
ICE-BeeM: Identifiable Conditional Energy-Based Deep Models Based on Nonlinear ICA

Ilyes Khemakhem¹

Ricardo P. Monti¹

Diederik P. Kingma²

Aapo Hyvärinen^{3,4}

¹ Gatsby Unit, University College London

² Google Research

³ Université Paris-Saclay, Inria

⁴ University of Helsinki

ilyesk@gatsby.ucl.ac.uk

Abstract

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 for the estimation of 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.

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 very difficult 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, in the limit of infinite data, multiple representation functions can fit the data, then the true representation function is unidentifiable.

Until recently (Hyvärinen and Morioka, 2016, 2017), results relating to identifiability of (explicit and implicit) latent variable models were mainly constrained to linear models (*e.g.*, as in linear ICA), as it was acknowledged that the flexibility of nonlinear mappings could yield arbitrary latent variables which fulfill 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. (2020) explicated 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.

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 modeled densities to be normalized nor easy to sample from. In fact, the energy model we suggest will have universal approximation capabilities. The energy function we will consider is defined in two steps: we learn two feature extractors, parameterized 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 modeled conditional densities are defined to be the exponential of the negative energy function.

A first important contribution of this paper 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 the linear transformation is essentially a permutation. These conditions are functional, *i.e.* they abstract away the architecture of the networks. As a concrete example, we provide a neural network architecture based on fully connected layers, for which the functional conditions hold, and is thus identifiable. Moreover, we do not make any assumptions on the distributions of the learned representations. 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.

Our second contribution is to develop a framework we call Independently Modulated Component Analysis (IMCA): a deep latent variable model where the latents are non-independent (thus generalizing nonlinear ICA), with an arbitrary global dependency structure. Nonlinear ICA research has formalized the trade-off between expressivity of the mapping between latents to observations (from linear to nonlinear) and distributional assumptions over latent variables (from independent to conditionally independent given auxiliary variables). However, the need for (conditional) independence in order to obtain identifiability results may sometimes be seen as a limitation, for example in the context of learning disentangled representations. Therefore, it would be important to relax the assumption of independence while maintaining identifiability. This was achieved before in the linear case (Monti and Hyvärinen, 2018; Hyvärinen and Hurri, 2004), and we show how it may be achieved in the nonlinear setting. We show how our ICE-BeeM can estimate this generative model, thus connecting both the generative and non-generative views.

Finally, we show empirically that ICE-BeeM learns identifiable representations from real-world image datasets. As a further, rather different application of our results, we show how identifiability of ICE-BeeM can be leveraged for transfer learning and semi-supervised learning. In fact, we believe that the identifiability results are generally important for principled application of EBMs, whether for the purposes of disentanglement or otherwise.

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 C.

2.1 Model definition

We collect a dataset of observations 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 $\mathbf{f}_\theta(\mathbf{x}) \in \mathbb{R}^{d_z}$ and $\mathbf{g}_\theta(\mathbf{y}) \in \mathbb{R}^{d_z}$, which we parameterize by neural networks, and θ is the vector of weights and biases. To alleviate notations, we will drop θ when it's clear which quantities we refer to. These feature extractors are used to define the conditional energy function $\mathcal{E}_\theta(\mathbf{x}|\mathbf{y}) = \mathbf{f}_\theta(\mathbf{x})^T \mathbf{g}_\theta(\mathbf{y})$.

The parameter θ lives in the space Θ which is defined such that the normalizing constant $Z(\mathbf{y}; \theta) = \int_{\mathcal{X}} \exp(-\mathcal{E}_\theta(\mathbf{x}|\mathbf{y})) d\mathbf{x} < \infty$ is finite. Our family of conditional energy-based models has the form:

$$p_\theta(\mathbf{x}|\mathbf{y}) = \frac{\exp(-\mathbf{f}_\theta(\mathbf{x})^T \mathbf{g}_\theta(\mathbf{y}))}{Z(\mathbf{y}; \theta)} \quad (1)$$

As we will see later, this choice of 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 work, we will use Flow Contrastive Estimation (Gao et al., 2019) and Denoising Score Matching (Vincent, 2011), which are discussed and extended to the conditional case in Appendix B.

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 (1) from the same dataset, we want the learned features to be very similar.

This similarity between representations is better expressed as equivalence relations on the parameters θ 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 by Khemakhem et al. (2020) to address the fact that there typically exist many choices of neural network parameters θ that map to the same point in function-space. In our case, it is given by the following definitions:

Definition 1 (Weak identifiability). Let \sim_w^f and \sim_w^g be equivalence relations on Θ defined as:

$$\begin{aligned}\theta \sim_w^f \theta' &\Leftrightarrow \forall \mathbf{x}, \mathbf{f}_\theta(\mathbf{x}) = \mathbf{A}\mathbf{f}_{\theta'}(\mathbf{x}) + \mathbf{c} \\ \theta \sim_w^g \theta' &\Leftrightarrow \forall \mathbf{y}, \mathbf{g}_\theta(\mathbf{y}) = \mathbf{B}\mathbf{g}_{\theta'}(\mathbf{y}) + \mathbf{e}\end{aligned}\tag{2}$$

where \mathbf{A} and \mathbf{B} are $(d_z \times d_z)$ -matrices of rank at least $\min(d_z, d_x)$ and $\min(d_z, d_y)$ respectively, and \mathbf{c} and \mathbf{e} are vectors.

Definition 2 (Strong identifiability). Let \sim_s^f and \sim_s^g be the equivalence relations on Θ defined as:

$$\begin{aligned}\theta \sim_s^f \theta' &\Leftrightarrow \forall i, \forall \mathbf{x}, f_{i,\theta}(\mathbf{x}) = a_i f_{\sigma(i),\theta'}(\mathbf{x}) + c_i \\ \theta \sim_s^g \theta' &\Leftrightarrow \forall i, \forall \mathbf{x}, g_{i,\theta}(\mathbf{x}) = b_i g_{\gamma(i),\theta'}(\mathbf{x}) + e_i\end{aligned}\tag{3}$$

where σ and γ are permutations of $\llbracket 1, n \rrbracket$, a_i and b_i are non-zero scalars and c_i and e_i are scalars.

Two parameters are thus considered equivalent if they parameterize two feature extractors that are equal up to a linear transformation (2) or a scaled permutation (3). The subscripts w and s stand for weak and strong, respectively. Special cases are discussed in Appendix C.1.

Identifiability in the context of probability densities modeled by neural networks can be seen as a study of degeneracy of the networks. In applications where the representations are used in a downstream classification task, the weak identifiability (2) 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 (3), 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. Were one to learn the latents only up to a rotation, such causal analysis method would not work at all.

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 2.3. 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. Intuition behind the conditions, as well as a proof under milder assumptions, can be found in Appendix C.2.

Theorem 1. Let \sim_w^f and \sim_w^g be the equivalence relations in (2). Assume that for any choice of parameter θ :

1. The feature extractor \mathbf{f}_θ is differentiable, and its Jacobian $\mathbf{J}_{\mathbf{f}_\theta}$ is full rank.¹

¹Its rank is equal to its smaller dimension.

2. There exist $d_z + 1$ points $\mathbf{y}^0, \dots, \mathbf{y}^{d_z}$ such that the matrix $\mathbf{R}_\theta = (\mathbf{g}_\theta(\mathbf{y}^1) - \mathbf{g}_\theta(\mathbf{y}^0), \dots, \mathbf{g}_\theta(\mathbf{y}^{d_z}) - \mathbf{g}_\theta(\mathbf{y}^0))$ of size $d_z \times d_z$ is invertible.

then $p_\theta(\mathbf{x}|\mathbf{y}) = p_{\theta'}(\mathbf{x}|\mathbf{y}) \implies \theta \sim_w^f \theta'$.

With \mathbf{f}_θ and \mathbf{g}_θ switched, the same conclusion applies to \mathbf{g}_θ : $p_\theta(\mathbf{x}|\mathbf{y}) = p_{\theta'}(\mathbf{x}|\mathbf{y}) \implies \theta \sim_w^g \theta'$.

Finally, if both assumptions 1 and 2 are satisfied by both feature extractors \mathbf{f}_θ and \mathbf{g}_θ , then the matrices \mathbf{A} and \mathbf{B} in (2) have full row rank equal to d_z .

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^f and \sim_s^g in (3). 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 is based on restricting the feature extractor \mathbf{f} to be non-negative. It will induce constraints on the matrix \mathbf{A} defining the equivalence relation \sim_w^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 plane to itself is the identity.

The second alteration is based on augmenting \mathbf{f} by its square, effectively resulting in the $2d_z$ -dimensional feature extractor $\tilde{\mathbf{f}}(\mathbf{x}) = (\dots, f_i(\mathbf{x}), f_i^2(\mathbf{x}), \dots) \in \mathbb{R}^{2d_z}$. 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 $\tilde{\mathcal{E}}(\mathbf{x}|\mathbf{y}) = \tilde{\mathbf{f}}(\mathbf{x})^T \tilde{\mathbf{g}}(\mathbf{y})$. The advantage of this approach is that it doesn't require the feature extractors to be positive. However, it makes the effective size of the feature extractor equal to $2d_z$.

Identifiability results derived from these two alterations are summarized by the following theorem.

Theorem 2. Assume that $d_z \leq d_x$ and that the assumptions of Theorem 1 hold. Further assume that, for any choice of parameter θ , either one of the following conditions hold:

3. The feature extractor \mathbf{f}_θ is surjective, and its image is $\mathbb{R}_+^{d_x}$.
4. The feature extractor \mathbf{f}_θ is differentiable and surjective, its Jacobian $\mathbf{J}_{\mathbf{f}_\theta}$ is full rank; there exist $2d_z + 1$ points $\mathbf{y}^0, \dots, \mathbf{y}^{2d_z}$ such that the matrix $\tilde{\mathbf{R}}_\theta = (\tilde{\mathbf{g}}_\theta(\mathbf{y}^1) - \tilde{\mathbf{g}}_\theta(\mathbf{y}^0), \dots, \tilde{\mathbf{g}}_\theta(\mathbf{y}^{2d_z}) - \tilde{\mathbf{g}}_\theta(\mathbf{y}^0))$ of size $2d_z \times 2d_z$ is invertible; and we use the augmented energy function $\tilde{\mathcal{E}}(\mathbf{x}|\mathbf{y})$ in the definition of the model.

Then $p_\theta(\mathbf{x}|\mathbf{y}) = p_{\theta'}(\mathbf{x}|\mathbf{y}) \implies \theta \sim_s^f \theta'$ where \sim_s^f is defined in (3).

A more general form of the Theorem is provided in Appendix C.5. This theorem is fundamental as it proves very strong identifiability results for a conditional deep 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., 2020; Sorrenson et al., 2020). Most importantly, we do not make any assumption on the distribution of the latent features.

2.3 An identifiable neural network architecture

In this section, we give a concrete example of a neural network architecture that satisfies the functional assumptions of Theorem 1. We suppose that each of the networks \mathbf{f} and \mathbf{g} are parameterized 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, $\forall l \in \llbracket 1, L-1 \rrbracket$.²
- (b.) The weight matrices \mathbf{W}_l are full rank (its rank is equal to its smaller dimension), $\forall l \in \llbracket 1, L \rrbracket$.
- (c.) The row dimension of the weight matrices are either monotonically increasing or decreasing: $d_l \geq d_{l+1}, \forall l \in \llbracket 0, L-1 \rrbracket$ or $d_l \leq d_{l+1}, \forall l \in \llbracket 0, L-1 \rrbracket$.

²A LeakyReLU has the form $h_l(x) = \max(0, x) + \alpha \min(0, x)$, $\alpha \in (0, 1)$.

(d.) All submatrices of \mathbf{W}_l of size $d_l \times d_l$ are invertible if $d_l < d_{l+1}, \forall l \in \llbracket 0, L-1 \rrbracket$.

This architecture satisfies the assumptions of Theorems 1 and 2, as is stated by the propositions below.

Proposition 1. *Consider an MLP \mathbf{f} whose architecture satisfies assumptions (a.), (b.) and (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 to the output of the network, then \mathbf{f} satisfies Assumption 3.*

Proposition 2. *Consider a nonlinear MLP \mathbf{g} whose architecture satisfies assumptions (a.), (b.), and (d.).³ Then, \mathbf{g} satisfies Assumptions 2 and 4.*

While assumptions (a.)-(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 1 and 2. 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 initialization of floating point parameters, which are then optimized with stochastic updates (SGD), will result in weight matrices that are almost certainly full rank.

2.4 Universal approximation capability

With such a potentially overcomplete network, we can further achieve universal approximation of the data distribution. It might initially seem that this is an impossible endeavor 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:

Theorem 3. *Let $p(\mathbf{x}|\mathbf{y})$ be a conditional probability density. Assume that \mathcal{X} and \mathcal{Y} are compact Hausdorff spaces, and that $p(\mathbf{x}|\mathbf{y}) > 0$ almost surely $\forall (\mathbf{x}, \mathbf{y}) \in \mathcal{X} \times \mathcal{Y}$. Then for each $\varepsilon > 0$, there exists $(\boldsymbol{\theta}, d_z) \in \Theta \times \mathbb{N}$, where d_z is the dimension of the feature extractor, 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 is capable of approximating 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 1 covers the overcomplete case as well, and highlights the importance of our contribution in comparison to previous identifiable deep models.

3 Independently modulated component analysis

Next, we show how ICE-BeeM relates to a generative, latent variable model. We develop here a novel framework that generalizes nonlinear ICA to non-independent latent variables, and show how we can use our energy model to estimate them.

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:

$$\mathbf{z} \sim p(\mathbf{z}|\mathbf{y}) \quad , \quad \mathbf{x} = \mathbf{h}(\mathbf{z}) \quad (4)$$

We will suppose here that $d_x = d_z = d$. The proofs, as well as an extension to $d_z < d_x$, can be found in Appendix D. The main modeling assumption we make on the latent variable is that its density has the following form:

$$p(\mathbf{z}|\mathbf{y}) = \mu(\mathbf{z}) e^{\sum_{i=1}^{d_z} \mathbf{T}_i(z_i)^T \boldsymbol{\lambda}_i(\mathbf{y}) - \Gamma(\mathbf{y})} \quad (5)$$

where $\mu(\mathbf{z})$ 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) and (5) together define a nonparametric model with parameters $(\mathbf{h}, \mathbf{T}, \boldsymbol{\lambda}, \mu)$. For the special case $\mu(\mathbf{z}) = \prod_i \mu_i(z_i)$, the distribution of \mathbf{z} factorizes across dimensions, and the components

³The particular case of linear MLPs is discussed in Appendix C.4.

z_i are independent. Then the generative model gives rise to a nonlinear ICA model, and it was studied to a great depth by Khemakhem et al. (2020).

We propose to generalize such earlier models by allowing for an arbitrary base measure $\mu(\mathbf{z})$, i.e. the components of the latent variable are no longer independent, as μ doesn't necessarily factorize across dimensions. We call this new framework *Independently Modulated Component Analysis* (IMCA). We show in Appendix E 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.

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 $\tilde{\mathbf{f}}(\mathbf{x}) = (\mathbf{H}_1(f_1(\mathbf{x})), \dots, \mathbf{H}_d(f_d(\mathbf{x})))$. In Section 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 $\mathbf{f}_\theta(\mathbf{x})$, 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 (5), as formalized by the following theorem.

Theorem 4. *Assume:*

- (i) *The observed data follows the exponential IMCA model of equations (4)-(5).*
- (ii) *The mixing function \mathbf{h} is a \mathcal{D}^2 -diffeomorphism.⁴*
- (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. 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 $\mathbf{y}^0, \dots, \mathbf{y}^k$ such that the matrix $\mathbf{L} = (\boldsymbol{\lambda}(\mathbf{y}_1) - \boldsymbol{\lambda}(\mathbf{y}_0), \dots, \boldsymbol{\lambda}(\mathbf{y}_k) - \boldsymbol{\lambda}(\mathbf{y}_0))$ 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 (1) 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 k -dimensional, and their dimensions to be chosen from $(\dim(\mathbf{T}_1), \dots, \dim(\mathbf{T}_d))$ without replacement.*

Then, in the limit of infinite data, $\mathbf{H}_i(f_i(\mathbf{x})) = \mathbf{A}_i \mathbf{T}_{\gamma(i)}(z_{\gamma(i)}) + \mathbf{b}_i$ where γ is a permutation of $\llbracket 1, d \rrbracket$ 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 point-wise nonlinearities.

4 Relation to previous work on nonlinear ICA

Our results greatly extend existing identifiability results and models. The closest latent variable model identifiability theory to ours is that of nonlinear ICA theory (Hyvärinen and Morioka, 2016; Hyvärinen et al., 2019; Khemakhem et al., 2020). 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.

On this front, our first contribution was to identify that conditional independence is not necessary for identifiability, and to propose the more general IMCA framework. Our proofs extend previous ones to the non-independent case, and are the most general to date, even considering linear ICA theory. In fact, as a second contribution, our conditional EBM generalizes previous results by completely dropping any distributional assumptions on the representations—which are ubiquitous in the latent variable case.

Third, most of our theoretical results hold for overcomplete representations, which means that unlike the earlier works cited above, our model can be shown to even have universal approximation capabilities. Fourth, while recent identifiability theory focused on providing functional conditions for identifiability, such work is a bit removed from the reality of neural network training. Our results on network architectures are the first step towards bridging the gap between theory and practice.

⁴That is: invertible, all second order cross-derivatives of the function and its inverse exist.

