g(y) + w(y)m, \tag{5.11}$$

where n and m have a Gaussian distribution, are only undistinguishable in very specific and rare cases.

The log-likelihood of the direction (5.10), denoted by p_1, is given by

$$\log p_1(x, y) = \log p_x(x) - \frac{1}{2}\left(\frac{y - f(x)}{v(x)}\right)^2 - \log v(x) - \frac{1}{2}\log 2\pi, \tag{5.12}$$

and the log-likelihood of (5.11), denoted by p_2, is given by

$$\log p_2(x, y) = \log p_y(y) - \frac{1}{2}\left(\frac{x - g(y)}{w(y)}\right)^2 - \log w(y) - \frac{1}{2}\log 2\pi. \tag{5.13}$$

If the data follows both models, these are equal:

$$\log p_x(x) - \frac{1}{2}\left(\frac{y - f(x)}{v(x)}\right)^2 - \log v(x) = \log p_y(y) - \frac{1}{2}\left(\frac{x - g(y)}{w(y)}\right)^2 - \log w(y). \tag{5.14}$$

Denote $\frac{1}{v(x)}$ by $\overline{v}(x)$ and likewise for w. Now, take the derivative of both sides with respect to x:

$$(\log p_x)'(x) - \overline{v}(x)(y - f(x))(y\overline{v}'(x) - (f\overline{v})'(x)) - (\log v)'(x) = -(x - g(y))\overline{w}^2(y). \tag{5.15}$$

Take the derivative of both sides of this with respect to y:

$$-\overline{v}(x)[2y\overline{v}'(x)-(f\overline{v})'(x)-f(x)\overline{v}'(x)] = -x(\overline{w}^2)'(y)+g'(y)\overline{w}^2(y)+g(y)(\overline{w}^2)'(y). \tag{5.16}$$

Again, take the derivative of both sides with respect to x:

$$-y(\overline{v}^2)''(x) + [\overline{v}((f\overline{v})' + f\overline{v}')]'(x) = -(\overline{w}^2)'(y), \tag{5.17}$$

and once more, take the derivative of both sides of this with respect to y:

$$-(\overline{v}^2)''(x) = -(\overline{w}^2)''(y), \tag{5.18}$$

which is possible only if both sides are constant, which is equivalent to $\overline{v}^2$ and $\overline{w}^2$ being second-order polynomials. In other words,

$$\overline{v}^2(x) = \alpha x^2 + \beta x + \gamma, \quad v^2(x) = \frac{1}{\alpha x^2 + \beta x + \gamma}, \tag{5.19}$$

where the parameters must be such that the $\overline{v}$ is always positive. The same holds for w:

$$\overline{w}^2(y) = \alpha' y^2 + \beta' y + \gamma', \quad w^2(y) = \frac{1}{\alpha' y^2 + \beta' y + \gamma'}. \tag{5.20}$$

Furthermore, equation (5.17) together with the fact that $(\overline{v}^2)''(x) = \text{const}$ implies that

$$[\overline{v}((f\overline{v})' + f\overline{v}')]'(x) = [f'\overline{v}^2 + 2f(\overline{v}^2)')]'(x) = (f\overline{v}^2)''(x) = \text{const} \tag{5.21}$$

or

$$f(x)\overline{v}^2(x) = \alpha_0 x^2 + \beta_0 x + \gamma_0, \tag{5.22}$$

which means that f has the following form:

$$f(x) = \frac{\alpha_0 x^2 + \beta_0 x + \gamma_0}{\alpha x^2 + \beta x + \gamma}. \tag{5.23}$$

The same analysis yields a similar form for g:

$$g(y) = \frac{\alpha_0' y^2 + \beta_0' y + \gamma_0'}{\alpha' x^2 + \beta' x + \gamma'}. \tag{5.24}$$

For $\overline{v}$ to be always positive, the coefficients (α, β, γ) in equation (5.19) must satisfy one of the following conditions:

1. $\alpha > 0$ and $4\alpha\gamma - \beta^2 > 0$.

2. $\alpha = \beta = 0$ and $\gamma > 0$.

Similarly, for $\overline{w}$ to be always positive, the coefficients $(\alpha', \beta', \gamma')$ in equation (5.20) must satisfy one of the following conditions:

1'. $\alpha' > 0$ and $4\alpha'\gamma' - \beta'^2 > 0$.

2'. $\alpha' = \beta' = 0$ and $\gamma' > 0$.

First case: 1. + 1'. In the first case, we conclude that $v = \frac{1}{Q}$ and $f = \frac{P}{Q}$ where Q is a polynomial of degree two, $Q > 0$ and P is a polynomial of degree two or less. Furthermore, $\lim_{-\infty} f = \lim_{+\infty} f = \frac{\alpha_0}{\alpha}$, regardless of whether α_0 is zero or not. This implies that f can't be invertible. Going back to equation (5.14) and plugging these expressions:

$$\begin{aligned}
&\log p_x(x) + \frac{1}{2}\log(\alpha x^2 + \beta x + \gamma) - \frac{1}{2}\frac{(\alpha_0 x^2 + \beta_0 x + \gamma_0)^2}{\alpha x^2 + \beta x + \gamma} \\
&\qquad - \gamma_0' x + \frac{1}{2}\gamma' x^2 + (\alpha_0 x^2 + \beta_0 x)y - \frac{1}{2}(\alpha x^2 + \beta x)y^2 \\
&= \log p_y(y) + \frac{1}{2}\log(\alpha' y^2 + \beta' y + \gamma') - \frac{1}{2}\frac{(\alpha_0' y^2 + \beta_0' y + \gamma_0')^2}{\alpha' y^2 + \beta' y + \gamma'} \\
&\qquad - \gamma_0 y + \frac{1}{2}\gamma y^2 + (\alpha_0' y^2 + \beta_0' y)x - \frac{1}{2}(\alpha' y^2 + \beta' y)x^2, \quad (5.25)
\end{aligned}$$

whcih we can write as

$$A(x) - B(y) - \frac{1}{2}(\alpha - \alpha')x^2y^2 + \left(\alpha_0 - \frac{1}{2}\beta'\right)x^2 y - \left(\alpha_0' - \frac{1}{2}\beta\right)xy^2 + (\beta_0 - \beta_0')xy = 0, \quad (5.26)$$

where

$$\begin{aligned}
A(x) &= \log p_x(x) + \frac{1}{2}\log(\alpha x^2 + \beta x + \gamma) \quad (5.27)\\
&\quad - \frac{1}{2}\frac{(\alpha_0 x^2 + \beta_0 x + \gamma_0)^2}{\alpha x^2 + \beta x + \gamma} - \gamma_0' x + \frac{1}{2}\gamma' x^2, \\
B(y) &= \log p_y(y) + \frac{1}{2}\log(\alpha' y^2 + \beta' y + \gamma') \quad (5.28)\\
&\quad - \frac{1}{2}\frac{(\alpha_0' y^2 + \beta_0' y + \gamma_0')^2}{\alpha' y^2 + \beta' y + \gamma'} - \gamma_0 y + \frac{1}{2}\gamma y^2.
\end{aligned}$$

By first setting $x = 0$ in equation (5.26), we find that $A(x) = B(0)$. Similarly, by now setting $y = 0$, we find that $B(y) = A(0)$. This in particular means that $A(x) - B(y)$ is constant, which, when plugged back in equation (5.26), would imply that all the monomials are zero. Finally, this would in turn imply the following:

$$\alpha = \alpha',\ \alpha_0 = -\frac{1}{2}\beta',\ \alpha_0' = -\frac{1}{2}\beta,\ \beta_0 = \beta_0', \tag{5.29}$$

$$\log p_x(x) = -\frac{1}{2}\gamma' x^2 + \gamma_0' x + \frac{1}{2}\frac{(\alpha_0 x^2 + \beta_0 x + \gamma_0)^2}{\alpha x^2 + \beta x + \gamma} - \frac{1}{2}\log(\alpha x^2 + \beta x + \gamma) + C, \tag{5.30}$$

$$\log p_y(y) = -\frac{1}{2}\gamma y^2 + \gamma_0 y + \frac{1}{2}\frac{(\alpha_0' y^2 + \beta_0' y + \gamma_0')^2}{\alpha' y^2 + \beta' y + \gamma'} - \frac{1}{2}\log(\alpha' y^2 + \beta' y + \gamma') + C. \tag{5.31}$$

Next we need to ensure we have well-defined probability densities. From the above equations, we can check the coefficient of the quadratic term, which dominates at infinity, is $\frac{1}{2\alpha}(\alpha_0^2 - \alpha\gamma')$ for p_x. Requiring this to be negative is exactly the condition for the density family we made in Definition 5.2.

For p_y, we get the dominant quadratic term with the coefficient $\frac{1}{2\alpha'}(\alpha_0'^2 - \alpha'\gamma)$, and with substitutions we find the condition for its negativity as $\beta^2 < 4\alpha\gamma$ which is, again, the same as a condition in the Definition.

Second, the constant C has to be such that the probability density functions integrate to one. In fact, C can be freely chosen, but importantly, it has to be the same for both densities. As a special case, this constraint is obviously fulfilled if the densities are the same, i.e. the parameters with and without prime are the same ($\alpha = \alpha'$ etc.). We shall show below that such parameter values can be found.

In fact, we can see how the parameters of the inverse model are determined from the parameters of the true model as follows. Define

$$\delta := \gamma', \delta_0 := \gamma_0'. \tag{5.32}$$

So we can write the above as

$$\log p_x(x) = -\frac{1}{2}\delta x^2 + \delta_0 x + \frac{1}{2}\frac{(\alpha_0 x^2 + \beta_0 x + \gamma_0)^2}{\alpha x^2 + \beta x + \gamma} - \frac{1}{2}\log(\alpha x^2 + \beta x + \gamma) + C, \tag{5.33}$$

$$\log p_y(y) = -\frac{1}{2}\gamma y^2 + \gamma_0 y + \frac{1}{2}\frac{(-\beta y^2/2 + \beta_0 y + \delta_0)^2}{\alpha y^2 - 2\alpha_0 y + \delta} - \frac{1}{2}\log(\alpha y^2 - 2\alpha_0 y + \delta) + C, \tag{5.34}$$

with

$$C = A(0) = B(0) = \log p_x(0) + \frac{1}{2}\log\gamma - \frac{1}{2}\frac{\gamma_0^2}{\gamma} = \log p_y(0) + \frac{1}{2}\log\gamma' - \frac{1}{2}\frac{\gamma_0'^2}{\gamma'}, \tag{5.35}$$

and where all the parameters defining p_y are now obtained from the parameters defining p_x, f, v (which are here denoted by the parameters without prime for this specific purpose). Likewise, we see that we also get g and w using those same parameters.

Now, we show that in spite of the different constraints, a solution in this family does exist. Let us consider the case where $p_x = p_y$, which would ensure that we can normalize the densities with a common C. This can be achieved by equating corresponding constants above which only requires

$$\beta = -2\alpha_0, \tag{5.36}$$

$$\delta = \gamma, \tag{5.37}$$

$$\delta_0 = \gamma_0, \tag{5.38}$$

which is still perfectly possible, even considering the constraints on the parameters in the Definition, which can be satisfied by simply taking non-negative α, γ, γ', and then fixing α_0 to be small enough in absolute value (which implies the same for β). Thus, a solution for the inverse direction does exist. (But note we didn't prove that it exists for data coming from any p_x, f, v in our family; we have proven unidentifiability only for some parameter values.)