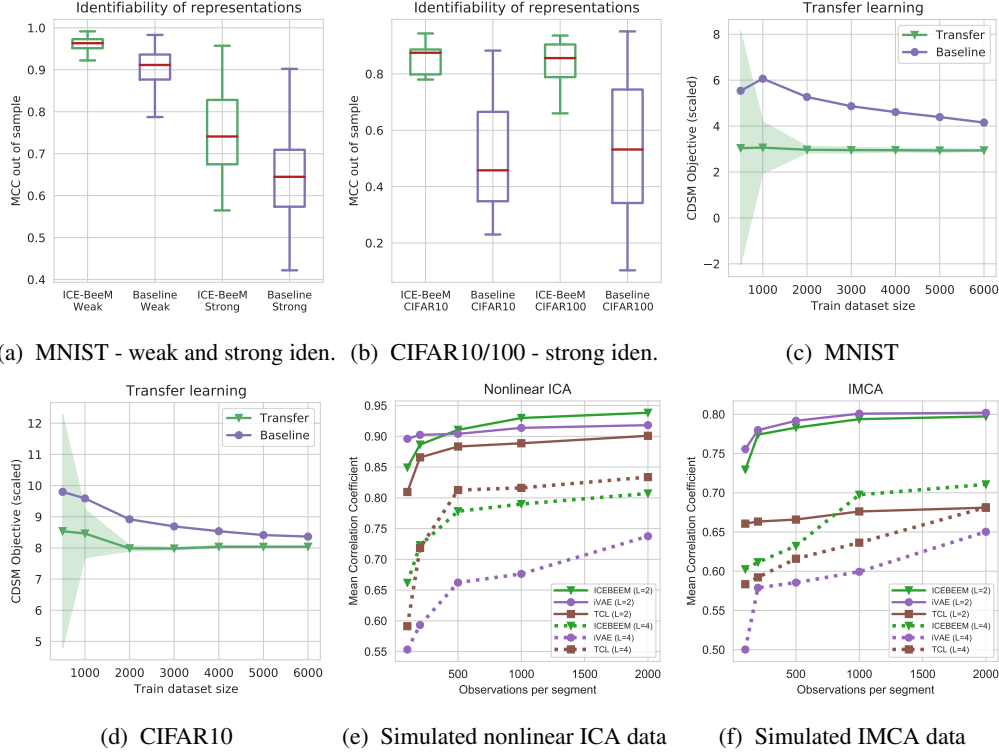


Figure 1: (a) – (b) Quantifying the identifiability of learnt representations using MCC (higher is better). (c) – (d) Transfer learning onto unseen classes using denoising score matching objective (lower is better). (e) – (f) Simulations on artificial nonlinear ICA/IMCA data (higher is better).

5 Experiments

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 initializations, 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. The different architectures used throughout these experiments are described in Appendix A.1. Code for reproducibility is available [here](#).

Quantifying identifiability We start by empirically validating Theorems 1 and 2 on image datasets. 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 initializations. We subsequently compute the mean correlation coefficient (MCC, see Appendix A.2) between learned representations obtained via distinct random initializations; 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 (2). Throughout experiments, we employ CCA to learn the linear mapping \mathbf{A} . However, our main interest is studying the strong identifiability of EBM architectures, defined in (3). 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 distinct image datasets: MNIST, CIFAR 10 and 100. For each dataset, we train models using 20 distinct random initializations and compare inferred representations. Conditional denoising score matching (CDSM, see Appendix B.1) was employed to train all networks. Results presented in

Table 1: (a) Transfer learning. (b) Semi-supervised learning.

(a) CDSM objective (lower is better)					(b) Classification Accuracy (higher is better)		
Dataset	$\mathbf{f} \cdot \mathbf{g}_\theta$	$\mathbf{f} \cdot \mathbf{1}$	$\mathbf{f}_\theta \cdot \mathbf{g}_\theta$	$\mathbf{f}_\theta \cdot \mathbf{1}$	Dataset	ICE-BeeM	Uncond. EBM
MNIST	2.95	23.43	4.22	3.64	FMNIST	77.07 \pm 1.39	56.33 \pm 3.18
CIFAR10	8.03	23.08	8.37	8.16	CIFAR10	64.42 \pm 1.09	51.88 \pm 1.33

Figures [1a-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 A.3 for further details and experiments.

Application to transfer learning Second, we present an application of ICE-BeeM to transfer learning. We suppose that the auxiliary variable $y \in \mathbb{R}$ is the index of a dataset or a task. We propose an intuitively appealing approach, where we approximate the unnormalized log-pdf in y -th dataset $\log p(\mathbf{x}|y)$ by a linear combination of a learned "basis" functions $f_{i,\theta}$ as $\log p(\mathbf{x}; y) + \log Z(\theta) \approx \sum_{i=1}^k g_i(y) f_{i,\theta}(\mathbf{x})$, where the $g_i(y)$ are scalar parameters that act as coefficients in the basis $(f_{i,\theta})$. For a new unseen dataset y_{new} , reducing the transfer learning to the estimation of the $g_i(y_{\text{new}})$ clearly requires that we have estimated the true f_i , at least up to a linear transformation. This is exactly what can be achieved by ICE-BeeM based on weak identifiability. 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 $\mathbf{g}_\theta(y_{\text{new}})$ for the unseen classes (we denote this by $\mathbf{f} \cdot \mathbf{g}_\theta$; unseen classes are 8 & 9). We allow \mathbf{g}_θ to be parameterized 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}_θ and \mathbf{g}_θ are trained directly on data from unseen classes only (*i.e.* there is no transfer learning—denoted $\mathbf{f}_\theta \cdot \mathbf{g}_\theta$). Results are presented in Figures [1c] and [1d] 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 $\mathbf{f}_\theta \cdot \mathbf{1}$). The average CDSM scores are reported in Table [1a], where the transfer learning with an identifiable EBM (*i.e.*, using ICE-BeeM) performs best. See Appendix A.4 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.

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}_θ to classify unseen classes 8-9 using a logistic regression. In both cases, training proceeded via CDSM. Table [1b] 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 A.5 for further details and experiments.

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., 2020) 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 to be a segment index. First, we let the data follow a nonlinear ICA model, which is a special case of equation (4) 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 $\lambda(\mathbf{y})$ 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 $\Sigma_0 \in \mathbb{R}^{d \times d}$, such that $\mu(\mathbf{z}) \propto e^{-0.5 \mathbf{z}^T \Sigma_0^{-1} \mathbf{z}}$. The covariance matrix of each segment is now equal to $\Sigma(\mathbf{y}) = (\Sigma_0^{-1} + \text{diag}(\lambda(\mathbf{y})))^{-1}$, meaning the latent variables are no longer conditionally independent. In both cases, a randomly initialized neural network with 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 A.6.

In the case of ICE-BeeM, conditional flow contrastive estimation (CFCE, see Appendix B.2) was employed to estimate network parameters. To evaluate the performance of the method, we compute the mean correlation coefficient (MCC, see Appendix A.2) between the true latent variables and the recovered latents estimated by all three methods. Results for nonlinear ICA are provided in Figure [1e], where we note that ICE-BeeM performs competitively with respect to both iVAE and TCL. We note that as the depth of the mixing network, L , increases the performance of all methods decreases. Results for IMCA are provided in Figure [1f] 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.

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.

We further prove a fundamental connection between EBMs and latent variable models, showing that ICE-BeeM is able to estimate nonlinear ICA, as a special case. In fact, it can even estimate a generalized version where the components do not need to be independent: they only need to be independently modulated by another variable such as a time index, history or noisy labels.

Empirically, we showed on real-world image datasets that our 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.

Identifiability is fundamental for meaningful and principled disentanglement; it is necessary to make any interpretation of the features meaningful; it is also crucial in such applications as causal discovery (Monti et al., 2019) and transfer learning. The present results go further than any identifiability results hitherto and extend them to the EBM framework. We believe this paves the way for many new applications of EBMs, by giving them a theoretically sound basis.

Broader Impact

This work is mainly theoretical, and aims to provide theoretical guarantees for the identifiability of a large family of deep models. Identifiability is very important, as it is key for reproducible science and interpretable results. For instance, if the networks behind search engines were identifiable, then their results would be consistent for most users. In addition, using perfectly identifiable networks in real life applications eliminates the randomness and arbitrariness of the system, and gives more control to the operator.

In general, identifiability is a desirable property. The system we develop here does not make any decisions, and thus can not exhibit any bias. Our theoretical guarantees abstract away the nature of the data and the practical implementation. Therefore, our work doesn't encourage the use of biased data or networks with potentially dangerous consequences.

Acknowledgments and Disclosure of Funding

I.K. and R.P.M. were supported by the Gatsby Charitable Foundation. A.H. was supported by a Fellowship from CIFAR, and from the DATAIA convergence institute as part of the "Programme d'Investissement d'Avenir", (ANR-17-CONV-0003) operated by Inria.

References

- Arbel, M., Sutherland, D., Bińkowski, M., and Gretton, A. (2018). On gradient regularizers for mmd gans. In *Advances in Neural Information Processing Systems*, pages 6700–6710.
- Brosowski, B. and Deutsch, F. (1981). An elementary proof of the stone-weierstrass theorem. *Proceedings of the American Mathematical Society*, pages 89–92.
- Ceylan, C. and Gutmann, M. U. (2018). Conditional Noise-Contrastive Estimation of Unnormalised Models. *arXiv:1806.03664 [cs, stat]*.
- Durkan, C., Bekasov, A., Murray, I., and Papamakarios, G. (2019). Neural spline flows. In *Advances in Neural Information Processing Systems*, pages 7509–7520.
- Gao, R., Nijkamp, E., Kingma, D. P., Xu, Z., Dai, A. M., and Wu, Y. N. (2019). Flow contrastive estimation of energy-based models. *arXiv preprint arXiv:1912.00589*.
- Gutmann, M. and Hyvärinen, A. (2010). Noise-contrastive estimation: A new estimation principle for unnormalized statistical models. In *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics*, pages 297–304.
- Gutmann, M. U. and Hyvärinen, A. (2012). Noise-contrastive estimation of unnormalized statistical models, with applications to natural image statistics. *Journal of Machine Learning Research*, 13(Feb):307–361.
- Hyvärinen, A. (2005). Estimation of non-normalized statistical models by score matching. *Journal of Machine Learning Research*, 6(Apr):695–709.
- Hyvärinen, A. (2007). Some extensions of score matching. *Computational statistics & data analysis*, 51(5):2499–2512.
- Hyvärinen, A. and Hurri, J. (2004). Blind separation of sources that have spatiotemporal variance dependencies. *Signal Processing*, 84(2):247–254.
- Hyvärinen, A. and Morioka, H. (2016). Unsupervised feature extraction by time-contrastive learning and nonlinear ICA. In *Advances in Neural Information Processing Systems*, pages 3765–3773.
- Hyvärinen, A. and Morioka, H. (2017). Nonlinear ICA of temporally dependent stationary sources. In *The 20th International Conference on Artificial Intelligence and Statistics*.
- Hyvärinen, A. and Oja, E. (2000). Independent component analysis: Algorithms and applications. *Neural Networks*, 13(4):411–430.
- Hyvärinen, A. and Pajunen, P. (1999). Nonlinear independent component analysis: Existence and uniqueness results. *Neural Networks*, 12(3):429–439.
- Hyvärinen, A., Sasaki, H., and Turner, R. (2019). Nonlinear ICA Using Auxiliary Variables and Generalized Contrastive Learning. In *The 22nd International Conference on Artificial Intelligence and Statistics*, pages 859–868.
- Khemakhem, I., Kingma, D. P., Monti, R. P., and Hyvärinen, A. (2020). Variational Autoencoders and Nonlinear ICA: A Unifying Framework. In *The 23rd International Conference on Artificial Intelligence and Statistics*.
- Kingma, D. P. and Ba, J. (2014). Adam: A Method for Stochastic Optimization. *arXiv:1412.6980 [cs]*.
- Kingma, D. P. and Dhariwal, P. (2018). Glow: Generative flow with invertible 1x1 convolutions. In *Advances in Neural Information Processing Systems*, pages 10215–10224.
- Kingma, D. P. and Welling, M. (2013). Auto-Encoding Variational Bayes. *arXiv:1312.6114 [cs, stat]*.
- Lin, G., Milan, A., Shen, C., and Reid, I. (2017). Refinenet: Multi-path refinement networks for high-resolution semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1925–1934.

- Monti, R. P. and Hyvärinen, A. (2018). A Unified Probabilistic Model for Learning Latent Factors and Their Connectivities from High-Dimensional Data. *arXiv:1805.09567 [cs, stat]*.
- Monti, R. P., Zhang, K., and Hyvärinen, A. (2019). Causal discovery with general non-linear relationships using non-linear ica. *arXiv preprint arXiv:1904.09096*.
- Rezende, D. J. and Mohamed, S. (2015). Variational Inference with Normalizing Flows. *arXiv:1505.05770 [cs, stat]*.
- Saremi, S., Mehrjou, A., Schölkopf, B., and Hyvärinen, A. (2018). Deep Energy Estimator Networks. *arXiv preprint arXiv:1805.08306*.
- Song, Y. and Ermon, S. (2019). Generative modeling by estimating gradients of the data distribution. In *Advances in Neural Information Processing Systems*, pages 11895–11907.
- Sorrenson, P., Rother, C., and Köthe, U. (2020). Disentanglement by Nonlinear ICA with General Incompressible-flow Networks (GIN). *arXiv:2001.04872 [cs, stat]*.
- Uehara, M., Kanamori, T., Takenouchi, T., and Matsuda, T. (2020). A unified statistically efficient estimation framework for unnormalized models. In *International Conference on Artificial Intelligence and Statistics*, pages 809–819.
- Vincent, P. (2011). A connection between score matching and denoising autoencoders. *Neural computation*, 23(7):1661–1674.

Appendix for

ICE-BeeM: Identifiable Conditional Energy-Based Deep Models Based on Nonlinear ICA

We divide the Appendix into 5 main sections:

- Section **A**: we give extensive details on the experimental setup, as well as additional experiments;
- Section **B**: we discuss the estimation algorithms we used with ICE-BeeM and how they can be extended to the conditional setting;
- Section **C**: we prove the identifiability of ICE-BeeM and its universal approximation capability;
- Section **D**: we show how ICE-BeeM estimates IMCA;
- Section **E**: we provide a thorough theoretical analysis of the IMCA framework and draw parallels to the identifiability results in nonlinear ICA.

A Experimental protocol

A.1 Model architecture details

In this section, we describe the neural network architectures used for the experiments of Section 5.1, on the image datasets (MNIST, FashionMNIST, CIFAR10 and CIFAR100). Code to reproduce these experiments can be found in the supplementary material.

We can distinguish three different types of configurations:

1. A series of fully connected layers — denoted *MLP*. This configuration satisfies the assumptions of Section 2.3.
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 [2].

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 (condition 3 of Theorem 2) or with augmented features (condition 4 of Theorem 2). 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 5.1 in Table [3].

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

Table 2: Architecture detail

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: $n_c = 3, w = 32$ CIFAR100: $n_c = 3, w = 32$
	Output: d_z	
<i>MLP</i>	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	
<i>ConvMLP</i>	Input: $d_x = w \times w \times c$ Conv $w \times w \times 32$, BatchNorm, ReLU Conv $w \times w \times 64$, BatchNorm, ReLU MaxPool $\frac{w}{2} \times \frac{w}{2} \times 64$ Conv $\frac{w}{2} \times \frac{w}{2} \times 128$, BatchNorm, ReLU Conv $\frac{w}{2} \times \frac{w}{2} \times 256$, BatchNorm, ReLU MaxPool $\frac{w}{4} \times \frac{w}{4} \times 256$ Conv $1 \times 1 \times 256$ Dropout(0.1) FC 256, LeakyReLU(0.1) FC d_z	stride 1 for all conv. layers padding 1, filter size 3 padding 1, filter size 3 padding 1, filter size 3 padding 1, filter size 3 padding 1, filter size 3 padding 0, filter size $\frac{w}{4}$
<i>Unet</i>	Input: $d_x = w \times w \times n_c$ Conv $w \times w \times 64$ 4-cascaded RefinNet activation: ELU normalization: InstanceNorm+ InstanceNorm+, ELU Conv $w \times w \times n_c$ FC d_z	stride 1 for all conv. layers padding 1, filter size 3 see Song and Ermon (2019) exponential LU see Song and Ermon (2019) padding 1, filter size 3 only if $d_z < d_x$

Table 3: Architectures used in the experiments

Fig./Tab.	Dataset	Description	Configuration
Fig. [1a]	MNIST	Quantifying quality of representations	<i>Unet-a</i>
Fig. [1b]	CIFAR10	Quantifying quality of representations	<i>Unet</i>
Fig. [1b]	CIFAR100	Quantifying quality of representations	<i>Unet</i>
Fig. [1c]	MNIST	Transfer learning	<i>ConvMLP-50</i>
Fig. [1d]	CIFAR10	Transfer learning	<i>ConvMLP-90</i>
Tab. [1a]	MNIST	Transfer learning	<i>ConvMLP-50</i>
Tab. [1a]	CIFAR10	Transfer learning	<i>ConvMLP-90</i>
Tab. [1b]	FashionMNIST	Semi-supervised learning	<i>ConvMLP-50</i>
Tab. [1b]	CIFAR10	Semi-supervised learning	<i>ConvMLP-50p</i>

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 B.1) to train the energy models. The noise parameter used is $\sigma = 0.01$.

A.2 The MCC metric

To quantify identifiability, we use the mean correlation coefficient (MCC) metric, which computes the maximum linear correlations up to permutation of components. To obtain the value of this metric between two vectors \mathbf{x} and \mathbf{y} , we first calculate all pairs of correlation coefficients between the components x_i of \mathbf{x} , and the components y_j of \mathbf{y} . Since the order of the components in each vector can be arbitrary, we have to account for possible permutations between the indices i and j . This is done by solving a linear sum assignment problem (for instance, using the auction algorithm). We finally average over all correlation coefficients (after finding the right permutation). This makes the MCC metric invariant by permutation and component-wise transformations (as a consequence of the transformation invariance of the correlation coefficient).

To better understand this metric, let’s consider the following example. Let $\mathbf{x} \in \mathbb{R}^2$ be a bivariate random variable such that $x_1 \perp\!\!\!\perp x_2$, and let $\mathbf{y} = (x_2^2, x_1^2)$. If we don’t account for any permutations, then the average correlation is equal to $\frac{1}{2} \sum_i \text{corr}(x_i, y_i) = 0$ 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 determines that of \mathbf{y} . Thus, we have to find the optimal permutation of the elements of \mathbf{y} in order to maximize the average correlation. The MCC does this by computing all pair-wise correlations, and finding the assignment that maximizes the average correlation.

When the latent ground truth is known (Section 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.

When the ground truth is unknown (Section 5.1—real image datasets), we compare pairs of learnt representations, each from a different random initialization. A consistently high MCC means that changing the random state of the model doesn’t drastically change the learnt representations.

A.3 Quality of representations

We argued that conditioning enables EBMs to learn identifiable representations. The results in Section 5.1 validate this. The plots presented in Figures [1a] and [1b] were produced using the *Unet* configuration, described in Table [3]. 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 (2) and the permutation σ in equation (3) 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 [2] and [3].

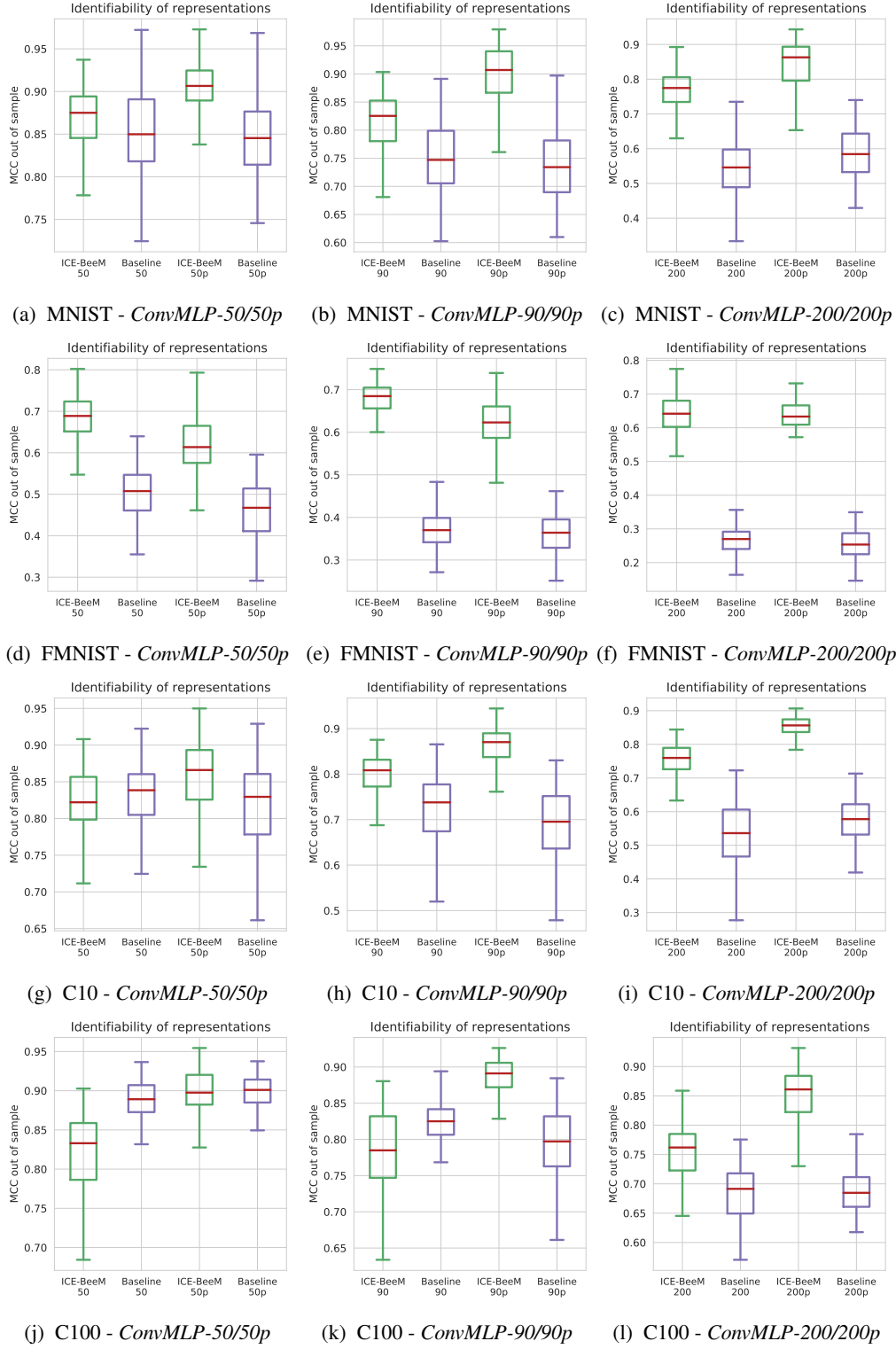


Figure 2: Further experiments on the *strong* identifiability of learnt representations using the *ConvMLP* architecture on image datasets — C10/100 stands for CIFAR10 and CIFAR100, respectively.

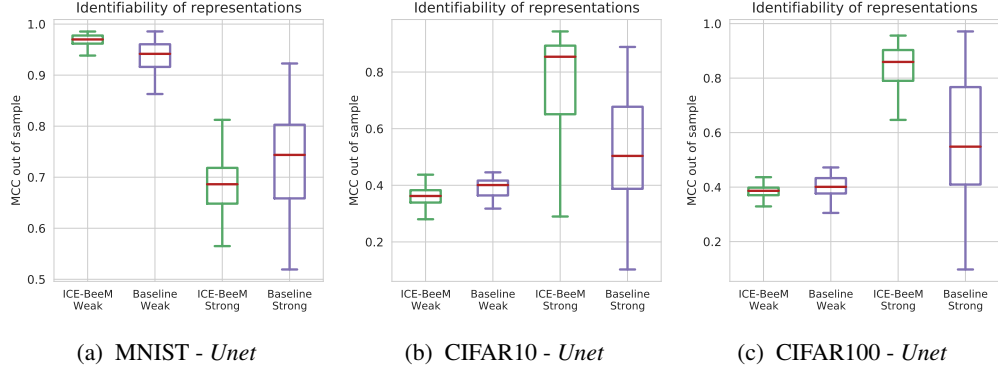


Figure 3: Further experiments on the identifiability of representations using the *Unet* architecture on image datasets.

A.4 Transfer learning experiments

A.4.1 Intuition

As a practical application of our framework where identifiability is important, we consider meta-learning, in particular multi-task and transfer learning. Assume we have N datasets, which could be, *e.g.*, different subjects in biomedical settings, or different image datasets. This fits well with our framework, where $y = 1, \dots, N$ 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 each 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 $f_{i,\theta}$ as

$$\log p(\mathbf{x}; y) + \log Z(\theta) \approx \sum_{i=1}^k g_i(y) f_{i,\theta}(\mathbf{x}) \quad (6)$$

where the $g_i(y)$ are scalar parameters as a function of y , which act as coefficients in the basis $(f_{i,\theta})$. 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 (6). This leads to a drastic simplification: we can learn the basis functions $f_{i,\theta}$ 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 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 1. 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 Theorem 2.

Training can be done by any method for EBM estimation. In particular, it is very easy by score matching because equation (6) is an exponential family for fixed f_i (Hyvärinen, 2007).

A.4.2 Further 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 labels 85-99 the train partition for CIFAR100.

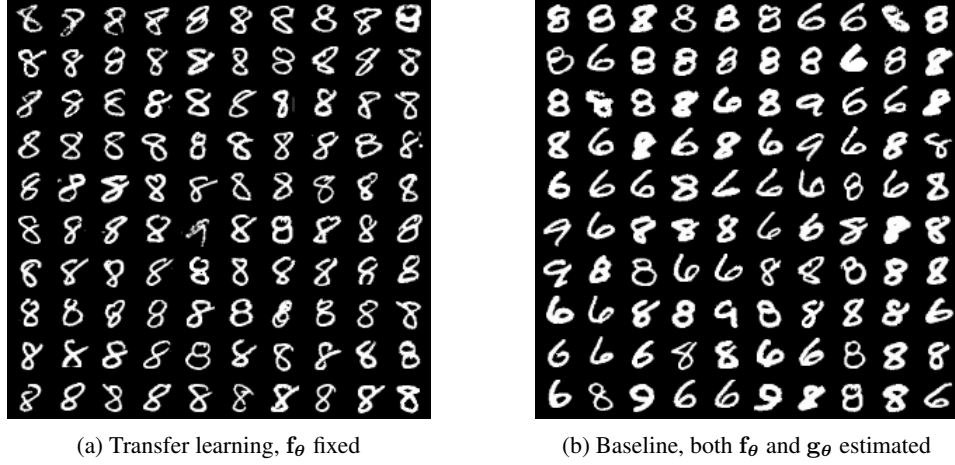


Figure 4: Further results for transfer learning experiments on MNIST. In the case of transfer learning 99 out of a hundred returned digits are class 8 compared to only 58 in the baseline.

We considered a subset of size 6000 to produce the values in Table [1a]. This table should be read in conjunction with Figures [1c]-[1d] for a proper evaluation of performance.

We present further plots and results of transfer learning experiments in Figures [5]-[6] and Table [4] ran on MNIST, FashionMNIST, CIFAR10 and CIFAR100 for a variety of different configurations, for different configurations and datasets. We considered a subset of size 6000 to produce the values in Table 4. 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 [5]-[6].

As an additional way to visualize the results, Figure [4a] 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 [4b]), incorrectly assigns high confidences to other digits.

A.5 Semi-supervised learning

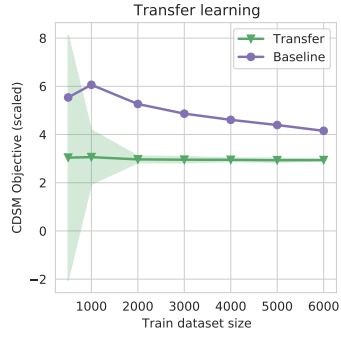
In this experiment, we train both an identifiable ICE-BeeM model and an unconditional (non-identifiable) EBM on classes 0-7. The purpose of this step is to learn a feature extractor f_θ that is able of learning meaningful features from the images. To test the quality of the features learnt by both models (the ICE-BeeM, and the unconditional EBM), we use the feature map f_θ 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 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.

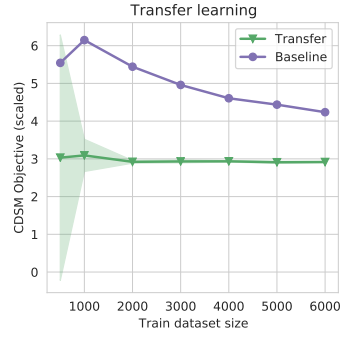
We present further results for the semi-supervised learning experiments in Table [5], ran on MNIST, FashionMNIST, CIFAR10 for a variety of different configurations.

A.6 IMCA and nonlinear ICA simulations

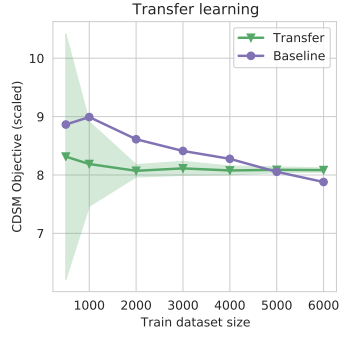
We give here more detail on the data generation process for the simulations in Section 5.2, as well as the architectures used.



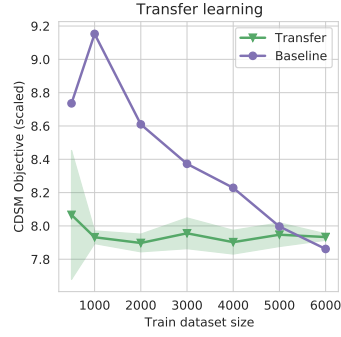
(a) MNIST - *ConvMLP-50*



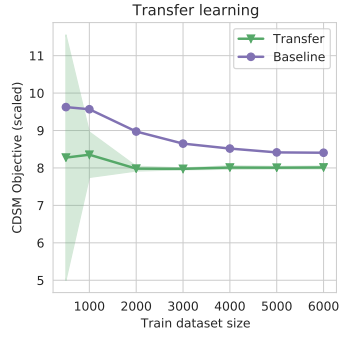
(b) MNIST - *ConvMLP-200*



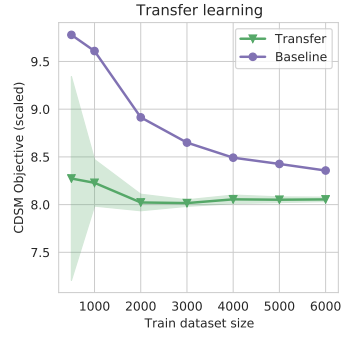
(c) FMNIST - *ConvMLP-90*



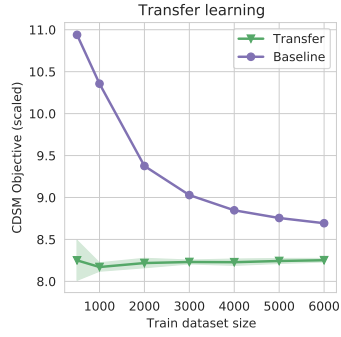
(d) FMNIST - *ConvMLP-90p*



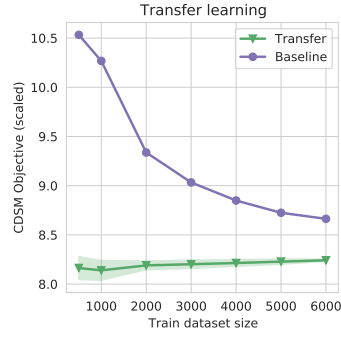
(e) CIFAR10 - *ConvMLP-200*



(f) CIFAR10 - *ConvMLP-200p*

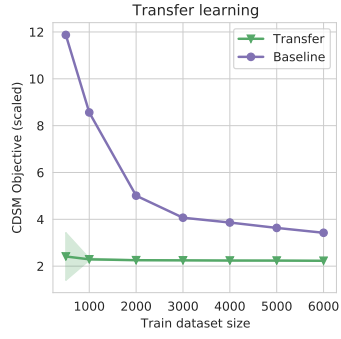


(g) CIFAR100 - *ConvMLP-50*

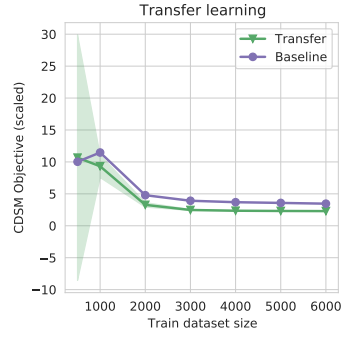


(h) CIFAR100 - *ConvMLP-50p*

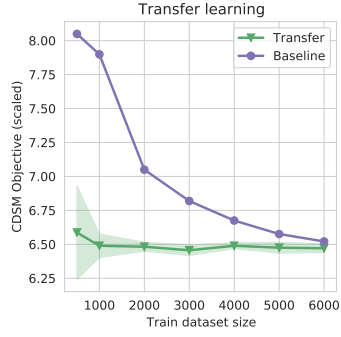
Figure 5: Further transfer learning — the dataset/configuration combo are reported in the captions.



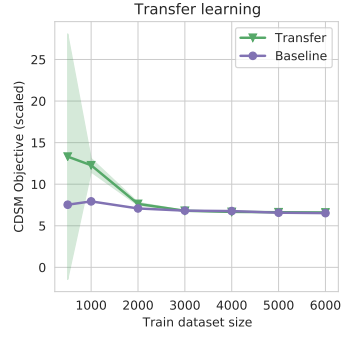
(a) MNIST - *Unet*



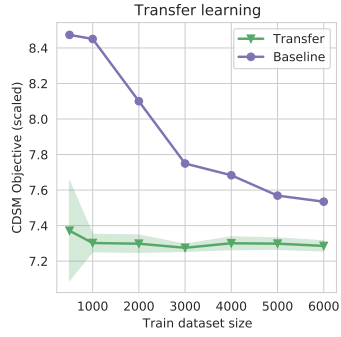
(b) MNIST - *Unet-a*



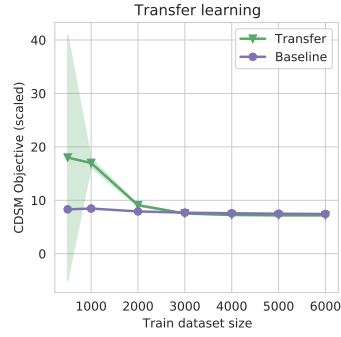
(c) FMNIST - *Unet*



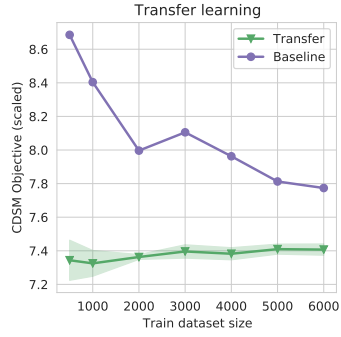
(d) FMNIST - *Unet-a*



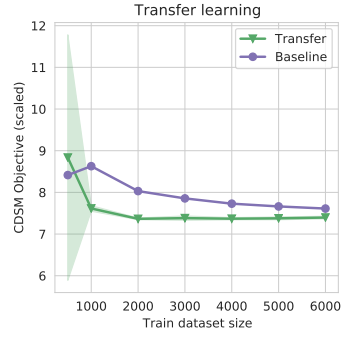
(e) CIFAR10 - *Unet*



(f) CIFAR10 - *Unet-a*



(g) CIFAR100 - *Unet*



(h) CIFAR100 - *Unet-a*

Figure 6: Further transfer learning — the dataset/configuration combo are reported in the captions.