Second case: 2. + 2'. In the second case, we have that v is constant. Going back to equation (5.14), multiplying by -2, plugging the solutions just

obtained:

$$-2\log p_x(x) + \gamma\left(y - \frac{\alpha_0}{\gamma}x^2 - \frac{\beta_0}{\gamma}x - \frac{\gamma_0}{\gamma}\right)^2 - \log\gamma = \\ -2\log p_y(y) + \gamma'\left(x - \frac{\alpha_0'}{\gamma'}y^2 - \frac{\beta_0'}{\gamma'}y - \frac{\gamma_0'}{\gamma'}\right)^2 - \log\gamma', \quad (5.39)$$

which can be expanded into, after grouping together monomials:

$$\begin{aligned} -2\log p_x(x) &+ \frac{\alpha_0^2}{\gamma}x^4 + 2\frac{\alpha_0\beta_0}{\gamma}x^3 + \left(\frac{\beta_0^2 + \alpha_0\gamma_0}{\gamma} - \gamma'\right)x^2 \\ &+ 2\left(\frac{\beta_0\gamma_0}{\gamma} + \gamma_0'\right)x - 2\alpha_0 x^2 y - 2\beta_0 xy + \text{const} \\ = -2\log p_y(y) &+ \frac{{\alpha_0'}^2}{\gamma'}y^4 + 2\frac{\alpha_0'\beta_0'}{\gamma'}y^3 + \left(\frac{{\beta_0'}^2 + \alpha_0'\gamma_0'}{\gamma'} - \gamma\right)y^2 \\ &+ 2\left(\frac{\beta_0'\gamma_0'}{\gamma'} + \gamma_0\right)y - 2\alpha_0' y^2 x - 2\beta_0' xy, \quad (5.40) \end{aligned}$$

or again

$$A(x) - B(y) - 2\alpha_0 x^2 y + 2\alpha_0' y^2 x + 2(\beta_0' - \beta_0)xy = \text{const}, \quad (5.41)$$

where

$$\begin{aligned} A(x) &= -2\log p_x(x) + \frac{\alpha_0^2}{\gamma}x^4 + 2\frac{\alpha_0\beta_0}{\gamma}x^3 \quad (5.42) \\ &+ \left(\frac{\beta_0^2 + \alpha_0\gamma_0}{\gamma} - \gamma'\right)x^2 + 2\left(\frac{\beta_0\gamma_0}{\gamma} + \gamma_0'\right)x, \\ B(y) &= -2\log p_y(y) + \frac{{\alpha_0'}^2}{\gamma'}y^4 + 2\frac{\alpha_0'\beta_0'}{\gamma'}y^3 \quad (5.43) \\ &+ \left(\frac{{\beta_0'}^2 + \alpha_0'\gamma_0'}{\gamma'} - \gamma\right)y^2 + 2\left(\frac{\beta_0'\gamma_0'}{\gamma'} + \gamma_0\right)y. \end{aligned}$$

By setting $y = 0$ in equation (5.41), we have that $A(x) = \text{const}$ for all x. Similarly, by setting $x = 0$, we get $B(y) = \text{const}$ for all y. We conclude that the remaining monomials must be zero. In particular, this implies that $\alpha_0 = \alpha_0' = 0$ and $\beta_0 = \beta_0'$. This in turn means that f and g are linear.

Finally, by plugging this into equations (5.42) and (5.43), we get:

$$\log p_x(x) = \frac{1}{2}\left(\frac{\beta_0^2}{\gamma} - \gamma'\right)x^2 + \left(\frac{\beta_0\gamma_0}{\gamma} + \gamma_0'\right)x + \text{const}. \quad (5.44)$$

$$\log p_y(y) = \frac{1}{2}\left(\frac{{\beta_0'}^2}{\gamma'} - \gamma\right)y^2 + \left(\frac{\beta_0'\gamma_0'}{\gamma'} + \gamma_0\right)y + \text{const}'. \quad (5.45)$$

We deduce that x and y must be Gaussian. We don't prove the normalizability of the probability density functions in detail here since it is well-known that such Gaussian, unidentifiable models exist.

Third (and fourth) case: 1. + 2'. or 2. + 1'. Since these two cases are symmetric, we will suppose that v is constant (2.) and $\overline{w}$ is a polynomial of second degree (1'.). Going back to equation (5.14) and plugging the expressions for f, v, g, w:

$$\begin{aligned}\log p_y(y) + \frac{1}{2}\log(\alpha' y^2 + \beta' y + \gamma') - \frac{1}{2}\frac{(\alpha'_0 y^2 + \beta'_0 y + \gamma'_0)^2}{\alpha' y^2 + \beta' y + \gamma'} \\ - \gamma_0 y + \frac{1}{2}\gamma y^2 + (\alpha'_0 y^2 + \beta'_0 y)x - \frac{1}{2}(\alpha' y^2 + \beta' y)x^2 \\ = \log p_x(x) + \frac{1}{2}\log(\gamma) - \frac{1}{2}\frac{(\alpha_0 x^2 + \beta_0 x + \gamma_0)^2}{\gamma} \\ - \gamma'_0 x + \frac{1}{2}\gamma' x^2 + (\alpha_0 x^2 + \beta_0 x)y, \quad (5.46)\end{aligned}$$

or again

$$A(x) - B(y) + \frac{1}{2}\alpha' x^2 y^2 + \left(\alpha_0 - \frac{1}{2}\beta'\right) x^2 y - \alpha'_0 x y^2 + (\beta_0 - \beta'_0)xy = 0, \quad (5.47)$$

where

$$\begin{aligned}A(x) &= \log p_x(x) + \frac{1}{2}\log(\gamma) - \frac{1}{2}\frac{(\alpha_0 x^2 + \beta_0 x + \gamma_0)^2}{\gamma} \quad (5.48)\\ &\quad - \gamma'_0 x + \frac{1}{2}\gamma' x^2, \\ B(y) &= \log p_y(y) + \frac{1}{2}\log(\alpha' y^2 + \beta' y + \gamma') - \frac{1}{2}\frac{(\alpha'_0 y^2 + \beta'_0 y + \gamma'_0)^2}{\alpha' y^2 + \beta' y + \gamma'} \quad (5.49)\\ &\quad - \gamma_0 y + \frac{1}{2}\gamma y^2.\end{aligned}$$

Proceeding like above, we can deduce that $A(x) - B(y)$ is a constant, and that all the monomials in equation (5.47) are zero. In particular, $\alpha' = 0$, which contradicts 1'.: this third case is thus not possible. □

5.A.2 Affine autoregressive flows are transitive

Proposition 5.4. *Consider 2 autoregressive transformations* $\mathbf{f}$ *and* $\mathbf{g}$ *with the same ordering* π*. Then their composition* $\mathbf{h} = \mathbf{g} \circ \mathbf{f}$ *is also an autoregressive with the same ordering* π*.*

Proof. Without loss of generality, assume that π is the identity. Let $(\mathbf{x}, \mathbf{y}, \mathbf{z})$ be such that

$$\mathbf{y} = \mathbf{f}(\mathbf{z}), \tag{5.50}$$
$$\mathbf{x} = \mathbf{g}(\mathbf{y}) = \mathbf{g} \circ \mathbf{f}(\mathbf{z}). \tag{5.51}$$

Since $\mathbf{f}$ and $\mathbf{g}$ are autoregressive, we can rewrite this system using equation (5.4) as

$$y_i = \tau(z_i, \mathbf{y}_{<i}), \tag{5.52}$$
$$x_j = \tau'(y_j, \mathbf{x}_{<j}). \tag{5.53}$$

The transformers τ and τ' are invertible with respect to their first argument. Denoting those inverses as α and α'. Then

$$z_i = \alpha(y_i, y_{<i}) \tag{5.54}$$
$$y_j = \alpha'(x_j, x_{<j}) \tag{5.55}$$

And thus,

$$z_i = \alpha(\alpha'(x_i, x_{<i}), \beta(x_{<i})) \tag{5.56}$$

for some function β (not necessarily invertible). Since α and α' are invertible with respect to their first argument, this means that the mapping $x_i \mapsto z_i$ in equation (5.56) is also invertible, and we can write

$$x_i = \tau''(z_i, x_{<i}), \tag{5.57}$$

where τ'' is invertible wrt to its first argument. This proves that $\mathbf{h} = \mathbf{g} \circ \mathbf{f}$ is also an autoregressive flow. □

Proposition 5.5. *Consider k affine autoregressive flows $\mathbf{T}_1, \dots, \mathbf{T}_k$ of the form (5.6) with the same ordering π. Then their composition $\mathbf{T} = \mathbf{T}_1 \circ \cdots \circ \mathbf{T}_k$ is also an affine autoregressive flow of the form (5.6) with the same ordering π.*

Proof. We will suppose that $d = 2$. The proof for $d > 2$ is very similar but requires more complex notations. We will denote by z_l^j the j-th ($j = 1, 2$) output of the l-th sub-flow. Not that we can parametrize $\mathbf{T}$ or $\mathbf{T}^{-1}$ to be an affine transformation. In these notations, if $\mathbf{T}$ follows equation (5.6), then

$\mathbf{z}^k = \mathbf{z}$ and $\mathbf{z}^0 = \mathbf{x}$. If instead, $\mathbf{T}^{-1}$ follows equation (5.6), then $\mathbf{z}^0 = \mathbf{z}$ and $\mathbf{z}^k = \mathbf{x}$. Each flow $l \geq 1$ has the expression:

$$z_1^l = \left(z_1^{l-1} - t_1^l\right) e^{-s_1^l}, \tag{5.58}$$

$$z_2^l = \left(z_2^{l-1} - t_2^l(z_1^l)\right) e^{-s_2^l(z_1^l)}. \tag{5.59}$$

First, define

$$\overline{s}_1^{l,k} = \sum_{j=l+1}^{k} s_1^j, \tag{5.60}$$

$$\overline{t}_1^{l,k} = \sum_{j=l+1}^{k} t_1^j e^{\sum_{i=l+1}^{j-1} s_1^i} = \sum_{j=l+1}^{k} t_1^j e^{\overline{s}_1^{l,j-1}}, \tag{5.61}$$

where all sums are zero if they have no summands. Then it is easy to show by induction using equation (5.58) that

$$z_1^l = e^{\overline{s}_1^{l,k}} z_1^k + \overline{t}_1^{l,k}, \ \forall l \leq k, \tag{5.62}$$

and that

$$e^{\overline{s}_1^{l,j}} z_1^j + \overline{t}_1^{l,j} = e^{\overline{s}_1^{l,k}} z_1^k + \overline{t}_1^{l,k}, \ \forall l \leq \min(j,k). \tag{5.63}$$

Second, define

$$\overline{s}_2^k(u) = \sum_{l=1}^{k} s_2^l(e^{\overline{s}_1^{l,k}} u + \overline{t}_1^{l,k}), \tag{5.64}$$

$$\overline{t}_2^k(u) = \sum_{l=1}^{k} t_2^l(e^{\overline{s}_1^{l,k}} u + \overline{t}_1^{l,k}) e^{\sum_{i=1}^{l-1} s_2^i(e^{\overline{s}_1^{i,k}} u + \overline{t}_1^{i,k})}. \tag{5.65}$$

We will show by induction on k that

$$z_2^k = \left(z_2^0 - \overline{t}_2^k(z_1^k)\right) e^{-\overline{s}_2^k(z_1^k)}. \tag{5.66}$$

The case for $k = 1$ trivially holds. Now suppose that equation (5.66) holds for $k \geq 1$, and let's show it also holds for $k+1$. Using equation (5.59), we can write

$$z_2^{k+1} = \left(z_2^k - t_2^{k+1}(z_1^{k+1})\right) e^{-s_2^{k+1}(z_1^{k+1})}. \tag{5.67}$$

We need to show that

$$\overline{s}_2^{k+1}(z_1^{k+1}) = s_2^{k+1}(z_1^{k+1}) + \overline{s}_2^k(z_1^k), \tag{5.68}$$

$$\overline{t}_2^{k+1}(z_1^{k+1}) = t_2^{k+1}(z_1^{k+1}) e^{\overline{s}_2^k(z_1^k)} + \overline{t}_2^k(z_1^k). \tag{5.69}$$

This can be done using equation (5.63), the fact that $z_1^{k+1} = e^{\bar{s}_1^{k+1,k+1}} z_1^{k+1} + \bar{t}_1^{k+1,k+1}$ and the definitions of $\bar{s}_2^k$ and $\bar{t}_2^k$, which in turn allows us to conclude the induction proof.