Table 4: Transfer learning — CDSM score (lower is better)

Dataset	Configuration	$\mathbf{f} \cdot \mathbf{g}_\theta$	$\mathbf{f} \cdot \mathbf{1}$	$\mathbf{f}_\theta \cdot \mathbf{g}_\theta$	$\mathbf{f}_\theta \cdot \mathbf{1}$
MNIST	<i>ConvMLP-50</i>	2.95 ± 0.02	23.43 ± 0.04	4.22 ± 0.15	3.64 ± 0.10
	<i>ConvMLP-50p</i>	2.79 ± 0.00	796.99 ± 0.86	10.13 ± 4.74	3.63 ± 0.09
	<i>ConvMLP-90</i>	2.94 ± 0.01	12.18 ± 0.03	4.29 ± 0.13	3.67 ± 0.12
	<i>ConvMLP-90p</i>	3.03 ± 0.01	694.94 ± 1.03	10.22 ± 4.63	3.70 ± 0.12
	<i>ConvMLP-200</i>	2.91 ± 0.01	27.70 ± 0.02	4.29 ± 0.12	3.74 ± 0.09
	<i>ConvMLP-200p</i>	2.95 ± 0.01	805.45 ± 3.56	12.08 ± 3.79	3.71 ± 0.13
	<i>Unet</i>	2.23 ± 0.01	10.04 ± 0.01	3.44 ± 0.03	2.97 ± 0.25
	<i>Unet-a</i>	2.29 ± 0.01	6.18 ± 0.00	3.44 ± 0.02	6.27 ± 4.21
	<i>Unet-p</i>	14.00 ± 0.01	14.08 ± 0.00	11.97 ± 4.01	6.14 ± 4.17
	<i>Unet-50a</i>	2.61 ± 0.02	14.24 ± 0.01	3.79 ± 0.56	2.92 ± 0.20
	<i>MLP-50</i>	13.99 ± 0.01	13.99 ± 0.01	14.00 ± 0.01	14.00 ± 0.01
	<i>MLP-50p</i>	13.99 ± 0.01	14.00 ± 0.01	14.00 ± 0.01	14.00 ± 0.01
	<i>MLP-90</i>	14.00 ± 0.01	14.00 ± 0.01	14.00 ± 0.01	13.99 ± 0.01
	<i>MLP-90p</i>	13.99 ± 0.01	14.00 ± 0.01	14.00 ± 0.01	14.00 ± 0.01
	<i>MLP-200</i>	13.99 ± 0.01	14.00 ± 0.01	14.00 ± 0.01	14.00 ± 0.01
	<i>MLP-200p</i>	13.99 ± 0.01	13.99 ± 0.01	14.00 ± 0.01	14.00 ± 0.01
FMNIST	<i>ConvMLP-50</i>	7.88 ± 0.01	9.82 ± 0.03	7.88 ± 0.07	7.18 ± 0.25
	<i>ConvMLP-50p</i>	8.00 ± 0.02	197.84 ± 2.27	7.92 ± 0.18	7.10 ± 0.24
	<i>ConvMLP-90</i>	8.09 ± 0.02	10.86 ± 0.04	7.88 ± 0.05	7.14 ± 0.24
	<i>ConvMLP-90p</i>	7.94 ± 0.01	197.93 ± 2.33	7.87 ± 0.13	7.13 ± 0.20
	<i>ConvMLP-200</i>	7.98 ± 0.00	15.86 ± 0.01	7.91 ± 0.16	7.17 ± 0.21
	<i>ConvMLP-200p</i>	7.86 ± 0.01	196.14 ± 2.07	7.81 ± 0.15	7.11 ± 0.15
	<i>Unet</i>	6.47 ± 0.02	277.56 ± 1.06	6.52 ± 0.03	6.46 ± 0.07
	<i>Unet-a</i>	6.60 ± 0.02	24.62 ± 0.02	6.52 ± 0.02	6.41 ± 0.01
	<i>MLP-50</i>	13.99 ± 0.01	14.00 ± 0.01	13.99 ± 0.01	14.00 ± 0.01
	<i>MLP-200</i>	13.99 ± 0.01	14.00 ± 0.01	13.99 ± 0.01	14.00 ± 0.01
CIFAR10	<i>ConvMLP-50</i>	8.02 ± 0.01	32.09 ± 0.07	8.36 ± 0.03	8.15 ± 0.03
	<i>ConvMLP-50p</i>	8.04 ± 0.02	412.15 ± 2.54	8.35 ± 0.04	8.17 ± 0.01
	<i>ConvMLP-90</i>	8.03 ± 0.01	23.08 ± 0.04	8.37 ± 0.02	8.16 ± 0.05
	<i>ConvMLP-90p</i>	8.05 ± 0.01	408.51 ± 2.30	8.37 ± 0.04	8.16 ± 0.01
	<i>ConvMLP-200</i>	8.02 ± 0.02	13.35 ± 0.01	8.41 ± 0.07	8.13 ± 0.03
	<i>ConvMLP-200p</i>	8.06 ± 0.01	509.09 ± 2.31	8.35 ± 0.02	8.11 ± 0.03
	<i>Unet</i>	7.29 ± 0.01	118.93 ± 0.34	7.51 ± 0.05	9.21 ± 3.43
	<i>Unet-a</i>	7.18 ± 0.01	18.73 ± 0.01	7.48 ± 0.09	7.47 ± 0.13
	<i>Unet-50a</i>	7.30 ± 0.05	16.41 ± 0.00	7.64 ± 0.26	7.27 ± 0.03
	<i>MLP-50</i>	16.00 ± 0.00	16.00 ± 0.00	16.00 ± 0.00	16.00 ± 0.00
	<i>MLP-200</i>	16.00 ± 0.01	16.00 ± 0.00	16.00 ± 0.01	16.00 ± 0.00
CIFAR100	<i>ConvMLP-50</i>	8.25 ± 0.01	45.19 ± 0.15	8.69 ± 0.04	8.59 ± 0.02
	<i>ConvMLP-50p</i>	8.24 ± 0.01	2560.77 ± 7.15	8.68 ± 0.04	8.61 ± 0.04
	<i>ConvMLP-90</i>	8.23 ± 0.01	8.74 ± 0.01	8.68 ± 0.05	8.61 ± 0.03
	<i>ConvMLP-90p</i>	8.25 ± 0.01	3018.50 ± 7.27	8.65 ± 0.02	8.58 ± 0.03
	<i>ConvMLP-200</i>	8.26 ± 0.01	42.80 ± 0.09	8.69 ± 0.06	8.59 ± 0.03
	<i>ConvMLP-200p</i>	8.18 ± 0.01	3827.36 ± 16.14	8.65 ± 0.07	8.63 ± 0.05
	<i>Unet</i>	7.41 ± 0.02	106.28 ± 0.75	7.77 ± 0.05	8.38 ± 0.55
	<i>Unet-a</i>	7.39 ± 0.02	11.15 ± 0.01	7.82 ± 0.42	9.35 ± 3.33
	<i>Unet-50a</i>	7.54 ± 0.01	15.95 ± 0.00	7.97 ± 0.13	7.60 ± 0.05
	<i>MLP-50p</i>	16.00 ± 0.01	16.00 ± 0.00	16.00 ± 0.00	16.00 ± 0.00
	<i>MLP-200p</i>	16.00 ± 0.01	16.00 ± 0.00	16.00 ± 0.00	16.00 ± 0.00

Table 5: Semi-supervised learning — classification accuracy (higher is better)

Dataset	Configuration	ICE-BeeM	Unconditional EBM
MNIST	<i>ConvMLP-50</i>	76.98 ± 1.61	62.82 ± 1.48
	<i>ConvMLP-50p</i>	88.46 ± 1.14	66.58 ± 2.64
	<i>ConvMLP-90</i>	78.93 ± 1.51	71.61 ± 1.71
	<i>ConvMLP-90p</i>	78.66 ± 1.91	69.13 ± 1.49
	<i>ConvMLP-200</i>	81.21 ± 2.6	71.48 ± 2.23
	<i>ConvMLP-200p</i>	77.38 ± 1.32	68.99 ± 1.68
	<i>MLP-50</i>	91.74 ± 1.72	85.77 ± 1.14
	<i>MLP-50p</i>	92.21 ± 1.74	84.56 ± 1.1
	<i>MLP-90</i>	95.17 ± 0.46	85.91 ± 2.07
	<i>MLP-90p</i>	94.97 ± 0.7	85.97 ± 1.61
	<i>MLP-200</i>	94.36 ± 1.28	89.26 ± 1.7
	<i>MLP-200p</i>	91.81 ± 2.33	90.87 ± 1.05
	<i>Unet</i>	97.79 ± 0.34	98.39 ± 0.68
	<i>Unet-a</i>	97.18 ± 0.5	97.79 ± 0.78
	<i>Unet-50a</i>	97.52 ± 0.4	97.92 ± 0.49
	<i>Unet-20a</i>	95.64 ± 0.7	92.08 ± 1.71
FMNIST	<i>ConvMLP-50</i>	77.07 ± 1.39	56.33 ± 3.18
	<i>ConvMLP-50p</i>	71.67 ± 1.85	57.6 ± 2.24
	<i>ConvMLP-90</i>	74.13 ± 1.86	57.73 ± 3.12
	<i>ConvMLP-90p</i>	70.87 ± 1.13	60.07 ± 2.9
	<i>ConvMLP-200</i>	81.4 ± 1.93	68.27 ± 2.78
	<i>ConvMLP-200p</i>	78.47 ± 0.96	57.47 ± 2.62
	<i>MLP-50</i>	98.07 ± 1.06	90.47 ± 1.56
	<i>MLP-50p</i>	97.6 ± 0.53	90.47 ± 1.56
	<i>MLP-90</i>	97.8 ± 0.34	94.4 ± 0.53
	<i>MLP-90p</i>	97.8 ± 0.34	94.4 ± 0.53
	<i>MLP-200</i>	98.6 ± 0.49	94.87 ± 0.96
	<i>MLP-200p</i>	98.6 ± 0.65	95.33 ± 1.05
	<i>Unet</i>	99.67 ± 0.3	99.93 ± 0.13
	<i>Unet-a</i>	99.53 ± 0.16	99.87 ± 0.16
CIFAR10	<i>ConvMLP-50</i>	69.36 ± 2.23	56.39 ± 1.0
	<i>ConvMLP-50p</i>	64.42 ± 1.09	51.88 ± 1.33
	<i>ConvMLP-90</i>	68.24 ± 2.0	52.82 ± 0.95
	<i>ConvMLP-90p</i>	66.18 ± 1.01	52.33 ± 1.73
	<i>ConvMLP-200</i>	64.73 ± 1.36	54.18 ± 1.09
	<i>ConvMLP-200p</i>	66.3 ± 0.99	54.48 ± 1.28
	<i>MLP-50</i>	68.73 ± 1.35	70.27 ± 2.67
	<i>MLP-50p</i>	69.82 ± 1.78	69.36 ± 2.3
	<i>MLP-90</i>	71.58 ± 1.21	72.85 ± 1.16
	<i>MLP-90p</i>	71.12 ± 1.64	72.85 ± 1.16
	<i>MLP-200</i>	72.39 ± 1.92	72.97 ± 1.75
	<i>MLP-200p</i>	70.94 ± 1.25	71.97 ± 2.29
	<i>Unet</i>	80.27 ± 4.0	80.58 ± 0.9
	<i>Unet-a</i>	80.48 ± 1.45	80.48 ± 1.45
	<i>Unet-50a</i>	77.64 ± 1.02	73.79 ± 0.81
	<i>Unet-20a</i>	74.21 ± 0.73	68.82 ± 0.67

Data generation We generate 5-dimensional synthetic datasets following the nonlinear ICA model which is a special case of equation (4) 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 $\llbracket 1, M \rrbracket$. 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 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 $\mu(\mathbf{z}) \propto e^{-0.5 \mathbf{z}^T \boldsymbol{\Sigma}_0^{-1} \mathbf{z}}$. As before, we define $\boldsymbol{\lambda}(\mathbf{y})$ to be the distinct conditional precisions. The precision matrix of each segment is now equal to $\boldsymbol{\Sigma}(\mathbf{y})^{-1} = \boldsymbol{\Sigma}_0^{-1} + \text{diag}(\boldsymbol{\lambda}(\mathbf{y}))^{-1}$, 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 nonstationarity of the sources. TCL learns to invert the mixing function \mathbf{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., 2020), 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, 2013).

Training of ICE-BeeM via flow contrastive estimation To demonstrate that ICE-BeeM can be trained by any method for training EBM, we switched from denoising score matching to flow contrastive estimation (FCE, Appendix 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., 2019), an invertible 1×1 convolution (Kingma and Dhariwal, 2018), and an ActNorm layer (Kingma and Dhariwal, 2018). The flow parameters are updated by Adam optimizer, with a learning rate of 10^{-5} .

Used architectures The architectures used to produce Figures [1e] and [1f] are summarized by Table [6].

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 work we used two different options with good results for both: flow contrastive estimation (Gao et al., 2019) and denoising score matching (Vincent, 2011). Both methods can also be extended to the conditional case in a straightforward fashion.

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

Table 6: Architectures used in the simulations

Model	Optimizer	Architecture	
		Input	$d_x = 5$
		Condition	one hot encoded $d_y = M = 8$
		Latent	$d_z = d_x = 5$
		Num. layers	$L \in \{2, 4\}$
ICE-BeeM	Adam lr 3.10^{-4}	\mathbf{f}_θ	$(L + 1)$ -layer MLP batch norm after each FC layer hidden dim 32, LeakyReLU(0.1) act
		\mathbf{g}_θ	$(d_z \times d_y)$ 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)
			$(L - 2) \times [\text{FC } d_x, \text{maxout}(2)]$
			FC d_x , absolute value

properties of the observed data. To address this concern [Gao et al. \(2019\)](#) 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. We naturally get a conditional version for FCE by learning a conditional EBM ([Gao et al., 2019](#), eq. 12).

Score matching is another well-known method for learning unnormalized models ([Hyvärinen, 2005](#)). However, its computational implementation in deep networks is problematic, which is why [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](#)).

B.1 Conditional denoising score matching

We extend the original score matching objective to the conditional setting in a natural way: for a fixed \mathbf{y} , we compute the unconditional 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} . The expression of the conditional score matching objective is then:

$$\mathcal{J}_{\text{CSM}}(\boldsymbol{\theta}) = \mathbb{E}_{p(\mathbf{x}, \mathbf{y})} \|\nabla_{\mathbf{x}} \log p_{\boldsymbol{\theta}}(\mathbf{x}|\mathbf{y}) - \nabla_{\mathbf{x}} \log p(\mathbf{x}|\mathbf{y})\|^2 \quad (7)$$

We build on the recent developments by [Vincent \(2011\)](#), and introduce a conditional denoising score matching objective by replacing the unknown density by a kernel density estimator. Formally, given observations $\mathcal{D} = \{(\mathbf{x}^{(1)}, \mathbf{y}^{(1)}), \dots, (\mathbf{x}^{(N)}, \mathbf{y}^{(N)})\}$, 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:

$$q_b(\mathbf{y}) = \mathbb{E}_{\mathbf{y}' \sim q_{\mathcal{D}}} [l_b(\mathbf{y}|\mathbf{y}')] \quad (8)$$

$$q_{ab}(\mathbf{x}, \mathbf{y}) = \mathbb{E}_{(\mathbf{x}', \mathbf{y}') \sim q_{\mathcal{D}}} [k_a(\mathbf{x}|\mathbf{x}') l_b(\mathbf{y}|\mathbf{y}')] \quad (9)$$

$$q_{ab}(\mathbf{x}|\mathbf{y}) = \frac{q_{ab}(\mathbf{x}, \mathbf{y})}{q_b(\mathbf{y})} \quad (10)$$

where k_a and l_b are bounded kernel functions defined on \mathcal{X} and \mathcal{Y} and with bandwidths⁵ a and b , respectively. 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 (7) by their estimates $q_\sigma(\mathbf{x}, \mathbf{y})$ and $q_\sigma(\mathbf{x}|\mathbf{y})$, to arrive at the new objective

$$\mathcal{J}_{\text{CSM}_\sigma}(\boldsymbol{\theta}) = \mathbb{E}_{q_\sigma(\mathbf{x}, \mathbf{y})} \|\nabla_{\mathbf{x}} \log p_\theta(\mathbf{x}|\mathbf{y}) - \nabla_{\mathbf{x}} \log q_\sigma(\mathbf{x}|\mathbf{y})\|^2 \quad (11)$$

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 $k_\sigma(\mathbf{x}|\mathbf{y})$:

$$\mathcal{J}_{\text{CDSM}_\sigma}(\boldsymbol{\theta}) = \mathbb{E} \|\nabla_{\mathbf{x}} \log p_\theta(\mathbf{x}|\mathbf{y}) - \nabla_{\mathbf{x}} \log k_\sigma(\mathbf{x}|\mathbf{y}')\|^2 \quad (12)$$

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 $\mathcal{J}_{\text{CSM}_\sigma}$, but it has the advantage of being simpler to evaluate and interpret.

Above, we presented this objective when k_σ is the Gaussian kernel, and l_σ is simply the identity kernel.

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 (12) from the CSM objective (11):

$$\mathcal{J}_{\text{CSM}_\sigma}(\boldsymbol{\theta}) = \mathbb{E}_{q_\sigma(\mathbf{x}, \mathbf{y})} \left\| \nabla_{\mathbf{x}} \log \frac{p_\theta(\mathbf{x}|\mathbf{y})}{q_\sigma(\mathbf{x}|\mathbf{y})} \right\|^2 = \mathbb{E}_{q_\sigma(\mathbf{x}, \mathbf{y})} \|\nabla_{\mathbf{x}} \log p_\theta(\mathbf{x}|\mathbf{y})\|^2 - S(\boldsymbol{\theta}) + C_1 \quad (13)$$

where C_1 is a constant term that only depends on $q_\sigma(\mathbf{x}|\mathbf{y})$, and

$$\begin{aligned} S(\boldsymbol{\theta}) &= \mathbb{E}_{q_\sigma(\mathbf{x}, \mathbf{y})} \langle \nabla_{\mathbf{x}} \log p_\theta(\mathbf{x}|\mathbf{y}), \nabla_{\mathbf{x}} \log q_\sigma(\mathbf{x}|\mathbf{y}) \rangle \\ &= \int q_\sigma(\mathbf{x}, \mathbf{y}) \langle \nabla_{\mathbf{x}} \log p_\theta(\mathbf{x}|\mathbf{y}), \frac{\nabla_{\mathbf{x}} q_\sigma(\mathbf{x}|\mathbf{y})}{q_\sigma(\mathbf{x}|\mathbf{y})} \rangle d\mathbf{x} d\mathbf{y} \\ &= \int q_\sigma(\mathbf{y}) \langle \nabla_{\mathbf{x}} \log p_\theta(\mathbf{x}|\mathbf{y}), \nabla_{\mathbf{x}} q_\sigma(\mathbf{x}|\mathbf{y}) \rangle d\mathbf{x} d\mathbf{y} \\ &= \int q_\sigma(\mathbf{y}) \langle \nabla_{\mathbf{x}} \log p_\theta(\mathbf{x}|\mathbf{y}), \nabla_{\mathbf{x}} \frac{\int p_{\mathcal{D}}(\mathbf{x}', \mathbf{y}') k_\sigma(\mathbf{x}|\mathbf{x}') l_\sigma(\mathbf{y}|\mathbf{y}') d\mathbf{x}' d\mathbf{y}'}{q_\sigma(\mathbf{y})} \rangle d\mathbf{x} d\mathbf{y} \\ &= \int \int p_{\mathcal{D}}(\mathbf{x}', \mathbf{y}') l_\sigma(\mathbf{y}|\mathbf{y}') k_\sigma(\mathbf{x}|\mathbf{x}') \langle \nabla_{\mathbf{x}} \log p_\theta(\mathbf{x}|\mathbf{y}), \nabla_{\mathbf{x}} \log k_\sigma(\mathbf{x}|\mathbf{x}') \rangle d\mathbf{x}' d\mathbf{y}' d\mathbf{x} d\mathbf{y} \\ &= \mathbb{E}_{p_{\mathcal{D}}(\mathbf{x}', \mathbf{y}') k_\sigma(\mathbf{x}|\mathbf{x}') l_\sigma(\mathbf{y}|\mathbf{y}')} \langle \nabla_{\mathbf{x}} \log p_\theta(\mathbf{x}|\mathbf{y}), \nabla_{\mathbf{x}} \log k_\sigma(\mathbf{x}|\mathbf{x}') \rangle \end{aligned}$$