Finally, by replacing $\mathbf{z}^0$ and $\mathbf{z}^k$ by $\mathbf{x}$ and $\mathbf{z}$ respectively in equations (5.62) and (5.66), we have

$$x_1 = e^{\bar{s}_1^{0,k}} z_1 + \bar{t}_1^{0,k}, \tag{5.70}$$

$$x_2 = e^{\bar{s}_2^k(x_1)} z_2 + \bar{t}_2^k(x_1), \tag{5.71}$$

which proves the transitivity of affine autoregressive flows. □

5.A.3 Affine flows are not universal density approximators

Proposition 5.6. *Let $\mathbf{T} : \mathbb{R}^d \to \mathbb{R}^d$ be an affine autoregressive transformation. Let $\mathbf{z}$ be a standard Gaussian, and let $\mathbf{x} = \mathbf{T}(\mathbf{z})$. Then there is no parametrization of $\mathbf{T}$ such that $\mathbf{x}$ has an isotropic Gumbel distribution.*

Proof. It is enough to prove this theorem for $d = 2$. Let $\mathbf{x} = \mathbf{T}(\mathbf{z})$. Then

$$\log p_{\mathbf{x}}(\mathbf{x}) = \log p_{\mathbf{z}}(\mathbf{T}^{-1}(\mathbf{x})) + \log \left|\det J_{\mathbf{T}^{-1}}(\mathbf{x})\right| \tag{5.72}$$

and

$$x_1 = e^{s_1} z_1 + t_1, \tag{5.73}$$

$$x_2 = e^{s_2(x_1)} z_2 + t_i(x_1). \tag{5.74}$$

The Jacobian log-determinant of $\mathbf{T}^{-1}$ is simply $\log \left|\det J_{\mathbf{T}^{-1}}(\mathbf{z})\right| = -s_1 - s_2(x_1)$. Note that this determinant is only a function of x_1. This is the main reason why affine autoregressive flows are not universal density approximators.

To see this, suppose that x_1 and x_2 are independent, and that each has a Gumbel distribution. Plugging this into equation (5.72), we get

$$\begin{aligned} -\left(x_1 + e^{-x_1}\right) - \left(x_2 + e^{-x_2}\right) = \\ - s_1 - s_2(x_1) - (x_1 - t_1)^2 e^{-2s_1} - (x_2 - t_2(x_1))^2 e^{-2s_2(x_1)}. \end{aligned} \tag{5.75}$$

This equation is valid for all $(x_1, x_2) \in \mathbb{R}^2$. In particular, let $x_1 = 0$. Then for any x_2, after rearranging and grouping terms, we get

$$e^{-x_2} = \alpha x_2^2 + \beta x_2 + \gamma. \tag{5.76}$$

This can't hold for all values of x_2, which results in a contradiction. Thus, we conclude that an affine autoregressive flow can't represent any distribution, unlike general unconstrained autoregressive flows. □

5.A.4 Universality of the causal function

Proposition 5.7. *Consider k affine autoregressive flows $\mathbf{T}_1, \dots, \mathbf{T}_k$, and let $\mathbf{T} = \mathbf{T}_1 \circ \cdots \circ \mathbf{T}_k$. Denote by t_j^l and s_j^l the coefficients of the l-th sub-flow $\mathbf{T}_l$, and by $\bar{t}_j^k$ and $\bar{s}_j^k$ those of $\mathbf{T}$. Suppose that all of the s_j^l and t_j^l are feed-forward neural networks that have universal approximation capability (assuming all technical conditions hold). Then $\bar{t}_j^k$ and $\bar{s}_j^k$ also have universal approximation capability.*

Proof. We will suppose for the proof that $d = 2$. The proof for $d > 2$ is similar. According to Proposition 5.5, $\mathbf{T}$ is also an affine autoregressive flow, and $\bar{t}_2^k$ and $\bar{s}_2^k$ have the following expressions:

$$\bar{s}_2^k(u) = \sum_{l=1}^{k} s_2^l(e^{\bar{s}_1^{l,k}} u + \bar{t}_1^{l,k}), \tag{5.77}$$

$$\bar{t}_2^k(u) = \sum_{l=1}^{k} t_2^l(e^{\bar{s}_1^{l,k}} u + \bar{t}_1^{l,k}) e^{\sum_{i=1}^{l-1} s_2^i(e^{\bar{s}_1^{i,k}} u + \bar{t}_1^{i,k})}, \tag{5.78}$$

where $\bar{t}_1^{l,k}$ and $\bar{s}_1^{l,k}$ are defined in equations (5.60) and (5.61) respectively.

On the one hand, translating and scaling the argument u of $\bar{s}_2^k$ by $\bar{t}_1^{l,k}$ and $\bar{s}_1^{l,k}$ only changes the bias and the slope of the input layer of each of the s_2^l, $l = 1, \dots, k$. Thus, one can interpret equation (5.77) as the output of an additional final layer of the neural network whose outputs are the s_2^l functions. The number of flows k in this case increases the width of this final layer. Using the classical result of the universal approximation theorem of feed-forward networks with arbitrary width (Hornik, 1991), we conclude that $\bar{s}_2^k$ also satisfies such properties.

Interestingly, note that this results holds even if each of the s_j^l function is simply an affine function followed by a nonlinearity (*i.e.* a 1-hidden layer feed-forward network).

On the other hand, since each of the t_2^l have universal approximation capability, each can in particular approximate a function of the form $u \mapsto$

$f_l(u)e^{\sum_{i=1}^{l-1} s_2^i(e^{\bar{s}_1^{i,k}}u+\bar{t}_1^{i,k})}$, where f_l is a simple affine function followed by a nonlinearity σ (*i.e.* a 1-hidden layer feed-forward network). Thus, $\bar{t}_2^k$ can approximate a function of the form $\sum_{l=1}^k f_l$, which, by the same argument used above, will have universal approximation capability (Hornik, 1991). □

5.B Experimental protocol

5.B.1 Architectures and hyperparameters

The optimization was done using Adam, with learning rate $\texttt{lr} = 0.001$, $\boldsymbol{\beta} = (0.9, 0.999)$, along with a scheduler that reduces the learning rate by a factor of 0.1 on plateaux. All flows use an isotropic Laplace distribution as a prior. The different architectures and hyperparameter used for the experiments are as follows:

- **Causal discovery simulations:** The flow $\mathbf{T}$ is a composition of 2 sub-flows $\mathbf{T}_1$ and $\mathbf{T}_2$. For each of the $\mathbf{T}_l$, both s_j and t_j are multi-layer perceptrons (MLPs), with 1 hidden layer and 10 hidden units. Each direction was trained for 200 epochs, with a mini-batch of 128 data points. The same architecture was used for all panels of Figure [5.1].

- **Cause-effect pairs:** The flow $\mathbf{T}$ is a composition of 4 sub-flows $\mathbf{T}_1$ to $\mathbf{T}_4$. For each of the $\mathbf{T}_l$, both s_j and t_j are MLPs, with either 1 or 3 hidden layers, each with 5 hidden units. For each direction, we train two different flows (with 1 or 3 hidden layers), and select the flow that yields higher test likelihood. Each direction was trained for 750 epochs, with a mini-batch of 128 data points. For each pair, 80% of the data points were used for training, and the remaining 20% to evaluate the likelihood. The same architecture was used to classify all the pairs.

- **EEG arrow of time:** The flow $\mathbf{T}$ is a composition of 4 sub-flows $\mathbf{T}_1, \cdots, \mathbf{T}_4$. For each of the $\mathbf{T}_l$, both s_j and t_j are MLPs, with 4 hidden layers, each with 10 hidden units. Each direction was trained for 400 epochs, with a mini-batch of 32 data points. For each channel, 80% of the data points were used for training, and the remaining 20% to evaluate the likelihood. The same architecture was used to classify all the channels.

- **Interventions on simulated data:** The flow $\mathbf{T}$ is a composition of 5 sub-flows $\mathbf{T}_1, \cdots, \mathbf{T}_5$. For each of the $\mathbf{T}_l$, both s_j and t_j are MLPs, with 1 hidden layers, each with 10 hidden units. We train the flow, conditioned on the causal ordering, to fit the correct SEM. Training was done for 750 epochs, with a mini-batch of 32 data points.

- **Interventions on es-fMRI data:** The flow $\mathbf{T}$ is a composition of 5 sub-flows $\mathbf{T}_1, \cdots, \mathbf{T}_5$. For each of the $\mathbf{T}_l$, both s_j and t_j are MLPs, with a single hidden layer consisting of 2 hidden units. In order to obtain interventional predictions, a CAREFL model was first trained using resting-state fMRI data conditioned upon the causal ordering. Since we did not seek to infer the causal structure, 100% of the training data was employed (this is in contrast to causal discovery experiments which only trained models on 80% of the data).

5.B.2 Preprocessing of EEG data

The openly available EEG data from Dornhege et al. (2004) contains recordings for 5 healthy subjects. For each subject, the data has been sampled at 100Mhz and 1000Mhz. For our experiments, we considered subject number 3, and used the data sampled at 1000Mhz. In particular, we only considered $n = 150$ and $n = 500$ time points. Each of the 118 EEG channels was then reversed with probability 0.5.

The task is to properly infer the arrow of time for each of the 118 EEG, considered separately. We transform a univariate time series $(x_t)_{t\in[\![1,n]\!]}$ corresponding to 1 channel into bivariate causal data by shifting it by a lag parameter l, to obtain data of the form $(x_t, x_{t+l})_{t\in[\![1,n-l]\!]}$. For the results plotted in Figure [5.4], we used three values of lag for ANM, RECI, the linear LR and CAREFL-NS: $l \in \{1, 2, 3\}$, which we then combined into one dataset. For CAREFL, we used only two values of lag: $l \in \{1, 2\}$.

5.B.3 Preprocessing of functional MRI data

Results included in this manuscript come from preprocessing performed using `FMRIPREP` (Esteban et al., 2019), a `Nipype` based tool (Gorgolewski et al., 2011). Each T1w (T1-weighted) volume was corrected for INU (intensity non-

uniformity) using `N4BiasFieldCorrection v2.1.0` and skull-stripped using the OASIS template from `antsBrainExtraction.sh v2.1.0` . Brain surfaces were reconstructed using recon-all from `FreeSurfer v6.0.1`, and the brain mask estimated previously was refined with a custom variation of the method to reconcile `ANTs`-derived and `FreeSurfer`-derived segmentations of the cortical grey-matter of Mindboggle. Spatial normalization to the ICBM 152 Nonlinear Asymmetrical template version 2009c was performed through nonlinear registration with the `antsRegistration` tool of `ANTs v2.1.0`, using brain-extracted versions of both T1w volume and template. Brain tissue segmentation of cerebrospinal fluid (CSF), white-matter (WM) and gray-matter (GM) was performed on the brain-extracted T1w using `fast`.

Functional data was slice time corrected using `3dTshift` from `AFNI v16.2.07` and motion corrected using `mcflirt`. This was followed by co-registration to the corresponding T1w using boundary-based registration with six degrees of freedom, using `bbregister` (`FreeSurfer v6.0.1`). Motion correcting transformations, BOLD-to-T1w transformation and T1w-to-template (MNI) warp were concatenated and applied in a single step using `antsApplyTransforms` using Lanczos interpolation.

Regional time series were subsequently calculated from the processed FMRI data (transformed into MNI space) using `NiLearn` (Abraham et al., 2014) and the Harvard-Atlas probabilistic atlas, with regions thresholded at 25% probability and binarised. Given the regional location of intracortical stimulation in the subjects, FMRI time series from the Cingulate gyrus and Heschl's gyrus were selected for analysis.