Plugging this back into equation (13), we find that

$$\begin{aligned} \mathcal{J}_{\text{CSM}_\sigma}(\boldsymbol{\theta}) &= \mathbb{E} \|\nabla_{\mathbf{x}} \log p_\theta(\mathbf{x}|\mathbf{y}) - \nabla_{\mathbf{x}} \log k_\sigma(\mathbf{x}|\mathbf{x}')\|^2 + C_1 - C_2 \\ &= \mathcal{J}_{\text{CDSM}_\sigma}(\boldsymbol{\theta}) + C_1 - C_2 \end{aligned}$$

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 $k_\sigma(\mathbf{x}|\mathbf{x}')$. \square

B.2 Conditional flow contrastive estimation

As described above, FCE learns the parameter for the density p_θ of an EBM by performing a surrogate classification task: noise is generated from a noise distribution q_α which is parameterized 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:

$$\mathcal{J}_{\text{FCE}}(\boldsymbol{\theta}, \boldsymbol{\alpha}) = \mathbb{E}_{p_{\text{data}}(\mathbf{x})} \log \frac{p_\theta(\mathbf{x})}{q_\alpha(\mathbf{x}) + p_\theta(\mathbf{x})} + \mathbb{E}_{q_\alpha(\mathbf{x})} \log \frac{q_\alpha(\mathbf{x})}{q_\alpha(\mathbf{x}) + p_\theta(\mathbf{x})} \quad (14)$$

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

⁵the bandwidths satisfy $a = a_N$ and $b = b_N$, and are positive bandwidth sequences which decay to 0 as $N \rightarrow +\infty$, where N is the size of the dataset \mathcal{D} .

naturally: we replace the model density by the conditional density $p_{\theta}(\mathbf{x}|\mathbf{y})$. In the conditional case, it follows that noise samples should also be associated with a conditioning variable, \mathbf{y} . One way this can be achieved is by also 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. This would require marginalizing over the conditioning variable, \mathbf{y} . The objective simply becomes:

$$\mathcal{J}_{\text{CFCE}}(\theta, \alpha) = \mathbb{E}_{p_{\text{data}}(\mathbf{x}, \mathbf{y})} \log \frac{p_{\theta}(\mathbf{x}|\mathbf{y})}{q_{\alpha}(\mathbf{x}, \mathbf{y}) + p_{\theta}(\mathbf{x}|\mathbf{y})} + \mathbb{E}_{q_{\alpha}(\mathbf{x}, \mathbf{y})} \log \frac{q_{\alpha}(\mathbf{x}, \mathbf{y})}{q_{\alpha}(\mathbf{x}, \mathbf{y}) + p_{\theta}(\mathbf{x}|\mathbf{y})} \quad (15)$$

We can write the flow density as $q_{\alpha}(\mathbf{x}, \mathbf{y}) = p(\mathbf{y})q_{\alpha}(\mathbf{x}|\mathbf{y})$. 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 sample draw an index from a uniform distribution, and use the conditional flow to sample an observation.

C Identifiability of the conditional energy-based model

Recall the form of our conditional energy model

$$p_{\theta}(\mathbf{x}|\mathbf{y}) = Z(\mathbf{y}; \theta)^{-1} \exp(-\mathbf{f}_{\theta}(\mathbf{x})^T \mathbf{g}_{\theta}(\mathbf{y})) \quad (16)$$

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 $n = d_z$.

C.1 More on the equivalence relations

The relation $\sim_w^{\mathbf{f}}$ in equation (2) 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 it 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 $\mathbf{f}_{\theta} = \mathbf{A}_1 \mathbf{f}_{\theta'} + \mathbf{c}_1$ and $\mathbf{f}_{\theta'} = \mathbf{A}_2 \mathbf{f}_{\theta} + \mathbf{c}_2$. We present the simpler version in the paper for clarity.

C.2 Proof of Theorem 1

We start by proving the main theoretical result of this paper, which applies to all dimensions of the feature extractor. Alternative and weaker assumptions are discussed after the proof.

Theorem 1 (Identifiable conditional EBMs). *Assume:*

1. *The feature extractor \mathbf{f} is differentiable, and its Jacobian $\mathbf{J}_{\mathbf{f}}$ is full rank.*
2. *There exist $n + 1$ points $\mathbf{y}^0, \dots, \mathbf{y}^n$ such that the matrix*

$$\mathbf{R} = (\mathbf{g}(\mathbf{y}^1) - \mathbf{g}(\mathbf{y}^0), \dots, \mathbf{g}(\mathbf{y}^n) - \mathbf{g}(\mathbf{y}^0)) \quad (17)$$

of size $n \times n$ is invertible.

then

$$p_{\theta}(\mathbf{x}|\mathbf{y}) = p_{\theta'}(\mathbf{x}|\mathbf{y}) \implies \theta \sim_w^{\mathbf{f}} \theta'$$

where $\sim_w^{\mathbf{f}}$ is defined as follows:

$$\theta \sim_w^{\mathbf{f}} \theta' \Leftrightarrow \mathbf{f}_{\theta}(\mathbf{y}) = \mathbf{A} \mathbf{f}_{\theta'}(\mathbf{y}) + \mathbf{c} \quad (18)$$

\mathbf{A} is a $(d_z \times d_z)$ -matrix of rank at least $\min(d_z, d_x)$.

If, instead or in addition, we assume that:

3. *The feature extractor \mathbf{g} is differentiable, and its Jacobian $\mathbf{J}_{\mathbf{g}}$ is full rank.*
4. *There exist $n + 1$ points $\mathbf{x}^0, \dots, \mathbf{x}^n$ such that the matrix*

$$\mathbf{Q} = (\mathbf{f}(\mathbf{x}^1) - \mathbf{f}(\mathbf{x}^0), \dots, \mathbf{f}(\mathbf{x}^n) - \mathbf{f}(\mathbf{x}^0))$$

of size $n \times n$ is invertible.

then

$$p_{\theta}(\mathbf{x}|\mathbf{y}) = p_{\theta'}(\mathbf{x}|\mathbf{y}) \implies \theta \sim_w^{\mathbf{g}} \theta'$$

where $\sim_w^{\mathbf{g}}$ is defined as follows:

$$\theta \sim_w^{\mathbf{g}} \theta' \Leftrightarrow \mathbf{g}_{\theta}(\mathbf{y}) = \mathbf{B}\mathbf{g}_{\theta'}(\mathbf{y}) + \mathbf{e} \quad (19)$$

\mathbf{B} is a $(d_z \times d_z)$ -matrix of rank at least $\min(d_z, d_x)$.

Finally, if $d_z \geq \max(d_x, d_y)$ and all assumptions 1-4 hold, then the matrices \mathbf{A} and \mathbf{B} have full rank (equal to d_z).

Proof. 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 θ and $\tilde{\theta}$ such that

$$p_{\theta}(\mathbf{x}|\mathbf{y}) = p_{\tilde{\theta}}(\mathbf{x}|\mathbf{y}) \quad (20)$$

Then, by applying the logarithm to both sides, we get:

$$\log Z(\mathbf{y}; \theta) - \mathbf{f}_{\theta}(\mathbf{x})^T \mathbf{g}_{\theta}(\mathbf{y}) = \log Z(\mathbf{y}; \tilde{\theta}) - \mathbf{f}_{\tilde{\theta}}(\mathbf{x})^T \mathbf{g}_{\tilde{\theta}}(\mathbf{y}) \quad (21)$$

Consider the points $\mathbf{y}^0, \dots, \mathbf{y}^n$ provided by assumption 2 for \mathbf{g}_{θ} . We plug each of these points in (21) 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:

$$\mathbf{R}\mathbf{f}_{\theta}(\mathbf{x}) = \tilde{\mathbf{R}}\mathbf{f}_{\tilde{\theta}}(\mathbf{x}) + \mathbf{b} \quad (22)$$

where $\mathbf{R} = (\dots, \mathbf{g}_{\theta}(\mathbf{y}^l) - \mathbf{g}_{\theta}(\mathbf{y}^0), \dots)$, $\tilde{\mathbf{R}} = (\dots, \mathbf{g}_{\tilde{\theta}}(\mathbf{y}^l) - \mathbf{g}_{\tilde{\theta}}(\mathbf{y}^0), \dots)$, and $\mathbf{b} = (\dots, \log \frac{Z(\mathbf{y}^l; \theta)}{Z(\mathbf{y}^l; \tilde{\theta})} - \log \frac{Z(\mathbf{y}^0; \theta)}{Z(\mathbf{y}^0; \tilde{\theta})}, \dots)$. Since \mathbf{R} is invertible (by assumption 2), we multiply by its inverse from the left to get:

$$\mathbf{f}_{\theta}(\mathbf{x}) = \mathbf{A}\mathbf{f}_{\tilde{\theta}}(\mathbf{x}) + \mathbf{c} \quad (23)$$

where $\mathbf{A} = \mathbf{R}^{-1}\tilde{\mathbf{R}}$ and $\mathbf{c} = \mathbf{R}^{-1}\mathbf{b}$. Now since \mathbf{f}_{θ} is differentiable and its Jacobian is full rank (assumption 1), by differentiating the last equation we deduce that $\text{rank}(\mathbf{A}) \geq \min(n, d)$, which in turn proves that $\theta \sim_w^{\mathbf{f}} \tilde{\theta}$.

Finally, suppose that in addition, assumption 4 holds. Then there exists $\mathbf{x}^0, \dots, \mathbf{x}^n$ such that $\mathbf{Q} := (\dots, \mathbf{f}_{\theta}(\mathbf{x}^i) - \mathbf{f}_{\theta}(\mathbf{x}^0), \dots)$. Plugging these $n + 1$ points into equation (23), and subtracting the first equation for \mathbf{x}^0 from the remaining n equations, we get

$$\mathbf{Q} = \mathbf{A}(\dots, \mathbf{f}_{\tilde{\theta}}(\mathbf{x}^i) - \mathbf{f}_{\tilde{\theta}}(\mathbf{x}^0), \dots) \quad (24)$$

Since \mathbf{Q} is an $n \times n$ invertible matrix, we conclude that \mathbf{A} is also invertible, which concludes the proof. \square

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.

Proof under weaker assumptions Assumption 1 of full rank Jacobian can be weakened without changing the conclusion of Theorem 1. In fact, this assumption is only used right after equation (23) to prove that the matrix \mathbf{A} has a rank that is at least equal to $\min(n, d)$. Suppose instead that

- 1.' There exists a point $\mathbf{x}^0 \in \mathbb{R}^d$ where the Jacobian $\mathbf{J}_{\mathbf{f}_{\theta}}$ of \mathbf{f}_{θ} exists and is invertible

Then by computing the differential of equation (23) at \mathbf{x}^0 (assuming that $\mathbf{J}_{\mathbf{f}_{\tilde{\theta}}}(\mathbf{x}^0)$ exists), we can make the same conclusion on the rank of \mathbf{A} .

In fact, this condition can be scrapped altogether if we relax the definition of the equivalence class in Appendix C.1 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 1 can be replaced by requiring the Jacobian of \mathbf{g}_{θ} to be differentiable and full rank in at least one point, but this requires the conditioning variable to be continuous.

C.3 Proof of Proposition 1

Proposition 1. 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:
 $d_l \geq d_{l+1}, \forall l \in \llbracket 0, L-1 \rrbracket$ or $d_l \leq d_{l+1}, \forall l \in \llbracket 0, L-1 \rrbracket$.

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 ,

$$\mathbf{x}^0 = \mathbf{x} \quad (25)$$

$$\bar{\mathbf{x}}^l = \mathbf{W}_l \mathbf{x}^{l-1} + \mathbf{b}_l \quad (26)$$

$$\mathbf{x}^l = h(\mathbf{W}_l \mathbf{x}^{l-1} + \mathbf{b}_l) = h(\bar{\mathbf{x}}^l) \quad (27)$$

$$h(y) = \alpha y \mathbf{1}_{y < 0} + y \mathbf{1}_{y > 0} \quad (28)$$

with h in equation (27) is 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

$$v_k^l = h'(\bar{x}_k^l) = \begin{cases} 1 & \text{if } \bar{x}_k^l > 0 \\ \alpha & \text{if } \bar{x}_k^l < 0 \end{cases} \quad (29)$$

which is undefined if $\bar{x}_k^l = 0$, 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,

$$\mathbf{x}^L = h(\bar{\mathbf{x}}^L) = \mathbf{V}_L \bar{\mathbf{x}}^L = \mathbf{V}_L \mathbf{W}_L \mathbf{x}^{L-1} + \mathbf{V}_L \mathbf{b}_{L-1} = \dots = \bar{\mathbf{V}}^L \mathbf{x} + \bar{\mathbf{b}}^L \quad (30)$$

where $\bar{\mathbf{V}}^l = \mathbf{V}_l \mathbf{W}_l \mathbf{V}_{l-1} \mathbf{W}_{l-1} \dots \mathbf{V}_1 \mathbf{W}_1$, $\bar{\mathbf{b}}^0 = 0$ and $\bar{\mathbf{b}}^l = \mathbf{V}_l \mathbf{b}_l + \mathbf{V}_l \mathbf{W}_l \bar{\mathbf{b}}^{l-1}$. This is of course only possible if $\bar{x}_k^l \neq 0$ for all $l \in \llbracket 1, L \rrbracket$ and all $k \in \llbracket 1, d_l \rrbracket$. As such, define the set

$$\mathcal{N} = \bigcup_{l=1}^L \bigcup_{k=1}^{d_l} \{ \mathbf{x} \in \mathbb{R}^d | \bar{x}_k^l = 0 \} = \bigcup_{l=1}^L \bigcup_{k=1}^{d_l} \{ \mathbf{x} \in \mathbb{R}^d | (\bar{\mathbf{v}}_k^l)^T \mathbf{x} + \bar{b}_k^l = 0 \} \quad (31)$$

where $\bar{\mathbf{v}}_k^l$ is the k -th row of $\bar{\mathbf{V}}^l$. For each $\mathbf{x} \notin \mathcal{N}$, we have that \mathbf{V}_l is full rank, and, using Lemma 2, $\bar{\mathbf{V}}^l$ is also a full rank matrix.

While it is true that \bar{b}_k^l and $\bar{\mathbf{v}}_k^l$ are functions of \mathbf{x} , yet they only take a finite number of values. Thus, the set $\{ \mathbf{x} \in \mathbb{R}^d | (\bar{\mathbf{v}}_k^l)^T \mathbf{x} + \bar{b}_k^l = 0 \}$ is included in the union over all the values taken by \bar{b}_k^j and $\bar{\mathbf{v}}_k^j$ up to layer l . For each of these values, the set becomes a dot product between a row of $\bar{\mathbf{V}}^j$ which is independent of the input \mathbf{x} , and is nonzero because $\bar{\mathbf{V}}^j$ 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:

$$\frac{\partial \mathbf{x}^L}{\partial \mathbf{x}} = \prod_{l=L}^1 \frac{\partial \mathbf{x}^l}{\partial \mathbf{x}^{l-1}} = \prod_{l=L}^1 \frac{\partial \mathbf{x}^l}{\partial \bar{\mathbf{x}}^l} \frac{\partial \bar{\mathbf{x}}^l}{\partial \mathbf{x}^{l-1}} = \prod_{l=L}^1 \mathbf{V}_l \mathbf{W}_l = \bar{\mathbf{V}}^L \quad (32)$$

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 $\bar{\mathbf{x}}^l$ as a function of \mathbf{x}^{l-1} if $d_{l-1} \geq d_l$ and $\text{rank}(\mathbf{W}_l) = d_l$. \square

Lemma 1. 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 $\mathbb{R}_*^n := \mathbb{R}^n \setminus \{0\}$, 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 $m \geq n$,

$$\sigma_{\min}(\mathbf{A}) = \sqrt{\lambda_{\min}(\mathbf{A}^T \mathbf{A})} = \sqrt{\inf_{\mathbf{x} \in \mathbb{R}_*^n} \frac{\mathbf{x}^T \mathbf{A}^T \mathbf{A} \mathbf{x}}{\mathbf{x}^T \mathbf{x}}} = \inf_{\mathbf{x} \in \mathbb{R}_*^n} \frac{\|\mathbf{A} \mathbf{x}\|}{\|\mathbf{x}\|} \quad (33)$$

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 $\mathbf{N} \mathbf{x} \neq 0$, $\forall \mathbf{x} \in \mathbb{R}_*^n$,

$$\begin{aligned} \sigma_{\min}(\mathbf{MN}) &= \inf_{\mathbf{x} \in \mathbb{R}_*^p} \frac{\|\mathbf{MN} \mathbf{x}\|}{\|\mathbf{x}\|} \\ &= \inf_{\mathbf{x} \in \mathbb{R}_*^p} \frac{\|\mathbf{MN} \mathbf{x}\| \|\mathbf{N} \mathbf{x}\|}{\|\mathbf{N} \mathbf{x}\| \|\mathbf{x}\|} \\ &\geq \left(\inf_{\mathbf{x} \in \mathbb{R}_*^p} \frac{\|\mathbf{MN} \mathbf{x}\|}{\|\mathbf{N} \mathbf{x}\|} \right) \left(\inf_{\mathbf{x} \in \mathbb{R}_*^p} \frac{\|\mathbf{N} \mathbf{x}\|}{\|\mathbf{x}\|} \right) \\ &\geq \left(\inf_{\mathbf{x} \in \mathbb{R}_*^n} \frac{\|\mathbf{M} \mathbf{x}\|}{\|\mathbf{x}\|} \right) \left(\inf_{\mathbf{x} \in \mathbb{R}_*^p} \frac{\|\mathbf{N} \mathbf{x}\|}{\|\mathbf{x}\|} \right) \\ &= \sigma_{\min}(\mathbf{M})\sigma_{\min}(\mathbf{N}) \end{aligned}$$