We note that each patient received surgery and stimulation in different locations, as determined by their diagnosis and clinical criteria. As such, the two regions studied were selected so as to include as many subjects as possible in our experiments. Moreover, the Cingulate gyrus is a region associated with cognitive processes such as saliency and emotional processing (Vogt, 2019) whereas Heschl's gyrus covers primary auditory cortex, associated with early cortical processing of auditory information; as such connectivity between the regions captures the interaction between a higher-order heteromodal region and a unimodal sensory region.

Conclusion

Thesis summary

Modern neural network methods offer few statistical guarantees — even retraining the same model architecture with a different initialisation or a slightly different dataset may lead to a radically different outcome. This is particularly problematic in representation learning, where one often wants the learnt features to be reproducible and interpretable. Identifiability is an essential prerequisite to mitigate this behaviour. It has been studied intensively in linear models, but proved much harder to understand in the nonlinear domain. In recent years, there has been a resurgence in identifiability results, primarily in the nonlinear ICA literature. This thesis expands on and extends these findings across a variety of popular deep learning frameworks. It sheds some light on the relevance of identifiability for such models and prepares the road for its application to other contexts and model classes.

The first three chapters of the thesis are devoted to the identifiability of the representations learnt by some of the most popular deep learning framework. In Chapter 2, we began by presenting a unifying view of two complementary unsupervised representation learning methods: nonlinear ICA and variational autoencoders (VAEs). The resulting framework is the first rigorous demonstration of identifiability in the setting of VAEs. This offers it an advantage over conventional deep latent variable models employed with VAEs for representation learning, because it retrieves the original latent variables, allowing for principled disentanglement. It is further established that the new maximum-likelihood-based estimation method has numerous advantages over prior nonlinear ICA algorithms. In summary, it generalizes to the case of noisy observations and incomplete representations, is more principled than heuristic self-supervised approaches in previous studies, and may be used as a proxy for hyperparameter

selection. Finally, it improves on previous identifiability results by focusing on factorial exponential families. In Chapter 3, we developed a further generalization of this model by removing the independence assumption while retaining identifiability. This novel framework, termed as Independently Modulated Component Analysis, extends nonlinear ICA to the case where the latent variables can have any global dependence structure as long as they are independently modulated by another variable such as a time index, history, or noisy labels. Even when linear ICA theory is taken into account, the new identifiability results are the most broad to date. Next, we sought to further weaken the assumptions required for identifiability. To this purpose, Chapter 4 presents a new identifiable conditional energy-based model (EBM) for unsupervised representation learning. This is the first energy-based model to benefit from rigorous identifiability results. The model makes use of EBMs' flexibility and generality, extending identifiability results to overcomplete representations and even having universal approximation capabilities. In addition, we demonstrated how to use it to estimate latent variables in nonlinear ICA and IMCA models. On real-world image datasets, we demonstrated empirically that identifiable representations increase performance in transfer learning and semi-supervised learning applications. This paves the way for many new applications of EBMs, by giving them a theoretically sound basis.

The last chapter of this thesis is dedicated to another type of identifiability required for causal models. We suggested in Chapter 5 to exploit the similarities between autoregressive normalizing flow models and structural equation models (SEM), leading to a novel understanding of variable ordering in an autoregressive flow as the causal ordering of a SEM. We showed that affine flows in particular define a new class of causal models in which the cause modulates the noise distribution. We proved a completely new causal identifiability result for such models, generalizing additive noise models. Subsequently, we argued that autoregressive flow models are well suited to causal inference tasks, ranging from causal discovery to making counterfactual predictions. On the one hand, we demonstrated how to efficiently learn the causal structure by selecting the ordering with the highest test log-likelihood, therefore offering a measure of causal direction based on the likelihood ratio for nonlinear SEMs. On the other hand, and thanks to their invertibility, autoregressive flow models may readily answer interventional and counterfactual queries.

Perspectives and future work

Extensions of the nonlinear ICA and IMCA models. The identifiability theorems presented in Chapters 2 and 3 limit the latent variables to be in the exponential family. This is done chiefly to reduce the number of assumptions necessary for such results. More specifically, if we represent the dimension (order) of the sufficient statistic of the i-th latent variable by k_i, then the identifiability results only apply to two cases: either $k_i \geq 2$, $\forall i$ or $k_i = 1$, $\forall i$. In future work, we can study the mixed setting, or we can expand the IMCA framework outside of the exponential family setting by building on past work on nonlinear ICA with more general factorial prior distributions (Hyvärinen et al., 2019).

Identifiability in the absence of the auxiliary variable. The need for an auxiliary variable to ensure identifiability is a limitation of the models described in Chapters 2 to 4. It is already generally understood that dealing with nonlinear models does not allow for identifiability in general. Auxiliary variables are one method to break such models' symmetries, resulting in more "repeatable" representations and making the models identifiable. A promising research direction would be to relax or eliminate the need for such auxiliary variables. This can be accomplished by automatically learning such auxiliary variables from observations by solving a secondary task for instance (Willetts and Paige, 2021). Another approach would be to cleverly constrain the nonlinear mapping between the latent and observed space, while retaining most of the transformation' flexibility (Gresele et al., 2021). Finally, we can exploit the data's structural dependencies as a naturally present inductive bias to attain identifiability (Hälvä et al., 2021).

IMCA and causal discovery in the presence of confounding. A potential application of the IMCA framework that was not explored here is causal discovery in the presence of confounding. A confounder is a hidden variable that affects both dependent and independent variables, resulting in spurious associations in the causal graph. The correspondence between nonlinear ICA and causal models was recently utilized by Monti et al. (2019) to conduct causal discovery on non-stationary observational data. We can establish a similar

relationship between IMCA and confounded structural equation models (SEM) since the IMCA model allows for the latent (noise) variables to be dependent. The identifiability of IMCA implies that the causal direction of such an SEM is likewise identifiable. However, because it is based on independence tests, the estimation technique of Monti et al. (2019) cannot be used here. Instead, a non-constraint-based method, such as likelihood ratio measures (Chapter 5), might be pursued.

Identifiability of intermediate layers. Intermediate layers in neural networks are frequently used as useful features for a downstream task. They may even be preferable in some applications over the representations learned by the final layer (Mikolov et al., 2013; Alain and Bengio, 2018; Chen et al., 2020). An intriguing question arises: can the identifiability results of the representations learnt by ICE-BeeM be generalized to previous layers? In fact, to demonstrate that the MLP architecture presented in Chapter 4 is identifiable, we used some form of induction to "propagate identifiability" forward through the network. Thus, a potential avenue of research is to prove that the intermediate layers preceding a final MLP "chunk" in a neural network can inherit the identifiability guarantees of the final layer. Moreover, we can look to prove the identifiability of more general architectures. Convolutional networks are a suitable initial choice since they are frequently utilized in image learning and have a strong mathematical theory (Wiatowski and Bölcskei, 2017).

Identifiability of the affine noise model with non-Gaussian noise. The identifiability theory of the additive noise model, which is a special case of the affine noise model, also hold for non-Gaussian noise variables. In trials using Laplace distributed noise variables, our causal autoregressive flow model proved successful in estimating the causal direction. Making this rigorous by expanding the identifiability proof in Chapter 5, using inspiration from Hoyer et al. (2009) and Zhang and Hyvärinen (2009), is a possible direction for future research.

Bibliography

Abraham, Alexandre, Fabian Pedregosa, Michael Eickenberg, Philippe Gervais, Andreas Mueller, Jean Kossaifi, Alexandre Gramfort, Bertrand Thirion, and Gaël Varoquaux. "Machine Learning for Neuroimaging with Scikit-Learn". *Frontiers in neuroinformatics* 8 (2014), p. 14 (cit. on p. 236).

Achille, Alessandro and Stefano Soatto. "Information Dropout: Learning Optimal Representations through Noisy Computation". *IEEE transactions on pattern analysis and machine intelligence* 40.12 (2018), pp. 2897–2905 (cit. on pp. 29, 31).

Alain, Guillaume and Yoshua Bengio. "Understanding Intermediate Layers Using Linear Classifier Probes". Version 4. Nov. 22, 2018. arXiv: 1610.01644 [cs, stat] (cit. on p. 240).

Alemi, Alexander A., Ben Poole, Ian Fischer, Joshua V. Dillon, Rif A. Saurous, and Kevin Murphy. "Fixing a Broken ELBO". Nov. 1, 2017. arXiv: 1711.00464 [cs, stat] (cit. on p. 29).

Andersen, Erling Bernhard. "Sufficiency and Exponential Families for Discrete Sample Spaces". *Journal of the American Statistical Association* 65.331 (Sept. 1, 1970), pp. 1248–1255. ISSN: 0162-1459. DOI: 10.1080/01621459.1970.10481160 (cit. on p. 24).

Arbel, Michael and Arthur Gretton. "Kernel Conditional Exponential Family". *International Conference on Artificial Intelligence and Statistics*. PMLR, Apr. 7, 2018, pp. 1337–1346. arXiv: 1711.05363 (cit. on p. 167).

Arbel, Michael, Danica J Sutherland, Mikołaj Bińkowski, and Arthur Gretton. "On Gradient Regularizers for MMD GANs". *Advances in Neural Information Processing Systems*. Vol. 31. Curran Associates, Inc., Nov. 29, 2018. arXiv: 1805.11565 (cit. on p. 182).

Back, Andrew D. and Andreas S. Weigend. "A First Application of Independent Component Analysis to Extracting Structure from Stock Returns". *International journal of neural systems* 8.04 (1997), pp. 473–484 (cit. on p. 32).

Bell, Anthony J. and Terrence J. Sejnowski. "An Information-Maximization Approach to Blind Separation and Blind Deconvolution". *Neural computation* 7.6 (1995), pp. 1129–1159 (cit. on p. 32).

Ben-Israel, Adi. "The Change-of-Variables Formula Using Matrix Volume". *SIAM J. Matrix Anal. Appl.* 21.1 (Oct. 1999), pp. 300–312. ISSN: 0895-4798. DOI: 10.1137/S0895479895296896 (cit. on p. 92).

Bengio, Yoshua, Réjean Ducharme, Pascal Vincent, and Christian Janvin. "A Neural Probabilistic Language Model". *The journal of machine learning research* 3 (2003), pp. 1137–1155 (cit. on p. 28).

Bengio, Yoshua, Aaron Courville, and Pascal Vincent. "Representation Learning: A Review and New Perspectives". *IEEE transactions on pattern analysis and machine intelligence* 35.8 (2013), pp. 1798–1828. arXiv: 1206.5538 (cit. on pp. 28, 29).

Bertin-Mahieux, Thierry, Daniel PW Ellis, Brian Whitman, and Paul Lamere. "The Million Song Dataset". *Proceedings of the 12th International Society for Music Information Retrieval Conference*. 2011. DOI: 10.7916/D8NZ8J07 (cit. on p. 27).

Bertsekas, Dimitri P. "Auction Algorithms for Network Flow Problems: A Tutorial Introduction". *Computational optimization and applications* 1.1 (1992), pp. 7–66 (cit. on p. 78).

Bingham, Ella, Jukka Kuusisto, and Krista Lagus. "ICA and SOM in Text Document Analysis". *Proceedings of the 25th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*. 2002, pp. 361–362 (cit. on p. 32).

Bloebaum, Patrick, Dominik Janzing, Takashi Washio, Shohei Shimizu, and Bernhard Schölkopf. "Cause-Effect Inference by Comparing Regression Errors". *International Conference on Artificial Intelligence and Statistics*. 2018, pp. 900–909 (cit. on pp. 40, 42, 43, 52, 200, 213, 221).

Bollen, Kenneth A. *Structural Equations with Latent Variables*. Vol. 210. John Wiley & Sons, 1989 (cit. on p. 39).