If, instead, $\mathbf{A} \in \mathbb{R}^{m \times n}$ with $m \leq n$, then

$$\sigma_{\min}(\mathbf{A}) = \sqrt{\lambda_{\min}(\mathbf{A} \mathbf{A}^T)} = \sqrt{\inf_{\mathbf{x} \in \mathbb{R}_*^m} \frac{\mathbf{x}^T \mathbf{A} \mathbf{A}^T \mathbf{x}}{\mathbf{x}^T \mathbf{x}}} = \inf_{\mathbf{x} \in \mathbb{R}_*^m} \frac{\|\mathbf{A}^T \mathbf{x}\|}{\|\mathbf{x}\|} \quad (34)$$

Similarly, if the null space of \mathbf{M}^T is non trivial, then $\sigma_{\min}(\mathbf{M}^T) = \sigma_{\min}(\mathbf{M}) = 0$, and the inequality holds. Otherwise, we have $\mathbf{M}^T \mathbf{x} \neq 0$, $\forall \mathbf{x} \in \mathbb{R}_*^m$,

$$\begin{aligned} \sigma_{\min}(\mathbf{MN}) &= \inf_{\mathbf{x} \in \mathbb{R}_*^m} \frac{\|\mathbf{N}^T \mathbf{M}^T \mathbf{x}\|}{\|\mathbf{x}\|} \\ &= \inf_{\mathbf{x} \in \mathbb{R}_*^m} \frac{\|\mathbf{N}^T \mathbf{M}^T \mathbf{x}\| \|\mathbf{M}^T \mathbf{x}\|}{\|\mathbf{M}^T \mathbf{x}\| \|\mathbf{x}\|} \\ &\geq \left(\inf_{\mathbf{x} \in \mathbb{R}_*^m} \frac{\|\mathbf{N}^T \mathbf{M}^T \mathbf{x}\|}{\|\mathbf{M}^T \mathbf{x}\|} \right) \left(\inf_{\mathbf{x} \in \mathbb{R}_*^m} \frac{\|\mathbf{M}^T \mathbf{x}\|}{\|\mathbf{x}\|} \right) \\ &\geq \left(\inf_{\mathbf{x} \in \mathbb{R}_*^n} \frac{\|\mathbf{N}^T \mathbf{x}\|}{\|\mathbf{x}\|} \right) \left(\inf_{\mathbf{x} \in \mathbb{R}_*^m} \frac{\|\mathbf{M}^T \mathbf{x}\|}{\|\mathbf{x}\|} \right) \\ &= \sigma_{\min}(\mathbf{N})\sigma_{\min}(\mathbf{M}) \end{aligned}$$

which concludes the proof. \square

Lemma 2. 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 $i \in \llbracket 1, p \rrbracket$, and either $n_0 \leq n_1 \leq \dots \leq n_p$ or $n_0 \geq n_1 \geq \dots \geq n_p$, then the product $\mathbf{M}_1 \mathbf{M}_2 \dots \mathbf{M}_p$ 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 1, 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 . \square

C.4 Proof of Proposition 2

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 1, 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 parameterize the feature extractors, at least in the overcomplete case.

Proposition 2. 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 $\mathbf{y}^0, \dots, \mathbf{y}^{d_L}$ such that the matrix $\mathbf{R} = (\mathbf{g}(\mathbf{y}^1) - \mathbf{g}(\mathbf{y}^0), \dots, \mathbf{g}(\mathbf{y}^{d_L}) - \mathbf{g}(\mathbf{y}^0))$ 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, \dots, g_{d_L} are not linearly independent. However, this is in contradiction with the result of Lemma 8, which stipulates that g_1, \dots, 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 $\mathbf{y}^0, \dots, \mathbf{y}^{d_L}$ such that the matrix $\mathbf{R} = (\mathbf{g}(\mathbf{y}^1) - \mathbf{g}(\mathbf{y}^0), \dots, \mathbf{g}(\mathbf{y}^{d_L}) - \mathbf{g}(\mathbf{y}^0))$ is invertible. \square

Proof under weaker conditions Note that the proof argument used for the overcomplete case can be used for the undercomplete as well. This same argument can be proved for ReLU as the nonlinearity instead of LeakyReLU. We chose to give the proof for, and suggest to use the latter because it is needed for Proposition 1.

Lemma 3. 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

$$\mathbf{B} = \left(\begin{array}{c|c} \mathbf{A} & \begin{matrix} \gamma_1 \\ \vdots \\ \gamma_{n-1} \\ \lambda \end{matrix} \\ \hline \mathbf{a}_n & 1 \end{array} \right) \quad (35)$$

is invertible for any choice of $\gamma_1, \dots, \gamma_{n-1}$, and for $\lambda \neq 1$.

Proof. Denote by \mathbf{b}_i the i -th row of \mathbf{B} . Let $\alpha_1, \dots, \alpha_{n+1}$ such that

$$\sum_{i=1}^{n+1} \alpha_i \mathbf{b}_i = 0 \quad (36)$$

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 (36), 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. \square

Lemma 4. Consider n affine functions $f_i : \mathbf{x} \in \mathbb{R}^d \mapsto \mathbf{a}_i^T \mathbf{x} + b_i$, 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 $\mathcal{H}_1, \dots, \mathcal{H}_n$ 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 $S_{i,j}^n = \text{sign}_{\mathbf{x} \in \mathcal{H}_i}(f_j(\mathbf{x}))$ 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 $\mathcal{H}_1 = \{\mathbf{x} \in \mathbb{R}^d | \mathbf{a}_1^T \mathbf{x} + b_1 > 0, \mathbf{a}_2^T \mathbf{x} + b_2 > 0\}$ and $\mathcal{H}_2 = \{\mathbf{x} \in \mathbb{R}^d | \mathbf{a}_1^T \mathbf{x} + b_1 > 0, \mathbf{a}_2^T \mathbf{x} + b_2 < 0\}$ are not empty.

Second, suppose that there exists n regions $\mathcal{H}_1, \dots, \mathcal{H}_n$ such that the the matrix \mathbf{S}^n is invertible. Consider the affine function $f_{n+1} = \mathbf{a}_{n+1}^T \mathbf{x} + b_{n+1}$. 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 $\tilde{S}_{n,j}^n = \text{sign}_{\mathbf{x} \in \tilde{\mathcal{H}}_n}(f_j(\mathbf{x}))$ and $\tilde{S}_{i,j}^n = S_{i,j}^n$. Because $\tilde{\mathcal{H}}_n \subset \mathcal{H}_n$, we have that $\text{sign}_{\mathbf{x} \in \mathcal{H}_n}(f_j(\mathbf{x})) = \text{sign}_{\mathbf{x} \in \tilde{\mathcal{H}}_n}(f_j(\mathbf{x}))$ 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 $\text{sign}_{\mathbf{x} \in \mathcal{H}_n}(f_j(\mathbf{x})) = \text{sign}_{\mathbf{x} \in \tilde{\mathcal{H}}_n}(f_j(\mathbf{x}))$. 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

$$\mathbf{S}^{n+1} = \left(\begin{array}{c|c} \mathbf{S}^n & \begin{matrix} u_1 \\ \vdots \\ u_{n-1} \\ -1 \end{matrix} \\ \hline \mathbf{s}_n^n & 1 \end{array} \right) \quad (37)$$

where $u_i = \text{sign}_{\mathbf{x} \in \mathcal{H}_i} f_{n+1}(\mathbf{x})$ and \mathbf{s}_n^n is the n -th line of \mathbf{S}^n . According to Lemma 3, \mathbf{S}^{n+1} is invertible, which achieves the proof. \square

Lemma 5. 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 $g_i : \mathbf{x} \in \mathbb{R}^d \mapsto h(\mathbf{a}_i^T \mathbf{x} + b_i)$, 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 $f_i = \mathbf{a}_i^T \mathbf{x} + b_i$ 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 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 $\mathbf{S}^n = (\text{sign}_{\mathbf{x} \in \mathcal{H}_i}(f_j(\mathbf{x})))_{i,j}$ 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 $\text{sign}(x) = \lambda$ if $x < 0$).

Now consider $\alpha_1, \dots, \alpha_n$ such that

$$\sum_{i=1}^n \alpha_i g_i = 0 \quad (38)$$

Let $k \in \llbracket 1, n \rrbracket$, and evaluate this equation at $\mathbf{x} \in \mathcal{H}_k$. After taking the gradient with respect to \mathbf{x} , we get

$$\sum_i (\mathbf{1}_{\mathbf{x} \in \mathcal{H}_k, f_i(\mathbf{x}) \geq 0} + \lambda \mathbf{1}_{\mathbf{x} \in \mathcal{H}_k, f_i(\mathbf{x}) < 0}) \alpha_i \mathbf{a}_i = 0 \quad (39)$$

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 (39) as:

$$\tilde{\mathbf{s}}_k^T \mathbf{e}_l = 0 \quad (40)$$

Collating these equations for a fixed l and $k \in \llbracket 1, n \rrbracket$, we get

$$\tilde{\mathbf{S}} \mathbf{e}_l = 0 \quad (41)$$

which implies that $\mathbf{e}_l = 0$ because \mathbf{S} is invertible. In particular, $\alpha_i a_{i,l} = 0$ for all $i \in \llbracket 1, n \rrbracket$ and $l \in \llbracket 1, d \rrbracket$. This implies that $\mathbf{A}_J^T \boldsymbol{\alpha}_J = 0$, where $J \subset \llbracket 1, n \rrbracket$ of size $\min(n, d)$, $\mathbf{A}_J = (a_{i,l})_{i \in J, l \in \llbracket 1, d \rrbracket} \in \mathbb{R}^{d \times d}$ 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. \square

Lemma 6. 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 $g_i : \mathbf{x} \in \mathbb{R}^d \mapsto \mathbf{c}_i^T \mathbf{f}(\mathbf{x}) + d_i$, 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 $T_{i,j}^m = \text{sign}_{\mathbf{x} \in \mathcal{K}_i}(g_j(\mathbf{x}))$ is invertible.

Proof. Denote by $\tilde{\mathbf{c}}_i(\mathbf{x})$ the generalized slope of the p.w. 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 that 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 $J \subset \llbracket 1, m \rrbracket$ of size $\min(m, n)$.

If $m > n$, choose a subset $J \in \llbracket 1, m \rrbracket$ 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: $\sum_j (\sum_{i \in J} \alpha_i c_{i,j}) \mathbf{a}_j(\mathbf{x}) = 0$. 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 $j \in \llbracket 1, n \rrbracket$. 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 $J = \llbracket 1, m \rrbracket$, and conclude.

The rest of the proof follows the same argument of the proof of Lemma 4: 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 p.w. 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. \square

Lemma 7. 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 $g_i : \mathbf{x} \in \mathbb{R}^d \mapsto h(\mathbf{c}_i^T \mathbf{f}(\mathbf{x}) + d_i)$, 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 $\tilde{g}_i = \mathbf{c}_i^T \mathbf{f} + d_i$ such that $g_i = h(\tilde{g}_i)$. The assumptions of Lemma 6 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 $\mathbf{T}^m = (\text{sign}_{\mathbf{x} \in \mathcal{K}_i}(\tilde{g}_j(\mathbf{x})))_{i,j}$ 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:

$$\sum_j \left(\sum_i \alpha_i (\mathbf{1}_{\tilde{g}_i \geq 0}(\mathbf{x}) + \lambda \mathbf{1}_{\tilde{g}_i < 0}(\mathbf{x})) c_{i,j} \right) \alpha_j(\mathbf{x}) = 0 \quad (42)$$

Since $\mathbf{a}_1, \dots, \mathbf{a}_n$ are independent, we conclude that $\sum_i \alpha_i (\mathbf{1}_{\tilde{g}_i \geq 0}(\mathbf{x}) + \lambda \mathbf{1}_{\tilde{g}_i < 0}(\mathbf{x})) c_{i,j}$ for all $j \in \llbracket 1, m \rrbracket$. This in turn implies that

$$\sum_i \alpha_i (\mathbf{1}_{\tilde{g}_i \geq 0}(\mathbf{x}) + \lambda \mathbf{1}_{\tilde{g}_i < 0}(\mathbf{x})) \mathbf{c}_i = 0 \quad (43)$$

Let $k \in \llbracket 1, m \rrbracket$, and evaluate the last equation at $\mathbf{x} \in \mathcal{K}_k$:

$$\sum_i (\mathbf{1}_{\mathbf{x} \in \mathcal{H}_k, f_i(\mathbf{x}) \geq 0} + \lambda \mathbf{1}_{\mathbf{x} \in \mathcal{H}_k, f_i(\mathbf{x}) < 0}) \alpha_i \mathbf{c}_i = 0 \quad (44)$$

This last equation is similar to equation (39), and we can use the same argument used for the proof of Lemma 5 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. \square

Lemma 8. 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 5, we conclude that f_1, \dots, f_n are independent. If we suppose the result hold for $L \geq 2$, we can use Lemma 7 to prove that it also holds for $L + 1$. Finally, since all layers satisfy the same conditions, the conclusion also applies to intermediate layers. \square

C.5 Proof of Theorem 2

We will decompose Theorem 2 into two sub-theorems, which will make the proof easier to understand, but also more adaptable into future work. Each of these sub-theorems corresponds to one of the assumptions.

C.5.1 Positive features

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.

Theorem 2a. Assume the assumptions of Theorem 1 hold. Further assume that $n \leq d$, and that each f_i has a global minimum that is reached at least in the limit, and the feature extractor $\mathbf{f} = (f_1, \dots, f_n)$ is surjective onto the set that is defined by the lower bounds of the f_i . Then

$$p_{\theta}(\mathbf{x}|\mathbf{y}) = p_{\theta'}(\mathbf{x}|\mathbf{y}) \implies \theta \sim_s \theta'$$

where \sim_s is defined as follows:

$$\theta \sim_s \theta' \Leftrightarrow \forall i, f_{i,\theta}(\mathbf{x}) = a_i f_{\sigma(i),\theta'}(\mathbf{x}) + b_i \quad (45)$$

where σ is a permutation of $\llbracket 1, n \rrbracket$, a_i is a non zero scalar and b_i is a scalar.

Proof. Consider two different parameters θ and $\tilde{\theta}$ such that:

$$p_{\theta}(\mathbf{x}|\mathbf{y}) = p_{\tilde{\theta}}(\mathbf{x}|\mathbf{y}) \quad (46)$$

To simplify notations, denote by $\mathbf{f} = \mathbf{f}_{\theta}$ and $\tilde{\mathbf{f}} = \mathbf{f}_{\tilde{\theta}}$. We start the proof from the conclusion of Theorem 1, since its assumptions hold:

$$\mathbf{f}(\mathbf{x}) = \mathbf{A}\tilde{\mathbf{f}}(\mathbf{x}) + \mathbf{c} \quad (47)$$

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 $\mathbf{A}\mathbf{I}$ 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:

$$\mathbf{a}_i^T \mathbf{b}_j = \delta_{ij} \quad (48)$$

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 non-zero 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 non-zeros 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 $\tilde{\mathbf{B}}$. Thus, each \mathbf{a}_i can have only one non-zero entry, which, together with the invertibility of \mathbf{A} , proves it is a scaled permutation matrix.

Thus, there exists a permutation σ of $\llbracket 1, n \rrbracket$, such that $f_i(\mathbf{x}) = a_{i, \sigma(i)} \tilde{f}_{\sigma(i)}(\mathbf{x}) + c_i$, which concludes the proof. \square

C.5.2 Augmented features

Theorem 2b. Assume that $n \leq d$, and that:

1. The feature extractor \mathbf{f} is differentiable and surjective, and its Jacobian $\mathbf{J}_{\mathbf{f}}$ is full rank.
2. There exist $2n + 1$ points $\mathbf{y}^0, \dots, \mathbf{y}^{2n}$ such that the matrix

$$\tilde{\mathbf{R}} = (\tilde{\mathbf{g}}(\mathbf{y}^1) - \tilde{\mathbf{g}}(\mathbf{y}^0), \dots, \tilde{\mathbf{g}}(\mathbf{y}^{2n}) - \tilde{\mathbf{g}}(\mathbf{y}^0)) \quad (49)$$

of size $2n \times 2n$ is invertible.

Then

$$p_{\boldsymbol{\theta}}(\mathbf{x}|\mathbf{y}) = p_{\boldsymbol{\theta}'}(\mathbf{x}|\mathbf{y}) \implies \boldsymbol{\theta} \sim_s \boldsymbol{\theta}'$$

where \sim_s is defined in (45).

Proof. Similarly to the proof of Theorem 2a, 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 2.2.2.

Consider two different parameters $\boldsymbol{\theta}$ and $\tilde{\boldsymbol{\theta}}$ such that:

$$p_{\boldsymbol{\theta}}(\mathbf{x}|\mathbf{y}) = p_{\tilde{\boldsymbol{\theta}}}(\mathbf{x}|\mathbf{y}) \quad (50)$$

Since we have similar assumptions to Theorem 1, 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 $\mathbf{H}(\mathbf{f}_{\tilde{\boldsymbol{\theta}}})$:

$$\mathbf{H}(\mathbf{f}_{\boldsymbol{\theta}}(\mathbf{x})) = \mathbf{A}\mathbf{H}(\mathbf{f}_{\tilde{\boldsymbol{\theta}}}(\mathbf{x})) + \mathbf{c} \quad (51)$$

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:

$$\begin{pmatrix} \mathbf{f}_{\boldsymbol{\theta}}(\mathbf{x}) \\ \mathbf{f}_{\boldsymbol{\theta}}^2(\mathbf{x}) \end{pmatrix} = \begin{pmatrix} \mathbf{A}^{(1)} & \mathbf{A}^{(2)} \\ \mathbf{A}^{(3)} & \mathbf{A}^{(4)} \end{pmatrix} \begin{pmatrix} \mathbf{f}_{\tilde{\boldsymbol{\theta}}}(\mathbf{x}) \\ \mathbf{f}_{\tilde{\boldsymbol{\theta}}}^2(\mathbf{x}) \end{pmatrix} + \begin{pmatrix} \boldsymbol{\alpha} \\ \boldsymbol{\beta} \end{pmatrix} \quad (52)$$

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:

$$f_i(\mathbf{x}) = \sum_{j=1}^n A_{ij}^{(1)} h_j(\mathbf{x}) + A_{ij}^{(2)} h_j^2(\mathbf{x}) + \alpha_i \quad (53)$$

and the last n lines are:

$$f_i^2(\mathbf{x}) = \sum_{j=1}^n A_{ij}^{(3)} h_j(\mathbf{x}) + A_{ij}^{(4)} h_j^2(\mathbf{x}) + \beta_i \quad (54)$$

Fix an index i in equations (53) and (54). 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 (54) and (53) after squaring, we get, using our new notations:

$$\left(\sum_j a_j h_j(\mathbf{x}) + b_j h_j^2(\mathbf{x}) + \alpha \right)^2 = \sum_j c_j h_j(\mathbf{x}) + d_j h_j^2(\mathbf{x}) + \beta \quad (55)$$

Step 1 First, since \mathbf{h} is surjective, there exists a point where it is equal to zero. Evaluating equation (55) at this point shows that $\beta = \alpha^2$.