Bottou, Léon, Jonas Peters, Joaquin Quiñonero-Candela, Denis X. Charles, D. Max Chickering, Elon Portugaly, Dipankar Ray, Patrice Simard, and Ed Snelson. "Counterfactual Reasoning and Learning Systems: The Example of Computational Advertising." *Journal of Machine Learning Research* 14.11 (2013) (cit. on p. 38).

Brakel, Philemon and Yoshua Bengio. "Learning Independent Features with Adversarial Nets for Non-Linear ICA". 2017. arXiv: 1710.05050 (cit. on p. 63).

Brosowski, Bruno and Frank Deutsch. "An Elementary Proof of the Stone-Weierstrass Theorem". *Proceedings of the American Mathematical Society* (1981), pp. 89–92 (cit. on p. 178).

Burda, Yuri, Roger Grosse, and Ruslan Salakhutdinov. "Importance Weighted Autoencoders". Sept. 1, 2015. arXiv: 1509.00519 [cs, stat] (cit. on p. 62).

Burgess, Christopher P., Irina Higgins, Arka Pal, Loic Matthey, Nick Watters, Guillaume Desjardins, and Alexander Lerchner. "Understanding Disentangling in β-VAE". 2018. arXiv: 1804.03599 (cit. on pp. 29, 62).

Calhoun, Vince D., Tulay Adali, Lars Kai Hansen, Jan Larsen, and James J. Pekar. "ICA of Functional MRI Data: An Overview". *In Proceedings of the International Workshop on Independent Component Analysis and Blind Signal Separation*. Citeseer, 2003 (cit. on p. 32).

Cardoso, Jean-François. "The Three Easy Routes to Independent Component Analysis; Contrasts and Geometry". *Proc. ICA*. Vol. 2001. Citeseer, 2001, pp. 1–6 (cit. on p. 32).

Castro, Daniel C., Ian Walker, and Ben Glocker. "Causality Matters in Medical Imaging". *Nature Communications* 11.1 (2020), pp. 1–10 (cit. on p. 38).

Ceylan, Ciwan and Michael U. Gutmann. "Conditional Noise-Contrastive Estimation of Unnormalised Models". *International Conference on Machine Learning*. PMLR, 2018, pp. 726–734. arXiv: 1806.03664 (cit. on p. 132).

Chang, Angel X., Thomas Funkhouser, Leonidas Guibas, Pat Hanrahan, Qixing Huang, Zimo Li, Silvio Savarese, Manolis Savva, Shuran Song, and Hao Su. "Shapenet: An Information-Rich 3d Model Repository". 2015. arXiv: 1512.03012 (cit. on p. 27).

Chen, Ricky T. Q., Xuechen Li, Roger B Grosse, and David K Duvenaud. "Isolating Sources of Disentanglement in Variational Autoencoders". *Advances*

in Neural Information Processing Systems. Vol. 31. Curran Associates, Inc., 2018. arXiv: 1802.04942 (cit. on pp. 28–31, 62, 81, 107).

Chen, Ting, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. "A Simple Framework for Contrastive Learning of Visual Representations". June 30, 2020. arXiv: 2002.05709 [cs, stat] (cit. on p. 240).

Comon, Pierre. "Independent Component Analysis, a New Concept?" *Signal processing* 36.3 (1994), pp. 287–314 (cit. on pp. 31–33, 114).

Dahl, George E., Dong Yu, Li Deng, and Alex Acero. "Context-Dependent Pre-Trained Deep Neural Networks for Large-Vocabulary Speech Recognition". *IEEE Transactions on audio, speech, and language processing* 20.1 (2011), pp. 30–42 (cit. on p. 28).

Darmois, George. "Analyse Générale Des Liaisons Stochastiques: Etude Particulière de l'analyse Factorielle Linéaire". *Revue de l'Institut International de Statistique* (1953), pp. 2–8 (cit. on p. 34).

Delorme, Arnaud, Terrence Sejnowski, and Scott Makeig. "Enhanced Detection of Artifacts in EEG Data Using Higher-Order Statistics and Independent Component Analysis". *Neuroimage* 34.4 (2007), pp. 1443–1449 (cit. on p. 32).

Deng, Jia, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. "Imagenet: A Large-Scale Hierarchical Image Database". *IEEE Conference on Computer Vision and Pattern Recognition*. IEEE, 2009, pp. 248–255 (cit. on p. 27).

Devlin, Jacob, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding". *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*. NAACL-HLT 2019. Minneapolis, Minnesota: Association for Computational Linguistics, June 2019, pp. 4171–4186. DOI: 10.18653/v1/N19-1423. arXiv: 1810.04805 (cit. on p. 28).

Dinh, Laurent, David Krueger, and Yoshua Bengio. "NICE: Non-linear Independent Components Estimation". Oct. 30, 2014. arXiv: 1410.8516 [cs] (cit. on pp. 51, 67, 201).

Dinh, Laurent, Jascha Sohl-Dickstein, and Samy Bengio. "Density Estimation Using Real NVP". May 27, 2016. arXiv: 1605.08803 [cs, stat] (cit. on pp. 51, 201, 205, 209).

Dornhege, Guido, Benjamin Blankertz, Gabriel Curio, and Klaus-Robert Müller. "Boosting Bit Rates in Noninvasive EEG Single-Trial Classifications by Feature Combination and Multiclass Paradigms". *IEEE transactions on bio-medical engineering* 51.6 (June 2004), pp. 993–1002. ISSN: 0018-9294. DOI: 10.1109/TBME.2004.827088. pmid: 15188870 (cit. on pp. 217, 235).

Durkan, Conor, Artur Bekasov, Iain Murray, and George Papamakarios. "Neural Spline Flows". *Advances in Neural Information Processing Systems*. Vol. 32. Curran Associates, Inc., 2019. arXiv: 1906.04032 (cit. on pp. 165, 201, 205, 212).

— "Cubic-Spline Flows". June 5, 2019. arXiv: 1906.02145 [cs, stat] (cit. on pp. 201, 205).

Eriksson, Jan and Visa Koivunen. "Blind Identifiability of Class of Nonlinear Instantaneous ICA Models". *2002 11th European Signal Processing Conference*. IEEE, 2002, pp. 1–4 (cit. on p. 35).

Esmaeili, Babak, Hao Wu, Sarthak Jain, Alican Bozkurt, Narayanaswamy Siddharth, Brooks Paige, Dana H. Brooks, Jennifer Dy, and Jan-Willem Meent. "Structured Disentangled Representations". *The 22nd International Conference on Artificial Intelligence and Statistics*. PMLR, 2019, pp. 2525–2534. arXiv: 1804.02086 (cit. on pp. 28, 29, 31, 63).

Esteban, Oscar, Christopher J. Markiewicz, Ross W. Blair, Craig A. Moodie, A. Ilkay Isik, Asier Erramuzpe, James D. Kent, Mathias Goncalves, Elizabeth DuPre, and Madeleine Snyder. "fMRIPrep: A Robust Preprocessing Pipeline for Functional MRI". *Nature methods* 16.1 (2019), pp. 111–116 (cit. on p. 235).

Foster, E. Michael. "Causal Inference and Developmental Psychology." *Developmental psychology* 46.6 (2010), p. 1454 (cit. on p. 38).

Gangl, Markus. "Causal Inference in Sociological Research". *Annual review of sociology* 36 (2010), pp. 21–47 (cit. on p. 38).

Gao, Ruiqi, Erik Nijkamp, Diederik P. Kingma, Zhen Xu, Andrew M. Dai, and Ying Nian Wu. "Flow Contrastive Estimation of Energy-Based Models". *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2020, pp. 7518–7528. arXiv: 1912.00589 (cit. on pp. 133, 165, 169, 170).

Gao, Shuyang, Rob Brekelmans, Greg Ver Steeg, and Aram Galstyan. "Auto-Encoding Total Correlation Explanation". *The 22nd International Confer-*

ence on Artificial Intelligence and Statistics. PMLR, 2019, pp. 1157–1166 (cit. on pp. 29, 31).

Germain, Mathieu, Karol Gregor, Iain Murray, and Hugo Larochelle. "MADE: Masked Autoencoder for Distribution Estimation". Feb. 11, 2015. arXiv: 1502.03509 [cs, stat] (cit. on p. 205).

Glymour, Clark, Kun Zhang, and Peter Spirtes. "Review of Causal Discovery Methods Based on Graphical Models". *Frontiers in Genetics* 10 (June 4, 2019), p. 524. ISSN: 1664-8021. DOI: 10.3389/fgene.2019.00524 (cit. on p. 38).

Gorgolewski, Krzysztof, Christopher D. Burns, Cindee Madison, Dav Clark, Yaroslav O. Halchenko, Michael L. Waskom, and Satrajit S. Ghosh. "Nipype: A Flexible, Lightweight and Extensible Neuroimaging Data Processing Framework in Python". *Frontiers in neuroinformatics* 5 (2011), p. 13 (cit. on p. 235).

Goyal, Anirudh, Alex Lamb, Jordan Hoffmann, Shagun Sodhani, Sergey Levine, Yoshua Bengio, and Bernhard Schölkopf. "Recurrent Independent Mechanisms". July 2, 2020. arXiv: 1909.10893 [cs, stat] (cit. on p. 38).

Gresele, Luigi, Paul K. Rubenstein, Arash Mehrjou, Francesco Locatello, and Bernhard Schölkopf. "The Incomplete Rosetta Stone Problem: Identifiability Results for Multi-View Nonlinear ICA". *Uncertainty in Artificial Intelligence*. PMLR, 2020, pp. 217–227. arXiv: 1905.06642 (cit. on p. 36).

Gresele, Luigi, Julius von Kügelgen, Vincent Stimper, Bernhard Schölkopf, and Michel Besserve. "Independent Mechanism Analysis, a New Concept?" Oct. 28, 2021. arXiv: 2106.05200 [cs, stat] (cit. on p. 239).

Gretton, Arthur, Olivier Bousquet, Alex Smola, and Bernhard Schölkopf. "Measuring Statistical Dependence with Hilbert-Schmidt Norms". *International Conference on Algorithmic Learning Theory*. Springer, 2005, pp. 63–77 (cit. on p. 85).

Grosz, Michael P., Julia M. Rohrer, and Felix Thoemmes. "The Taboo against Explicit Causal Inference in Nonexperimental Psychology". *Perspectives on Psychological Science* 15.5 (2020), pp. 1243–1255 (cit. on p. 38).

Gutmann, Michael and Aapo Hyvärinen. "Noise-Contrastive Estimation: A New Estimation Principle for Unnormalized Statistical Models". *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics*. 2010, pp. 297–304 (cit. on p. 132).

Gutmann, Michael U. and Aapo Hyvärinen. "Noise-Contrastive Estimation of Unnormalized Statistical Models, with Applications to Natural Image Statistics". *Journal of Machine Learning Research* 13 (Feb 2012), pp. 307–361 (cit. on p. 169).

Hälvä, Hermanni, Sylvain Le Corff, Luc Lehéricy, Jonathan So, Yongjie Zhu, Elisabeth Gassiat, and Aapo Hyvärinen. "Disentangling Identifiable Features from Noisy Data with Structured Nonlinear ICA". June 17, 2021. arXiv: 2106.09620 [cs, stat] (cit. on p. 239).

Harmeling, Stefan, Andreas Ziehe, Motoaki Kawanabe, and Klaus-Robert Müller. "Kernel-Based Nonlinear Blind Source Separation". *Neural Computation* 15.5 (2003), pp. 1089–1124 (cit. on p. 35).

Hecht-Nielsen, Robert. "Replicator Neural Networks for Universal Optimal Source Coding". *Science* 269.5232 (1995), pp. 1860–1863 (cit. on p. 35).

Heckman, James J. "Econometric Causality". *International statistical review* 76.1 (2008), pp. 1–27 (cit. on p. 38).

Higgins, Irina, Loic Matthey, Arka Pal, Christopher Burgess, Xavier Glorot, Matthew Botvinick, Shakir Mohamed, and Alexander Lerchner. "Beta-VAE: Learning Basic Visual Concepts with a Constrained Variational Framework". *ICLR*. ICLR. Vol. 2. 2017 (cit. on pp. 28–30, 62, 81, 107).

Higgins, Irina, David Amos, David Pfau, Sebastien Racaniere, Loic Matthey, Danilo Rezende, and Alexander Lerchner. "Towards a Definition of Disentangled Representations". Dec. 5, 2018. arXiv: 1812.02230 [cs, stat] (cit. on p. 62).

Hornik, Kurt. "Approximation Capabilities of Multilayer Feedforward Networks". *Neural Networks* 4.2 (Jan. 1, 1991), pp. 251–257. ISSN: 0893-6080. DOI: 10.1016/0893-6080(91)90009-T (cit. on pp. 177, 233, 234).

Hosseini, Shahram and Christian Jutten. "On the Separability of Nonlinear Mixtures of Temporally Correlated Sources". *IEEE signal processing letters* 10.2 (2003), pp. 43–46 (cit. on p. 35).

Hoyer, Patrik O., Dominik Janzing, Joris M. Mooij, Jonas Peters, and Bernhard Schölkopf. "Nonlinear Causal Discovery with Additive Noise Models". *Advances in Neural Information Processing Systems*. 2009, pp. 689–696 (cit. on pp. 40–45, 52, 200, 203, 206, 213, 218, 221, 223, 240).

Hu, Weihua, Matthias Fey, Marinka Zitnik, Yuxiao Dong, Hongyu Ren, Bowen Liu, Michele Catasta, and Jure Leskovec. "Open Graph Benchmark: Datasets for Machine Learning on Graphs". 2020. arXiv: 2005.00687 (cit. on p. 27).

Huang, Chin-Wei, David Krueger, Alexandre Lacoste, and Aaron Courville. "Neural Autoregressive Flows". *International Conference on Machine Learning*. PMLR, July 3, 2018, pp. 2078–2087. arXiv: 1804.00779 (cit. on pp. 45, 51, 201, 203).

Hyvärinen, Aapo. "Fast and Robust Fixed-Point Algorithms for Independent Component Analysis". *IEEE transactions on Neural Networks* 10.3 (1999), pp. 626–634 (cit. on p. 32).

Hyvärinen, Aapo and Petteri Pajunen. "Nonlinear Independent Component Analysis: Existence and Uniqueness Results". *Neural Networks* 12.3 (Apr. 1, 1999), pp. 429–439. ISSN: 0893-6080. DOI: 10.1016/S0893-6080(98)00140-3 (cit. on pp. 31, 34, 35, 40, 46, 63, 102, 114, 130, 205).

Hyvärinen, Aapo, Juha Karhunen, and Erkki Oja. *Independent Component Analysis*. John Wiley & Sons, 2001 (cit. on pp. 32, 103).

Hyvärinen, Aapo, Jarmo Hurri, and Jaakko Väyrynen. "Bubbles: A Unifying Framework for Low-Level Statistical Properties of Natural Image Sequences". *JOSA A* 20.7 (2003), pp. 1237–1252 (cit. on p. 32).

Hyvärinen, Aapo and Jarmo Hurri. "Blind Separation of Sources That Have Spatiotemporal Variance Dependencies". *Signal Processing*. Special Section on Independent Component Analysis and Beyond 84.2 (Feb. 1, 2004), pp. 247–254. ISSN: 0165-1684. DOI: 10.1016/j.sigpro.2003.10.010 (cit. on pp. 32, 48, 114).

Hyvärinen, Aapo. "Estimation of Non-Normalized Statistical Models by Score Matching". *Journal of Machine Learning Research* 6 (Apr 2005), pp. 695–709 (cit. on pp. 132, 166).

— "Some Extensions of Score Matching". *Computational statistics & data analysis* 51.5 (2007), pp. 2499–2512 (cit. on p. 143).

Hyvärinen, Aapo and Stephen M. Smith. "Pairwise Likelihood Ratios for Estimation of Non-Gaussian Structural Equation Models". *Journal of Machine Learning Research* 14 (Jan 2013), pp. 111–152. ISSN: ISSN 1533-7928 (cit. on pp. 40, 42, 44, 52, 208, 213, 221).

Hyvärinen, Aapo and Hiroshi Morioka. "Unsupervised Feature Extraction by Time-Contrastive Learning and Nonlinear ICA". *Advances in Neural*

Information Processing Systems. 2016, pp. 3765–3773 (cit. on pp. 28, 35–37, 42, 47, 49, 50, 63, 66, 70, 77, 79, 82, 83, 88, 114, 130, 141, 147, 162, 164, 196, 221).

— "Nonlinear ICA of Temporally Dependent Stationary Sources". *Proceedings of The 20th International Conference on Artificial Intelligence and Statistics*. 2017 (cit. on pp. 28, 36, 37, 49, 63, 70, 77, 114, 130).

Hyvärinen, Aapo, Hiroaki Sasaki, and Richard Turner. "Nonlinear ICA Using Auxiliary Variables and Generalized Contrastive Learning". *Proceedings of The 22nd International Conference on Artificial Intelligence and Statistics*. Apr. 11, 2019, pp. 859–868 (cit. on pp. 28, 36–38, 42, 45–50, 63, 66, 70, 77, 79, 80, 82, 88, 114, 118, 121, 123, 126, 130, 131, 193, 221, 239).

Imai, Kosuke and David A. Van Dyk. "Causal Inference with General Treatment Regimes: Generalizing the Propensity Score". *Journal of the American Statistical Association* 99.467 (2004), pp. 854–866 (cit. on p. 38).

Jutten, Christian, Massoud Babaie-Zadeh, and Shahram Hosseini. "Three Easy Ways for Separating Nonlinear Mixtures?" *Signal Processing*. Special Section on Independent Component Analysis and Beyond 84.2 (Feb. 1, 2004), pp. 217–229. ISSN: 0165-1684. DOI: 10.1016/j.sigpro.2003.10.011 (cit. on p. 35).

Khemakhem, Ilyes, Diederik P. Kingma, Ricardo Pio Monti, and Aapo Hyvärinen. "Variational Autoencoders and Nonlinear ICA: A Unifying Framework". *Proceedings of The 23rd International Conference on Artificial Intelligence and Statistics*. Vol. 108. PMLR, June 2020, pp. 2207–2217. arXiv: 1907.04809 (cit. on pp. 61, 114, 118, 123, 125, 128, 130, 131, 137, 141, 147, 165, 193, 195, 196, 221).

Khemakhem, Ilyes, Ricardo Pio Monti, Diederik P. Kingma, and Aapo Hyvärinen. "ICE-BeeM: Identifiable Conditional Energy-Based Deep Models". *Advances in Neural Information Processing Systems*. Vol. 33. Dec. 2020. arXiv: 2002.11537 (cit. on pp. 113, 129, 221).

Khemakhem, Ilyes, Ricardo Pio Monti, Robert Leech, and Aapo Hyvärinen. "Causal Autoregressive Flows". *Proceedings of The 24th International Conference on Artificial Intelligence and Statistics*. The 24th International Conference on Artificial Intelligence and Statistics. Vol. 130. PMLR, 2021, pp. 3520–3528. arXiv: 2011.02268 (cit. on p. 199).

Kim, Hyunjik and Andriy Mnih. "Disentangling by Factorising". *International Conference on Machine Learning*. PMLR, 2018, pp. 2649–2658. arXiv: 1802.05983 (cit. on pp. 29–31, 63).

Kingma, Diederik P. and Max Welling. "Auto-Encoding Variational Bayes". *Proceedings of the 2nd International Conference on Learning Representations (ICLR)*. 2014. arXiv: 1312.6114 (cit. on pp. 28, 29, 62, 64, 68, 107, 122, 165).

Kingma, Diederik P. and Jimmy Ba. "Adam: A Method for Stochastic Optimization". Dec. 22, 2014. arXiv: 1412.6980 [cs] (cit. on pp. 80, 151).

Kingma, Diederik P., Tim Salimans, Rafal Jozefowicz, Xi Chen, Ilya Sutskever, and Max Welling. "Improved Variational Inference with Inverse Autoregressive Flow". *Advances in Neural Information Processing Systems*. Vol. 29. 2016, pp. 4743–4751. arXiv: 1606.04934 (cit. on pp. 62, 201, 212).

Kingma, Diederik P. and Prafulla Dhariwal. "Glow: Generative Flow with Invertible 1x1 Convolutions". *Advances in Neural Information Processing Systems*. Vol. 31. Curran Associates, Inc., 2018. arXiv: 1807.03039 (cit. on pp. 165, 201, 205).

Kobyzev, Ivan, Simon J. D. Prince, and Marcus A. Brubaker. "Normalizing Flows: An Introduction and Review of Current Methods". *IEEE Transactions on Pattern Analysis and Machine Intelligence* (2020), pp. 1–1. ISSN: 0162-8828, 2160-9292, 1939-3539. DOI: 10.1109/TPAMI.2020.2992934. arXiv: 1908.09257 (cit. on p. 201).

Kokonendji, Célestin C. and Tristan Senga Kiesse. "Discrete Associated Kernels Method and Extensions". *Statistical Methodology* 8.6 (2011), pp. 497–516 (cit. on p. 169).

Koopman, B. O. "On Distributions Admitting a Sufficient Statistic". *Transactions of the American Mathematical Society* 39.3 (1936), pp. 399–409. ISSN: 0002-9947, 1088-6850. DOI: 10.1090/S0002-9947-1936-1501854-3 (cit. on p. 24).

Korbar, Bruno, Du Tran, and Lorenzo Torresani. "Cooperative Learning of Audio and Video Models from Self-Supervised Synchronization". June 30, 2018. arXiv: 1807.00230 [cs] (cit. on p. 28).

Kreif, Noemi and Karla DiazOrdaz. "Machine Learning in Policy Evaluation: New Tools for Causal Inference". 2019. arXiv: 1903.00402 (cit. on p. 38).

Krizhevsky, A. "Learning Multiple Layers of Features from Tiny Images". 2009 (cit. on p. 27).

Kuhn, H. W. "The Hungarian Method for the Assignment Problem". *Naval Research Logistics Quarterly* 2.1-2 (Mar. 1, 1955), pp. 83–97. ISSN: 1931-9193. DOI: 10.1002/nav.3800020109 (cit. on p. 78).

Kumar, Abhishek, Prasanna Sattigeri, and Avinash Balakrishnan. "Variational Inference of Disentangled Latent Concepts from Unlabeled Observations". 2017. arXiv: 1711.00848 (cit. on p. 31).

Kupperman, Morton. "Probabilities of Hypotheses and Information-Statistics in Sampling from Exponential-Class Populations". *The Annals of Mathematical Statistics* 29.2 (June 1958), pp. 571–575. ISSN: 0003-4851, 2168-8990. DOI: 10.1214/aoms/1177706633 (cit. on p. 24).

Lacerda, Gustavo, Peter L. Spirtes, Joseph Ramsey, and Patrik O. Hoyer. "Discovering Cyclic Causal Models by Independent Components Analysis". 2012. arXiv: 1206.3273 (cit. on p. 40).

Lappalainen, Harri and Antti Honkela. "Bayesian Non-Linear Independent Component Analysis by Multi-Layer Perceptrons". *Advances in Independent Component Analysis*. Springer, 2000, pp. 93–121 (cit. on p. 35).

Le, Quoc V., Alexandre Karpenko, Jiquan Ngiam, and Andrew Y. Ng. "ICA with Reconstruction Cost for Efficient Overcomplete Feature Learning". *Advances in Neural Information Processing Systems 24*. Ed. by J. Shawe-Taylor, R. S. Zemel, P. L. Bartlett, F. Pereira, and K. Q. Weinberger. Curran Associates, Inc., 2011, pp. 1017–1025 (cit. on p. 32).