Step 2 Second, the left hand side of equation (55) 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 (55) at each of these points, we get

$$(a_l y + b_l y^2 + \alpha)^2 = c_l y + d_l y^2 + \beta \quad (56)$$

Divide both sides of this equation by y^4 , then take the limit $y \rightarrow \infty$. 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 $l \in \llbracket 1, n \rrbracket$, we can show that $\mathbf{b} = 0$.

Step 3 So far, we've shown that (55) becomes, after expanding the square in the lhs, and writing $\sum_j a_j h_j(\mathbf{x}) = \mathbf{a}^T \mathbf{h}(\mathbf{x})$:

$$(\mathbf{a}^T \mathbf{h}(\mathbf{x}))^2 + 2\alpha \mathbf{a}^T \mathbf{h}(\mathbf{x}) + \alpha^2 = \sum_j c_j h_j(\mathbf{x}) + d_j h_j^2(\mathbf{x}) + \alpha^2 \quad (57)$$

Let's again consider the vectors $\mathbf{e}_l(y)$ from earlier, and their preimages $\mathbf{x}_l(y)$. By evaluating (57) at the points $\mathbf{x}_l(y)$, we get

$$a_l^2 y^2 + 2\alpha a_l y + \alpha^2 = c_l y + d_l y^2 + \alpha^2 \quad (58)$$

Divide both sides by y , and take the limit $y \rightarrow 0$. The lhs converges to $2\alpha a_l$, while the rhs converges to c_l . Since this is valid for all $l \in \llbracket 1, n \rrbracket$, 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 (57), and writing $\sum_j d_j h_j^2(\mathbf{x}) = \mathbf{h}(\mathbf{x})^T \text{diag}(\mathbf{d}) \mathbf{h}(\mathbf{x})$, we are left with:

$$(\mathbf{a}^T \mathbf{h}(\mathbf{x}))^2 = \mathbf{h}(\mathbf{x})^T \text{diag}(\mathbf{d}) \mathbf{h}(\mathbf{x}) \quad (59)$$

By applying the trace operator to both sides of this equation, and rearranging terms, we get

$$\text{trace}((\mathbf{a}\mathbf{a}^T - \text{diag}(\mathbf{d})) \mathbf{h}(\mathbf{x}) \mathbf{h}(\mathbf{x})^T) = 0 \quad (60)$$

which is of the form $\text{trace}(\mathbf{C}^T \mathbf{B}(\mathbf{x})) = 0$. 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 non-zero 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 non-zero entry equal to 1 at row i and column j .

Now since \mathbf{h} is surjective, there exists \mathbf{x}_j such that $\mathbf{h}(\mathbf{x}_j) = \mathbf{e}_j$, and $\mathbf{h}(\mathbf{x}_j) \mathbf{h}(\mathbf{x}_j)^T = \mathbf{e}_j \mathbf{e}_j^T = \mathbf{E}_{jj}$. 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 $\mathbf{h}(\mathbf{x}_{j,l}) \mathbf{h}(\mathbf{x}_{j,l})^T = (\mathbf{e}_j + \mathbf{e}_l)(\mathbf{e}_j + \mathbf{e}_l)^T = \mathbf{E}_{jj} + \mathbf{E}_{ll} + (\mathbf{E}_{il} + \mathbf{E}_{li})$, which is a matrix in \mathcal{S}_n that is linearly independent of all \mathbf{E}_{jj} , and all other $(\mathbf{e}_s + \mathbf{e}_t)(\mathbf{e}_s + \mathbf{e}_t)^T$ where $(s, t) \neq (j, l)$ because they have non-zero entries at different rows and columns.

We have then found a total of $\frac{n(n+1)}{2}$ different vectors $(\mathbf{x}_1, \dots, \mathbf{x}_n, \mathbf{x}_{1,2}, \dots, \mathbf{x}_{n-1,n})$ such that their images by $\mathbf{h}\mathbf{h}^T$ form a basis of \mathcal{S}_n . If we now evaluate equation (60) at each of these points, we find that the matrix $\mathbf{a}\mathbf{a}^T - \text{diag}(\mathbf{d})$ is orthogonal to a basis of \mathcal{S}_n , which implies that it is necessarily equal to 0. This in turn implies that $\mathbf{a}\mathbf{a}^T$ 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 non-zero.

Step 5 So far, we have proven that, among other things, $A_{i,j}^{(2)} = 0$ for all i, j . We now go back to equation (53), which we can write as:

$$\mathbf{f}(\mathbf{x}) = \mathbf{A}^{(1)} \mathbf{h}(\mathbf{x}) + \alpha \quad (61)$$

Both \mathbf{f} and \mathbf{h} are differentiable, and according to assumption 2, $J_{\mathbf{f}}$ 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 $a_j = A_{ij}^{(1)}$. This is valid for all $i \in \llbracket 1, n \rrbracket$. Now since $\mathbf{A}^{(1)}$ is invertible, the non-zero entry $A_{ij}^{(1)}$ has to be in a different column for each row, otherwise some rows will be linearly dependent. Thus, there exists a permutation σ of $\llbracket 1, n \rrbracket$, such that $A_{i\sigma(i)}^{(1)} \neq 0$, and we deduce that

$$f_i(\mathbf{x}) = a_{\sigma(i)} h_{\sigma(i)}(\mathbf{x}) + \alpha_i \quad (62)$$

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. \square

C.6 Proof of Theorem 3

Theorem 3. *Let $p(\mathbf{x}|\mathbf{y})$ be a conditional probability density. Assume that \mathcal{X} and \mathcal{Y} are compact Hausdorff spaces, and that $p(\mathbf{x}|\mathbf{y}) > 0$ almost surely $\forall (\mathbf{x}, \mathbf{y}) \in \mathcal{X} \times \mathcal{Y}$. Then for each $\varepsilon > 0$, there exists $(\boldsymbol{\theta}, n) \in \Theta \times \mathbb{N}$, where n is the dimension of the feature extractor, such that $\sup_{\mathbf{x}, \mathbf{y}} |p_{\boldsymbol{\theta}}(\mathbf{x}|\mathbf{y}) - p(\mathbf{x}|\mathbf{y})| < \varepsilon$.*

Proof. We consider here two cases.

Continuous auxiliary variable Recall the form of our model:

$$\log p_{\boldsymbol{\theta}}(\mathbf{x}|\mathbf{y}) = -\log Z(\mathbf{y}) - \mathbf{f}(\mathbf{x})^T \mathbf{g}(\mathbf{y}) \quad (63)$$

By parameterizing each of f_i, g_i as neural networks, these functions can approximate continuous function on their respective domains arbitrarily well. According to Lemma 9, this implies that any continuous function on $\mathcal{X} \times \mathcal{Y}$ can be approximated arbitrarily well by a term of the form $-\mathbf{f}(\mathbf{x})^T \mathbf{g}(\mathbf{y})$.

Thus, 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:

$$\sup_{\mathbf{x}, \mathbf{y}} \left| \log p(\mathbf{x}|\mathbf{y}) + \sum_{i=1}^n f_i(\mathbf{x}; \boldsymbol{\theta}) g_i(\mathbf{y}; \boldsymbol{\theta}) \right| < \varepsilon \quad (64)$$

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:

$$\sup_{\mathbf{x}, \mathbf{y}} |p(\mathbf{x}|\mathbf{y}) - p_{\boldsymbol{\theta}}(\mathbf{x}|\mathbf{y})| < K\varepsilon \quad (65)$$

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 9 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. \square

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 $\mathcal{F}_{\mathcal{X}} \otimes \mathcal{F}_{\mathcal{Y}} := \{\sum_{i=1}^n f_i g_i | n \in \mathbb{N}, f_i \in \mathcal{F}_{\mathcal{X}}, g_i \in \mathcal{F}_{\mathcal{Y}}\}$ be the set of all linear combinations of products of functions from $\mathcal{F}_{\mathcal{X}}$ and $\mathcal{F}_{\mathcal{Y}}$ to \mathbb{R} . The energy function defining our model belongs to this last set. Finally, universal approximation is expressed in terms of 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 in $\mathcal{C}(\mathcal{X})$, we can always find a limit of a sequence of functions of $\mathcal{F}_{\mathcal{X}}$ that converges to it. We mathematically express density by writing $\overline{\mathcal{F}_{\mathcal{X}}} = \mathcal{C}(\mathcal{X})$.

Let $\mathcal{F}_{\mathcal{X}}$ (respectively $\mathcal{F}_{\mathcal{Y}}$) be the set of deep neural networks with input in \mathcal{X} (respectively in \mathcal{Y}). The universal approximation capability is summarised in the following Lemma.

Lemma 9 (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 of Theorem 3.
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 5.

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

$$\|f_i g_i - \tilde{f}_i \tilde{g}_i\|_{\infty} = \|f_i g_i - f_i \tilde{g}_i + f_i \tilde{g}_i - \tilde{f}_i \tilde{g}_i\|_{\infty} \leq \|f_i\|_{\infty} \|g_i - \tilde{g}_i\|_{\infty} + \|\tilde{g}_i\|_{\infty} \|f_i - \tilde{f}_i\|_{\infty} < \frac{\varepsilon}{k} \quad (66)$$

Using this, we conclude that

$$\|h - \sum_{i=1}^k \tilde{f}_i \tilde{g}_i\|_{\infty} \leq \sum_{i=1}^k \|f_i g_i - \tilde{f}_i \tilde{g}_i\|_{\infty} < \varepsilon \quad (67)$$

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 3).
- (iv) $\mathcal{C}(\mathcal{X}) \otimes \mathcal{C}(\mathcal{Y})$ separates points in $\mathcal{X} \times \mathcal{Y}$ (see Definition 3).

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-Weirstrass 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 $\overline{\mathcal{F}_{\mathcal{X}} \otimes \mathcal{F}_{\mathcal{Y}}} = \mathcal{C}(\mathcal{X} \times \mathcal{Y})$. \square

Definition 3. 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) $1 \in \mathcal{A}$.
 - (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 5 (Stone-Weirstrass). 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 $\mathcal{C}(K)$.

Proof. A proof to this theorem can be found in many references, for instance Brosowski and Deutsch (1981). \square

D 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:

$$\mathbf{z} \sim p(\mathbf{z}|\mathbf{y}) \quad (68)$$

$$\mathbf{x} = \mathbf{h}(\mathbf{z}) \quad (69)$$

We assume the latent variable in the IMCA model has a density of the form

$$p(\mathbf{z}|\mathbf{y}) = \mu(\mathbf{z}) e^{\sum_i \mathbf{T}_i(z_i)^T \boldsymbol{\lambda}_i(\mathbf{y}) - \Gamma(\mathbf{y})} \quad (70)$$

where μ is not necessarily factorial.

Further, we will suppose that the density $p(\mathbf{z}|\mathbf{y})$ belongs to the following subclass of the exponential families, introduced by [Khemakhem et al. \(2020\)](#):

Definition 4 (Strongly exponential). *We say that an exponential family distribution is strongly exponential if for any subset \mathcal{X} of \mathbb{R} the following is true:*

$$(\exists \boldsymbol{\theta} \in \mathbb{R}^k \mid \forall x \in \mathcal{X}, \langle \mathbf{T}(x), \boldsymbol{\theta} \rangle = \text{const}) \implies (\Lambda(\mathcal{X}) = 0 \text{ or } \boldsymbol{\theta} = 0) \quad (71)$$

where Λ is the Lebesgue measure.

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

$$p(\mathbf{z}|\mathbf{y}) = \mu(\mathbf{z}) e^{\sum_{i=1}^n \mathbf{T}_i(z_i)^T \boldsymbol{\lambda}_i(\mathbf{y}) - \Gamma(\mathbf{y})} \quad (72)$$

The term $e^{\sum_{i=n+1}^d \mathbf{T}_i(z_i)^T \boldsymbol{\lambda}_i}$ is absorbed into $\mu(\mathbf{z})$. 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

$$p_{\boldsymbol{\theta}}(\mathbf{x}|\mathbf{y}) = Z(\mathbf{y}; \boldsymbol{\theta})^{-1} \exp(-\mathbf{H}(\mathbf{f}_{\boldsymbol{\theta}}(\mathbf{x}))^T \mathbf{g}_{\boldsymbol{\theta}}(\mathbf{y})) \quad (73)$$

where $\mathbf{H}(\mathbf{f}(\mathbf{x})) = (\mathbf{H}_1(f_1(\mathbf{x})), \dots, \mathbf{H}_d(f_d(\mathbf{x})))$, 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.

D.1 Assumptions

We prove dimensionality reduction capability in Theorem 6. We will decompose Theorem 4 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.

- (i) The observed data follows the exponential IMCA model of equations (68)-(70).
- (ii) The mixing function $\mathbf{h} : \mathbb{R}^d \rightarrow \mathbb{R}^d$ in (69) is invertible.
- (iii) The sufficient statistics \mathbf{T}_i in (70) 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.
- (iv) There exist $k + 1$ distinct points $\mathbf{y}^0, \dots, \mathbf{y}^k$ such that the matrix

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

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

- (v) We fit the model (73) 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 $(\dim(\mathbf{T}_1), \dots, \dim(\mathbf{T}_d))$ without replacement.

- (vi) The sufficient statistic in (70) 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 (73) 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 (73) 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 (72).
- (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.

D.2 Lemmas

We rely on the following Lemmas from Khemakhem et al. (2020), which we state below in the interest of completeness.

Lemma 10. Consider an exponential family distribution with $k \geq 2$ components. Then the components of the sufficient statistic \mathbf{T} are linearly independent.

Lemma 11. Consider a 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 $1 \leq i \leq k$.

Lemma 12. Consider a strongly exponential distribution of size $k \geq 2$ with sufficient statistic $\mathbf{T}(x) = (T_1(x), \dots, T_k(x))$. Further assume that \mathbf{T} is differentiable almost everywhere. Then there exist k distinct values x_1 to x_k such that $(\mathbf{T}'(x_1), \dots, \mathbf{T}'(x_k))$ are linearly independent in \mathbb{R}^k .

Lemma 13. Consider a strongly exponential distribution of size $k \geq 2$ with sufficient statistic \mathbf{T} . Further assume that \mathbf{T} is twice differentiable almost everywhere. Then

$$\dim \left(\text{span} \left((T'_i(x), T''_i(x))^T, 1 \leq i \leq k \right) \right) \geq 2 \quad (74)$$

almost everywhere on \mathbb{R} .

Lemma 14. Consider n 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 consider that the sufficient statistics are twice differentiable. Define the vectors $\mathbf{e}^{(j,i)} \in \mathbb{R}^{2n}$, such that $\mathbf{e}^{(j,i)} = (0, \dots, 0, T'_{j,i}, T''_{j,i}, 0, \dots, 0)$, where the non-zero entries are at indices $(2j, 2j+1)$. Let $\mathbf{x} := (x_1, \dots, x_n) \in \mathbb{R}^n$. Then the matrix $\bar{\mathbf{e}}(\mathbf{x}) := (\mathbf{e}^{(1,1)}(x_1), \dots, \mathbf{e}^{(1,k)}(x_1), \dots, \mathbf{e}^{(n,1)}(x_n), \dots, \mathbf{e}^{(n,k)}(x_n))$ of size $(2n \times nk)$ has rank $2n$ almost everywhere on \mathbb{R}^n .

D.3 Proofs

As mentioned above, we decompose Theorem 4 into two smaller results, summarized in what follows by Theorems 4a and 4b.

Theorem 4a. Assume assumptions (i)-(v) hold. Then, after convergence of our model $p_\theta(\mathbf{x}|\mathbf{y})$ to the true density $p(\mathbf{x}|\mathbf{y})$, we can recover the latent variables up to an invertible linear transformation and point-wise nonlinearities, i.e.

$$\mathbf{H}(\mathbf{f}(\mathbf{x})) = \mathbf{A}\mathbf{T}(\mathbf{z}) + \mathbf{b} \quad (75)$$

where \mathbf{A} is an invertible matrix.