LeCun, Yann, Léon Bottou, Yoshua Bengio, and Patrick Haffner. "Gradient-Based Learning Applied to Document Recognition". *Proceedings of the IEEE* 86.11 (1998), pp. 2278–2324 (cit. on p. 27).

Lee, John M. *Introduction to Smooth Manifolds*. Graduate Texts in Mathematics. New York: Springer-Verlag, 2003. ISBN: 978-0-387-21752-9 (cit. on pp. 97, 98).

Lin, Guosheng, Anton Milan, Chunhua Shen, and Ian Reid. "RefineNet: Multi-Path Refinement Networks for High-Resolution Semantic Segmentation". *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2017, pp. 1925–1934. arXiv: 1611.06612 (cit. on p. 150).

Locatello, Francesco, Stefan Bauer, Mario Lucic, Gunnar Raetsch, Sylvain Gelly, Bernhard Schölkopf, and Olivier Bachem. "Challenging Common

Assumptions in the Unsupervised Learning of Disentangled Representations". *International Conference on Machine Learning*. PMLR, 2019, pp. 4114–4124. arXiv: 1811.12359 (cit. on pp. 31, 46, 63, 71).

Lopez-Paz, David, Philipp Hennig, and Bernhard Schölkopf. "The Randomized Dependence Coefficient". *Advances in Neural Information Processing Systems*. Vol. 26. Curran Associates, Inc., 2013 (cit. on pp. 79, 151).

Louizos, Christos, Uri Shalit, Joris Mooij, David Sontag, Richard Zemel, and Max Welling. "Causal Effect Inference with Deep Latent-Variable Models". May 24, 2017. arXiv: 1705.08821 [cs, stat] (cit. on p. 221).

Maaløe, Lars, Marco Fraccaro, Valentin Liévin, and Ole Winther. "BIVA: A Very Deep Hierarchy of Latent Variables for Generative Modeling". *Advances in Neural Information Processing Systems*. 2019, pp. 6548–6558 (cit. on p. 62).

Maas, Andrew, Raymond E. Daly, Peter T. Pham, Dan Huang, Andrew Y. Ng, and Christopher Potts. "Learning Word Vectors for Sentiment Analysis". *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*. 2011, pp. 142–150 (cit. on p. 27).

Makeig, Scott, Anthony Bell, Tzyy-Ping Jung, and Terrence J Sejnowski. "Independent Component Analysis of Electroencephalographic Data". *Advances in Neural Information Processing Systems*. Vol. 8. MIT Press, 1996 (cit. on p. 32).

Makeig, Scott, Tzyy-Ping Jung, Anthony J. Bell, Dara Ghahremani, and Terrence J. Sejnowski. "Blind Separation of Auditory Event-Related Brain Responses into Independent Components". *Proceedings of the National Academy of Sciences* 94.20 (1997), pp. 10979–10984 (cit. on p. 32).

Marcus, Mitchell, Beatrice Santorini, and Mary Ann Marcinkiewicz. "Building a Large Annotated Corpus of English: The Penn Treebank" (1993) (cit. on p. 27).

Mathieu, Emile, Tom Rainforth, N. Siddharth, and Yee Whye Teh. "Disentangling Disentanglement in Variational Autoencoders". Dec. 6, 2018. arXiv: 1812.02833 [cs, stat] (cit. on pp. 29, 31, 71).

McKeown, Martin J., Scott Makeig, Greg G. Brown, Tzyy-Ping Jung, Sandra S. Kindermann, Anthony J. Bell, and Terrence J. Sejnowski. "Analysis of fMRI Data by Blind Separation into Independent Spatial Components". *Human brain mapping* 6.3 (1998), pp. 160–188 (cit. on p. 32).

McKeown, Martin J. and Terrence J. Sejnowski. "Independent Component Analysis of fMRI Data: Examining the Assumptions". *Human brain mapping* 6.5-6 (1998), pp. 368–372 (cit. on p. 32).

Mikolov, Tomas, Kai Chen, Greg Corrado, and Jeffrey Dean. "Efficient Estimation of Word Representations in Vector Space". Sept. 6, 2013. arXiv: 1301.3781 [cs] (cit. on p. 240).

Milne, Elizabeth, Alison Scope, Olivier Pascalis, David Buckley, and Scott Makeig. "Independent Component Analysis Reveals Atypical Electroencephalographic Activity during Visual Perception in Individuals with Autism". *Biological psychiatry* 65.1 (2009), pp. 22–30 (cit. on p. 32).

Monti, Ricardo Pio and Aapo Hyvärinen. "A Unified Probabilistic Model for Learning Latent Factors and Their Connectivities from High-Dimensional Data". *Uncertainty in Artificial Intelligence* (Aug. 6, 2018). Ed. by Amir Globerson and Ricardo Silva, pp. 300–309. ISSN: 978-0-9966431-3-9. arXiv: 1805.09567 (cit. on pp. 48, 115, 119).

Monti, Ricardo Pio, Kun Zhang, and Aapo Hyvärinen. "Causal Discovery with General Non-Linear Relationships Using Non-Linear ICA". *35th Conference on Uncertainty in Artificial Intelligence, UAI 2019*. Vol. 35. 2019 (cit. on pp. 40–42, 44, 45, 84–86, 107, 134, 200, 220, 239, 240).

Mooij, Joris M., Jonas Peters, Dominik Janzing, Jakob Zscheischler, and Bernhard Schölkopf. "Distinguishing Cause from Effect Using Observational Data: Methods and Benchmarks". *The Journal of Machine Learning Research* 17.1 (2016), pp. 1103–1204 (cit. on p. 216).

Müller, Thomas, Brian McWilliams, Fabrice Rousselle, Markus Gross, and Jan Novák. "Neural Importance Sampling". Sept. 3, 2019. arXiv: 1808.03856 [cs, stat] (cit. on p. 205).

Murphy, Kevin and Saira Mian. *Modelling Gene Expression Data Using Dynamic Bayesian Networks*. Citeseer, 1999 (cit. on p. 38).

Neyman, Jerzy and Egon Sharpe Pearson. "IX. On the Problem of the Most Efficient Tests of Statistical Hypotheses". *Philosophical Transactions of the Royal Society of London. Series A, Containing Papers of a Mathematical or Physical Character* 231.694-706 (1933), pp. 289–337 (cit. on p. 208).

Nguyen, XuanLong, Martin J. Wainwright, and Michael I. Jordan. "Estimating Divergence Functionals and the Likelihood Ratio by Convex Risk Minimiza-

tion". *IEEE Transactions on Information Theory* 56.11 (2010), pp. 5847–5861 (cit. on p. 31).

Nuzillard, Danielle and Albert Bijaoui. "Blind Source Separation and Analysis of Multispectral Astronomical Images". *Astronomy and Astrophysics Supplement Series* 147.1 (2000), pp. 129–138 (cit. on p. 32).

Oja, E., K. Kiviluoto, and S. Malaroiu. "Independent Component Analysis for Financial Time Series". *Proceedings of the IEEE 2000 Adaptive Systems for Signal Processing, Communications, and Control Symposium (Cat. No.00EX373)*. Proceedings of the IEEE 2000 Adaptive Systems for Signal Processing, Communications, and Control Symposium (Cat. No.00EX373). Oct. 2000, pp. 111–116. DOI: 10.1109/ASSPCC.2000.882456 (cit. on p. 32).

Pajunen, Petteri, Aapo Hyvärinen, and Juha Karhunen. "Nonlinear Blind Source Separation by Self-Organizing Maps". *In Proc. Int. Conf. on Neural Information Processing*. 1996, pp. 1207–1210 (cit. on p. 35).

Papamakarios, George, Theo Pavlakou, and Iain Murray. "Masked Autoregressive Flow for Density Estimation". *Advances in Neural Information Processing Systems*. Vol. 30. Curran Associates, Inc., 2017. arXiv: 1705.07057 (cit. on pp. 201, 212).

Papamakarios, George, Eric Nalisnick, Danilo Jimenez Rezende, Shakir Mohamed, and Balaji Lakshminarayanan. "Normalizing Flows for Probabilistic Modeling and Inference". 2019. arXiv: 1912.02762 (cit. on p. 201).

Pawlowski, Nick, Daniel C. Castro, and Ben Glocker. "Deep Structural Causal Models for Tractable Counterfactual Inference". Oct. 22, 2020. arXiv: 2006.06485 [cs, stat] (cit. on p. 221).

Pearl, Judea. *Causality*. Cambridge: Cambridge University Press, 2009. ISBN: 978-0-521-89560-6. DOI: 10.1017/CBO9780511803161 (cit. on pp. 38, 200, 210, 211).

— "Causal Inference in Statistics: An Overview". *Statistics Surveys* 3 (2009), pp. 96–146. ISSN: 1935-7516. DOI: 10.1214/09-SS057 (cit. on pp. 39, 200, 211, 220).

Pearson, Karl and Francis Galton. "VII. Note on Regression and Inheritance in the Case of Two Parents". *Proceedings of the Royal Society of London* 58.347-352 (Jan. 1, 1895), pp. 240–242. DOI: 10.1098/rspl.1895.0041 (cit. on pp. 78, 79).

Peters, Jonas, Joris Mooij, Dominik Janzing, and Bernhard Schölkopf. "Causal Discovery with Continuous Additive Noise Models". Apr. 6, 2014. arXiv: 1309.6779 [stat] (cit. on pp. 40–44, 52, 200, 213).

Peters, Jonas, Peter Bühlmann, and Nicolai Meinshausen. "Causal Inference by Using Invariant Prediction: Identification and Confidence Intervals". *Journal of the Royal Statistical Society: Series B (Statistical Methodology)* 78.5 (2016), pp. 947–1012. ISSN: 1467-9868. DOI: 10.1111/rssb.12167 (cit. on pp. 40–42, 200).

Peters, Jonas, Dominik Janzing, and Bernhard Schölkopf. *Elements of Causal Inference: Foundations and Learning Algorithms*. Red. by Francis Bach. Adaptive Computation and Machine Learning Series. Cambridge, MA, USA: MIT Press, Nov. 29, 2017. 288 pp. ISBN: 978-0-262-03731-0 (cit. on pp. 28, 84).

Pfister, Niklas, Sebastian Weichwald, Peter Bühlmann, and Bernhard Schölkopf. "Robustifying Independent Component Analysis by Adjusting for Group-Wise Stationary Noise". *Journal of Machine Learning Research* 20.147 (2019), pp. 1–50. arXiv: 1806.01094 (cit. on p. 32).

Pham, Dinh-Tuan and J.-F. Cardoso. "Blind Separation of Instantaneous Mixtures of Nonstationary Sources". *IEEE Transactions on Signal Processing* 49.9 (2001), pp. 1837–1848 (cit. on p. 32).

Plumbley, Mark D. "Algorithms for Nonnegative Independent Component Analysis". *IEEE Transactions on Neural Networks* 14.3 (2003), pp. 534–543 (cit. on p. 32).

Podosinnikova, Anastasia, Francis Bach, and Simon Lacoste-Julien. "Rethinking LDA: Moment Matching for Discrete ICA". *Advances in Neural Information Processing Systems 28*. Ed. by C. Cortes, N. D. Lawrence, D. D. Lee, M. Sugiyama, and R. Garnett. Curran Associates, Inc., 2015, pp. 514–522 (cit. on p. 32).

Podosinnikova, Anastasia, Amelia Perry, Alexander Wein, Francis Bach, Alexandre d'Aspremont, and David Sontag. "Overcomplete Independent Component Analysis via SDP". Jan. 24, 2019. arXiv: 1901.08334 [cs, stat] (cit. on p. 32).

Poldrack, Russell A., Timothy O. Laumann, Oluwasanmi Koyejo, Brenda Gregory, Ashleigh Hover, Mei-Yen Chen, Krzysztof J. Gorgolewski, Jeffrey Luci, Sung Jun Joo, Ryan L. Boyd, Scott Hunicke-Smith, Zack Booth

Simpson, Thomas Caven, Vanessa Sochat, James M. Shine, Evan Gordon, Abraham Z. Snyder, Babatunde Adeyemo, Steven E. Petersen, David C. Glahn, D. Reese Mckay, Joanne E. Curran, Harald H. H. Göring, Melanie A. Carless, John Blangero, Robert Dougherty, Alexander Leemans, Daniel A. Handwerker, Laurie Frick, Edward M. Marcotte, and Jeanette A. Mumford. "Long-Term Neural and Physiological Phenotyping of a Single Human". *Nature Communications* 6.1 (1 Dec. 9, 2015), p. 8885. ISSN: 2041-1723. DOI: 10.1038/ncomms9885 (cit. on pp. 87, 107).

Rezende, Danilo Jimenez, Shakir Mohamed, and Daan Wierstra. "Stochastic Backpropagation and Approximate Inference in Deep Generative Models". Jan. 16, 2014. arXiv: 1401.4082 [cs, stat] (cit. on pp. 28, 29, 62, 64).

Rezende, Danilo Jimenez and Shakir Mohamed. "Variational Inference with Normalizing Flows". *International Conference on Machine Learning*. PMLR, May 21, 2015, pp. 1530–1538. arXiv: 1505.05770 (cit. on pp. 45, 68, 130, 165, 201).

Rolinek, Michal, Dominik Zietlow, and Georg Martius. "Variational Autoencoders Pursue PCA Directions (by Accident)". Dec. 17, 2018. arXiv: 1812.06775 [cs, stat] (cit. on pp. 31, 71).

Saremi, Saeed, Arash Mehrjou, Bernhard Schölkopf, and Aapo Hyvärinen. "Deep Energy Estimator Networks". 2018. arXiv: 1805.08306 (cit. on p. 167).

Schmidhuber, Jürgen, Martin Eldracher, and Bernhard Foltin. "Semilinear Predictability Minimization Produces Well-Known Feature Detectors". *Neural Computation* 8.4 (May 1996), pp. 773–786. ISSN: 0899-7667. DOI: 10.1162/neco.1996.8.4.773 (cit. on p. 28).

Schölkopf, Bernhard. "Causality for Machine Learning" (Nov. 24, 2019) (cit. on p. 38).

Seide, Frank, Gang Li, and Dong Yu. "Conversational Speech Transcription Using Context-Dependent Deep Neural Networks". *Twelfth Annual Conference of the International Speech Communication Association*. 2011 (cit. on p. 28).

Shimizu, Shohei, Patrik O. Hoyer, Aapo Hyvärinen, and Antti Kerminen. "A Linear Non-Gaussian Acyclic Model for Causal Discovery". *Journal of Machine Learning Research* 7 (Oct 2006), pp. 2003–2030. ISSN: ISSN 1533-7928 (cit. on pp. 40, 44, 45, 84, 200, 221).

Shimizu, Shohei, Takanori Inazumi, Yasuhiro Sogawa, Aapo Hyvärinen, Yoshinobu Kawahara, Takashi Washio, Patrik O. Hoyer, and Kenneth Bollen. "DirectLiNGAM: A Direct Method for Learning a Linear Non-Gaussian Structural Equation Model". *Journal of Machine Learning Research* 12 (Apr 2011), pp. 1225–1248 (cit. on pp. 40, 44, 200).

Song, Yang and Stefano Ermon. "Generative Modeling by Estimating Gradients of the Data Distribution" (July 12, 2019) (cit. on pp. 150, 152, 167).

Sorrenson, Peter, Carsten Rother, and Ullrich Köthe. "Disentanglement by Nonlinear ICA with General Incompressible-flow Networks (GIN)". Jan. 14, 2020. arXiv: 2001.04872 [cs, stat] (cit. on pp. 68, 130, 137).

Spearman, C. "The Proof and Measurement of Association between Two Things". *The American Journal of Psychology* 15.1 (1904), pp. 72–101. ISSN: 0002-9556. DOI: 10.2307/1412159. JSTOR: 1412159 (cit. on p. 79).

Spirtes, Peter, Clark N. Glymour, Richard Scheines, and David Heckerman. *Causation, Prediction, and Search*. MIT press, 2000 (cit. on pp. 38, 200, 209).

Spirtes, Peter and Kun Zhang. "Causal Discovery and Inference: Concepts and Recent Methodological Advances". *Applied Informatics* 3.1 (Feb. 18, 2016), p. 3. ISSN: 2196-0089. DOI: 10.1186/s40535-016-0018-x (cit. on pp. 39, 200).

Sprekeler, Henning, Tiziano Zito, and Laurenz Wiskott. "An Extension of Slow Feature Analysis for Nonlinear Blind Source Separation". *The Journal of Machine Learning Research* 15.1 (2014), pp. 921–947 (cit. on p. 35).

Sriperumbudur, Bharath, Kenji Fukumizu, Arthur Gretton, Aapo Hyvärinen, and Revant Kumar. "Density Estimation in Infinite Dimensional Exponential Families". *Journal of Machine Learning Research* 18.57 (2017), pp. 1–59. ISSN: 1533-7928 (cit. on p. 67).

Sugiyama, Masashi, Taiji Suzuki, and Takafumi Kanamori. "Density-Ratio Matching under the Bregman Divergence: A Unified Framework of Density-Ratio Estimation". *Annals of the Institute of Statistical Mathematics* 64.5 (2012), pp. 1009–1044 (cit. on p. 31).

Taleb, A. and C. Jutten. "Source Separation in Post-Nonlinear Mixtures". *IEEE Transactions on Signal Processing* 47.10 (Oct. 1999), pp. 2807–2820. ISSN: 1941-0476. DOI: 10.1109/78.790661 (cit. on p. 35).

Teshima, Takeshi, Issei Sato, and Masashi Sugiyama. “Few-Shot Domain Adaptation by Causal Mechanism Transfer”. 2020. arXiv: 2002.03497 (cit. on p. 38).

Thompson, William Hedley, Remya Nair, Hiroyuki Oya, Oscar Esteban, James M. Shine, Christopher Petkov, Russell A. Poldrack, Matthew Howard, and Ralph Adolphs. “Human Es-fMRI Resource: Concurrent Deep-Brain Stimulation and Whole-Brain Functional MRI”. *bioRxiv* (May 20, 2020), p. 2020.05.18.102657. DOI: 10.1101/2020.05.18.102657 (cit. on p. 219).

Tipping, Michael E. and Christopher M. Bishop. “Probabilistic Principal Component Analysis”. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)* 61.3 (1999), pp. 611–622 (cit. on p. 28).

Tucker, George, Dieterich Lawson, Shixiang Gu, and Chris J. Maddison. “Doubly Reparameterized Gradient Estimators for Monte Carlo Objectives”. 2018. arXiv: 1810.04152 (cit. on p. 62).

Uehara, Masatoshi, Takafumi Kanamori, Takashi Takenouchi, and Takeru Matsuda. “A Unified Statistically Efficient Estimation Framework for Unnormalized Models”. *International Conference on Artificial Intelligence and Statistics*. International Conference on Artificial Intelligence and Statistics. PMLR, June 3, 2020, pp. 809–819 (cit. on p. 132).

Van der Laan, Mark J. and Sherri Rose. *Targeted Learning in Data Science: Causal Inference for Complex Longitudinal Studies*. Springer, 2018 (cit. on p. 38).

Vincent, Pascal. “A Connection between Score Matching and Denoising Autoencoders”. *Neural computation* 23.7 (2011), pp. 1661–1674 (cit. on pp. 133, 165–167).

Vogt, Brent A. “Cingulate Cortex in the Three Limbic Subsystems”. *Handbook of clinical neurology* 160 (2019), pp. 39–51 (cit. on p. 236).

Wang, Mei and Weihong Deng. “Deep Visual Domain Adaptation: A Survey”. *Neurocomputing* 312 (2018), pp. 135–153 (cit. on p. 28).

Watanabe, Satosi. “Information Theoretical Analysis of Multivariate Correlation”. *IBM Journal of research and development* 4.1 (1960), pp. 66–82 (cit. on p. 47).

Wehenkel, Antoine and Gilles Louppe. “Graphical Normalizing Flows”. Oct. 13, 2020. arXiv: 2006.02548 [cs, stat] (cit. on p. 221).

Wiatowski, Thomas and Helmut Bölcskei. "A Mathematical Theory of Deep Convolutional Neural Networks for Feature Extraction". Oct. 24, 2017. arXiv: 1512.06293 [cs, math, stat] (cit. on p. 240).

Wilks, S. S. "The Large-Sample Distribution of the Likelihood Ratio for Testing Composite Hypotheses". *The Annals of Mathematical Statistics* 9.1 (Mar. 1938), pp. 60–62. ISSN: 0003-4851, 2168-8990. DOI: 10.1214/aoms/1177732360 (cit. on p. 25).

Willetts, Matthew and Brooks Paige. "I Don't Need $\mathbf{u}$: Identifiable Non-Linear ICA Without Side Information". June 9, 2021. arXiv: 2106.05238 [cs, stat] (cit. on p. 239).

Wu, Pengzhou and Kenji Fukumizu. "Causal Mosaic: Cause-Effect Inference via Nonlinear ICA and Ensemble Method". Jan. 7, 2020. arXiv: 2001.01894 [cs, stat] (cit. on p. 38).

Yanardag, Pinar and S. V. N. Vishwanathan. "Deep Graph Kernels". *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. 2015, pp. 1365–1374 (cit. on p. 27).

Yang, Mengyue, Furui Liu, Zhitang Chen, Xinwei Shen, Jianye Hao, and Jun Wang. "CausalVAE: Disentangled Representation Learning via Neural Structural Causal Models". July 1, 2020. arXiv: 2004.08697 [cs, stat] (cit. on p. 38).

Zhang, Kun and Aapo Hyvärinen. "On the Identifiability of the Post-Nonlinear Causal Model". *25th Conference on Uncertainty in Artificial Intelligence, UAI 2009*. Vol. 35. 2009 (cit. on pp. 40, 41, 200, 209, 220, 240).

Zhang, Kun, Jiji Zhang, and Bernhard Schölkopf. "Distinguishing Cause from Effect Based on Exogeneity". Fifteenth Conference on Theoretical Aspects of Rationality and Knowledge (TARK), 2015. Carnegie Mellon University, Apr. 22, 2015, pp. 261–271. arXiv: 1504.05651 [cs, stat] (cit. on pp. 40, 203).

Zhang, Kun, Zhikun Wang, Jiji Zhang, and Bernhard Schölkopf. "On Estimation of Functional Causal Models: General Results and Application to the Post-Nonlinear Causal Model". *ACM Transactions on Intelligent Systems and Technology* 7.2 (Dec. 17, 2015), 13:1–13:22. ISSN: 2157-6904. DOI: 10.1145/2700476 (cit. on p. 200).

Zhang, Kun, Biwei Huang, Jiji Zhang, Clark Glymour, and Bernhard Schölkopf. "Causal Discovery from Nonstationary/Heterogeneous Data: Skeleton Esti-

mation and Orientation Determination". *Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence*. Twenty-Sixth International Joint Conference on Artificial Intelligence. Melbourne, Australia: International Joint Conferences on Artificial Intelligence Organization, Aug. 2017, pp. 1347–1353. ISBN: 978-0-9992411-0-3. DOI: 10.24963/ijcai.2017/187 (cit. on pp. 40, 200).

Zhao, Shengjia, Jiaming Song, and Stefano Ermon. "InfoVAE: Information Maximizing Variational Autoencoders". 2017. arXiv: 1706.02262 (cit. on pp. 29, 31).

Zheng, Xun, Bryon Aragam, Pradeep K. Ravikumar, and Eric P. Xing. "DAGs with NO TEARS: Continuous Optimization for Structure Learning". *Advances in Neural Information Processing Systems*. 2018, pp. 9472–9483 (cit. on pp. 40, 44, 200).