Proof. We fit our density model (73) 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 $(\dim(\mathbf{T}_1), \dots, \dim(\mathbf{T}_d))$, as per assumption (v):

$$Z(\mathbf{y})^{-1} \exp \mathbf{H}(\mathbf{f}(\mathbf{x}))^T \mathbf{g}(\mathbf{y}) = p(\mathbf{x}|\mathbf{y}) \quad (76)$$

by doing the change of variable $\mathbf{x} = \mathbf{h}(\mathbf{z})$, taking the log on both sides, we get:

$$-\log Z(\mathbf{y}) + \mathbf{H}(\mathbf{f}(\mathbf{x}))^T \mathbf{g}(\mathbf{y}) = \log p(\mathbf{z}|\mathbf{y}) - \log |\det \mathbf{J}_{\mathbf{h}^{-1}}(\mathbf{x})| \quad (77)$$

$$= \log \mu(\mathbf{h}^{-1}(\mathbf{x})) + \mathbf{T}(\mathbf{z})^T \boldsymbol{\lambda}(\mathbf{y}) - \Gamma(\mathbf{y}) - \log |\det \mathbf{J}_{\mathbf{h}^{-1}}(\mathbf{x})| \quad (78)$$

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 $\bar{\boldsymbol{\lambda}}(\mathbf{y}) = \boldsymbol{\lambda}(\mathbf{y}) - \boldsymbol{\lambda}(\mathbf{y}_0)$, $\bar{\Gamma}(\mathbf{y}) = \Gamma(\mathbf{y}) - \Gamma(\mathbf{y}_0)$, $\bar{\mathbf{g}}(\mathbf{y}) = \mathbf{g}(\mathbf{y}) - \mathbf{g}(\mathbf{y}_0)$ and $\bar{Z}(\mathbf{y}) = \log Z(\mathbf{y}) - \log Z(\mathbf{y}_0)$. We plug each of those \mathbf{y}_l in (78) to obtain $k+1$ such equations. We subtract the first equation for \mathbf{y}_0 from the remaining k equations to get for $l = 1, \dots, k$:

$$-\bar{Z}(\mathbf{y}_l) + \mathbf{H}(\mathbf{f}(\mathbf{x}))^T \bar{\mathbf{g}}(\mathbf{y}_l) = \mathbf{T}(\mathbf{z})^T \bar{\boldsymbol{\lambda}}(\mathbf{y}_l) - \bar{\Gamma}(\mathbf{y}_l) \quad (79)$$

The **crucial point** here is that the non factorial terms $\mu(\mathbf{g}(\mathbf{x}))$ and $\tilde{\mu}(\tilde{\mathbf{g}}(\mathbf{x}))$ 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 $\tilde{\mathbf{L}} := (\dots, \bar{\mathbf{g}}(\mathbf{y}_l), \dots)$. Define $\mathbf{b} = (\dots, \bar{Z}(\mathbf{y}_l) - \bar{\Gamma}(\mathbf{y}_l), \dots)$. Expressing (79) for all points \mathbf{y}_l in matrix form, we get:

$$\tilde{\mathbf{L}}^T \mathbf{H}(\mathbf{f}(\mathbf{x})) = \mathbf{L}^T \mathbf{T}(\mathbf{z}) + \mathbf{b} \quad (80)$$

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

$$\mathbf{T}(\mathbf{z}) = \mathbf{A} \mathbf{H}(\mathbf{f}(\mathbf{x})) + \mathbf{c} \quad (81)$$

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

To prove that \mathbf{A} is invertible, we first take the gradient of equation (81) 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 non-zero 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 12, 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 point into $k_{\max} := \max_i k_i$ vectors $\mathbf{z}^{(j)} := (z_1^{(j)}, \dots, z_d^{(j)})$, 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 (81) after differentiating it, and collate the dk_{\max} equations in vector form:

$$\mathbf{M} = \mathbf{A} \tilde{\mathbf{M}} \quad (82)$$

where $\mathbf{M} := (\dots, \mathbf{J}_{\mathbf{T}}(\mathbf{z}^{(j)}), \dots)$ and $\tilde{\mathbf{M}} := (\dots, \mathbf{J}_{\mathbf{H} \circ \mathbf{h}}(\mathbf{z}^{(j)}), \dots)$. 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. \square

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. (2020) even in this non-independent case as follows:

Theorem 4b. Assume assumptions (i)-(v) hold. Further assume that either assumptions (vi)-(viii) or assumptions (vi')-(viii') hold. Then equation (75) can be reduced to the component level, i.e. for each $i \in \llbracket 1, d \rrbracket$:

$$\mathbf{H}_i(f_i(\mathbf{x})) = A_i \mathbf{T}_{\gamma(i)}(z_{\gamma(i)}) + \mathbf{b}_i \quad (83)$$

where γ is a permutation of $\llbracket 1, d \rrbracket$ 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)-(viii) We suppose that $k_i \geq 2$, $\forall i$. The assumptions of Theorem 4a hold, and so we have

$$\mathbf{H}(\mathbf{f}(\mathbf{h}(\mathbf{z}))) = \mathbf{A} \mathbf{T}(\mathbf{z}) + \mathbf{c} \quad (84)$$

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), (viii)), we can invert this relation and write $\mathbf{z} = \mathbf{v}(\mathbf{y})$. We introduce the notations $v_i^s(\mathbf{y}) := \frac{\partial v_i}{\partial y_s}(\mathbf{y})$, $v_i^{st}(\mathbf{y}) := \frac{\partial^2 v_i}{\partial y_s \partial y_t}(\mathbf{y})$, $T'_{a,b}(z) = \frac{dT_{a,b}}{dz}(z)$, $T''_{a,b}(z) = \frac{d^2 T_{a,b}}{dz^2}(z)$ and $H'_{a,b}(y) = \frac{dH_{a,b}}{dy}(y)$. Each line of equation (84) can be written as:

$$H_{i,l}(y_i) = \sum_{a=1}^d \sum_{b=1}^{k_i} A_{i,l,a,b} T_{a,b}(v_a(\mathbf{y})) + c_{a,b} \quad (85)$$

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 (85) with respect to $y_s, s \leq d$:

$$\delta_{is} H'_{i,l}(y_i) = \sum_{a=1}^d \sum_{b=1}^{k_i} A_{i,l,a,b} T'_{a,b}(v_a(\mathbf{y})) v_a^s(\mathbf{y}) \quad (86)$$

and by differentiating (86) with respect to $y_t, s < t \leq d$:

$$0 = \sum_{a,b} A_{i,l,a,b} (T'_{a,b}(v_a(\mathbf{y})) v_a^{s,t}(\mathbf{y}) + T''_{a,b}(v_a(\mathbf{y})) v_a^s(\mathbf{y}) v_a^t(\mathbf{y})) \quad (87)$$

This equation is valid for all pairs $(s, t), t > s$. Define $\mathbf{B}_a(\mathbf{y}) := (v_a^{1,2}(\mathbf{y}), \dots, v_a^{d-1,d}(\mathbf{y})) \in \mathbb{R}^{\frac{d(d-1)}{2}}$, $\mathbf{C}_a(\mathbf{y}) := (v_a^1(\mathbf{y})v_a^2(\mathbf{y}), \dots, v_a^{d-1}(\mathbf{y})v_a^d(\mathbf{y})) \in \mathbb{R}^{\frac{d(d-1)}{2}}$, $\mathbf{M}(\mathbf{y}) := (\mathbf{B}_1(\mathbf{y}), \mathbf{C}_1(\mathbf{y}), \dots, \mathbf{B}_d(\mathbf{y}), \mathbf{C}_d(\mathbf{y}))$, $\mathbf{e}^{(a,b)} := (0, \dots, 0, T'_{a,b}, T''_{a,b}, 0, \dots, 0) \in \mathbb{R}^{2d}$, such that the non-zero entries are at indices $(2a, 2a+1)$ and $\bar{\mathbf{e}}(\mathbf{y}) := (\mathbf{e}^{(1,1)}(y_1), \dots, \mathbf{e}^{(1,k_1)}(y_1), \dots, \mathbf{e}^{(d,1)}(y_d), \dots, \mathbf{e}^{(d,k_d)}(y_d)) \in \mathbb{R}^{2d \times k}$. Then by grouping equation (87) for all valid pairs (s, t) and pairs (i, l) and writing it in matrix form, we get:

$$\mathbf{M}(\mathbf{y}) \bar{\mathbf{e}}(\mathbf{y}) \mathbf{A} = 0 \quad (88)$$

Now by Lemma 14, we know that $\bar{\mathbf{e}}(\mathbf{y})$ has rank $2d$ almost surely on \mathcal{Z} . Since \mathbf{A} is invertible, it is full rank, and thus $\text{rank}(\bar{\mathbf{e}}(\mathbf{y}) \mathbf{A}) = 2d$ almost surely on \mathcal{Z} . It suffices then to multiply by its pseudo-inverse from the right to get

$$\mathbf{M}(\mathbf{y}) = 0 \quad (89)$$

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 non-zero entry in each row. By invertibility and continuity of $J_{\mathbf{v}}$, we deduce that the location of the non-zero entries are fixed and do not change as a function of \mathbf{y} . We deduce that there exists a permutation σ of $\llbracket 1, d \rrbracket$ 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 $\bar{\mathbf{H}}(\mathbf{z}) = \mathbf{H} \circ \mathbf{v}^{-1}(\mathbf{y}) - \mathbf{c}$. This function is a pointwise function because \mathbf{H} and \mathbf{v}^{-1} are such functions. Plugging this back into equation (84) yields:

$$\bar{\mathbf{H}}(\mathbf{z}) = \mathbf{A} \mathbf{T}(\mathbf{z}) \quad (90)$$

The last equation is valid for every component:

$$\bar{H}_{i,l}(z_i) = \sum_{a,b} A_{i,l,a,b} T_{a,b}(z_a) \quad (91)$$

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

$$0 = \sum_b A_{i,l,s,b} T'_{s,b}(z_s) \quad (92)$$

By Lemma 10, we get $A_{i,l,s,b} = 0$ for all $1 \leq b \leq k$. Since (92) is valid for all l and all $s \neq i$, we deduce that the matrix \mathbf{A} has a block diagonal form:

$$\mathbf{A} = \begin{pmatrix} \mathbf{A}_1 & & \\ & \ddots & \\ & & \mathbf{A}_n \end{pmatrix} \quad (93)$$

which achieves the proof.

One-dimensional sufficient statistics: assumptions (vi')-(viii') We now suppose that $k_i = 1, \forall i$. The proof of Khemakhem et al. (2020, Theorem 3) can be used here, where we define $\mathbf{v} = (\mathbf{f} \circ \mathbf{h})^{-1}$ and $h_{i,a} = D_{i,a}H_a(y_a) - D_{i,a}c_a$, where $\mathbf{D} = \mathbf{A}^{-1}$. We can then rewrite equation (85) for every component as:

$$T_i(v_i(\mathbf{z})) = \sum_{a=1}^d h_{i,a}(z_a) \quad (94)$$

which is the same as equation (45) of Khemakhem et al. (2020). 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. \square

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 (Khemakhem et al., 2020; Hyvärinen and Morioka, 2016). It turns out that we can also incorporate dimensionality reduction in IMCA and its estimation by ICE-BeeM, under some assumptions.

Theorem 6. Assume either of the following hold:

- Assumptions (i)-(x).
- Assumptions (i)- (v), (vi')- (viii'), and (ix)- (x).

Then \mathbf{f}_1 recovers only the modulated latent components as per Theorem 4b.

Proof. The proof of Theorem 4a 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 (78) on these points $\mathbf{y}_0, \dots, \mathbf{y}_k$, the constant term \mathbf{g}_2 and the non-modulated components cancel out, and we are left with the equation

$$\tilde{\mathbf{L}}^T \mathbf{H}_{1:n}(\mathbf{f}_1(\mathbf{x})) = \mathbf{L}^T \mathbf{T}_{1:n}(\mathbf{z}) + \mathbf{b} \quad (95)$$

We then use similar arguments to the proof of Theorem 4a to conclude that

$$\mathbf{H}_{1:n}(\mathbf{f}(\mathbf{x})) = \mathbf{A} \mathbf{T}_{1:n}(\mathbf{z}) + \mathbf{c} \quad (96)$$

where $\mathbf{A} \in \mathbb{R}^n$ a square invertible matrix. At this point, we can make the same conclusion as Theorem 4a, while reducing the dimension of the latent space.

We now explain how we can extend Theorem 4b 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 4b.

One-dimensional sufficient statistics Let $\mathbf{D} = \mathbf{A}^{-1}$ and $h_{i,a} = D_{i,a}H_a(y_a) - D_{i,a}c_a$. We can still write equation (96) like equation (94) as

$$T_i(v_i(\mathbf{z})) = \sum_{a=1}^n h_{i,a}(z_a) \quad (97)$$

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 2b as well, which achieves this part of the proof.

Multi-dimensional sufficient statistics We rewrite equation (96)

$$H_{i,l}(y_i) = \sum_{a=1}^n \sum_{b=1}^{k_i} A_{i,l,a,b} T_{a,b}(v_a(\mathbf{y})) + c_{a,b} \quad (98)$$

for all $i \leq n, l \leq k_i$. We proceed similarly to the proof of Theorem 4b, replacing all mentions of d by n and keeping all differentiations to indices $t, s \leq n$, up to equation (89), 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 (98) and differentiate it with respect to y_s , $s > n$:

$$0 = \sum_{a=1}^d \sum_{b=1}^{k_i} A_{i,l,a,b} T'_{a,b}(v_a(\mathbf{y})) v_a^s(\mathbf{y}) \quad (99)$$

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 proposition 11, 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. \square

E Independently modulated component analysis

As mentioned in section 3, linear latent variable models (Hyvärinen and Oja, 2000) and more recently nonlinear latent variable models may be identifiable provided some additional auxiliary variables (Khemakhem et al., 2020; Hyvärinen et al., 2019). 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. This avenue of research has thus formalized the trade-off between expressivity of the mapping between latents to observations (from linear to nonlinear) and distributional assumptions over latent variables (from independent to conditionally independent given auxiliary variables).

We would like to relax the assumption of independence while maintaining identifiability, resulting in the framework of Independently Modulated Component Analysis (IMCA). In this section of the Appendix, we will give a detailed analysis of the IMCA model independently of any estimation method, drawing parallels to the identifiability results of the nonlinear ICA model presented in Khemakhem et al. (2020).

E.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{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:

$$\begin{aligned} \mathbf{z} &\sim p(\mathbf{z}|\mathbf{y}) \\ \mathbf{x} &= \mathbf{h}(\mathbf{z}) \end{aligned} \quad (100)$$

The main modelisation assumption we make is on the latent variable distribution, given by the following definition, where \mathbf{u} is a dummy variable.

Definition 5 (Exponentially factorial distributions). *We say that a multivariate exponential family distribution is **exponentially factorial** if its density $p(\mathbf{u})$ has the form*

$$p(\mathbf{y}) = \mu(\mathbf{y}) \prod_i e^{\mathbf{T}_i(y_i)^T \boldsymbol{\lambda}_i - \Gamma(\boldsymbol{\lambda})}$$

We assume that the latent variable in the IMCA model has a conditional exponentially factorial distribution, where the parameters of the exponential family are a function of the auxiliary variable \mathbf{y} :

$$p(\mathbf{z}|\mathbf{y}) = \mu(\mathbf{z}) e^{\sum_i \mathbf{T}_i(z_i)^T \boldsymbol{\lambda}_i(\mathbf{y}) - \Gamma(\mathbf{y})} \quad (101)$$

Equations (100) and (101) together define the nonparametric IMCA model with parameters $(\mathbf{h}, \mathbf{T}, \boldsymbol{\lambda}, \mu)$. Most importantly, we allow for an arbitrary base measure $\mu(\mathbf{z})$, *i.e.* the components of the latent variable must no longer be independent, as μ doesn't necessarily factorize across dimensions. The crucial assumption is that the components of the latent variables are independently modulated given the auxiliary variable \mathbf{y} , and that through the term $\exp(\sum_i \mathbf{T}_i(z_i)^T \boldsymbol{\lambda}_i(\mathbf{y}))$.

E.2 Identifiability

The concept of identifiability is core to this work. As such, it is important to understand the different views one can have of this concept.

According to the conventional definition, a probabilistic model $\mathcal{P} = \{\mathcal{P}_\theta : \theta \in \Theta\}$ is identifiable *iff* the mapping $\theta \mapsto \mathcal{P}_\theta$ is bijective, *i.e.* $\mathcal{P}_{\theta_1} = \mathcal{P}_{\theta_2} \implies \theta_1 = \theta_2$. However, this definition 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. Identifiability in this context implies that the equivalence class of the ground truth parameter can be uniquely recovered from observations. This is relevant only if the definition of the equivalence class is sufficiently narrow and specific to be able to make meaningful conclusions. One example of such equivalence relations can be found in linear ICA: 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 is a good example of an equivalence class that doesn't restrict the practical utility of the ICA model.

An example of indeterminacy which is relevant to us here can be found in variational inference of latent variable models: two parameters are equivalent if they map to the same *inference* distribution (Khemakhem et al., 2020). This is the definition we will be using in this work. We will say that a generative model is identifiable if we can uniquely recover the latent variables, as given by the following definition.

Definition 6. Consider two different sets of parameters $(\mathbf{h}, \mathbf{T}, \boldsymbol{\lambda}, \mu)$ and $(\tilde{\mathbf{h}}, \tilde{\mathbf{T}}, \tilde{\boldsymbol{\lambda}}, \tilde{\mu})$, defining two densities p and p' . We say that the IMCA model is strongly identifiable if

$$p(\mathbf{x}|\mathbf{y}) = \tilde{p}(\mathbf{x}|\mathbf{y}) \implies \forall i, \mathbf{T}_i(z_i) = \mathbf{A}_i \tilde{\mathbf{T}}_{\gamma(i)}(\tilde{z}_{\gamma(i)}) + \mathbf{b}_i \quad (102)$$

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

$$p(\mathbf{x}|\mathbf{y}) = \tilde{p}(\mathbf{x}|\mathbf{y}) \implies \mathbf{T}(\mathbf{z}) = \mathbf{A} \tilde{\mathbf{T}}(\tilde{\mathbf{z}}) + \mathbf{b} \quad (103)$$

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

E.3 Theoretical analysis

In this section, we develop the theory of IMCA. We will give sufficient conditions that guarantee a strong identifiability of the latent components, and discuss a degenerate case where we only obtain a weaker form of identifiability.

E.3.1 Definitions

We will first define some sets of distributions which are subsets of the exponential family distribution. We will use u as a dummy variable, and introduce the definitions for the unconditional case. Note that all these definitions apply to the conditional case, when the parameters of the exponential family are a function of an auxiliary variable \mathbf{y} . For completeness, we restate here Definition 4.

Definition 7 (Strongly exponential distributions). We say that a univariate exponential family distribution with density $p(u) = \mu(u)e^{\mathbf{T}(u)^T \boldsymbol{\theta} - \Gamma(\boldsymbol{\theta})}$ is **strongly exponential** if for any subset \mathcal{U} of \mathbb{R} the following is true:

$$(\exists \boldsymbol{\theta} \in \mathbb{R}^k \mid \forall u \in \mathcal{U}, \langle \mathbf{T}(u), \boldsymbol{\theta} \rangle = \text{const}) \implies (\Lambda(\mathcal{U}) = 0 \text{ or } \boldsymbol{\theta} = 0) \quad (104)$$

where Λ is the Lebesgue measure.

We say that a multivariate distribution is strongly exponential if all its univariate marginals are.

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. This definition is very general, and is satisfied by all the usual exponential family distributions like the Gaussian, Laplace, Pareto, Chi-squared, Gamma, Beta, *etc.* We will only prove identifiability

results for strongly exponential families. The non-strongly exponential case will be explored in future work.

There is a certain class of exponential families for which we can only prove a weak form of identifiability. Loosely speaking, this is because this class doesn't constrain the latent space enough.

Definition 8 (Quasi-location exponential distributions). *We say that a univariate exponential family distribution with density $p(u) = \mu(u)e^{\mathbf{T}(u)^T \theta - \Gamma(\theta)}$ 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 a multivariate distribution is quasi-location exponential if all its univariate marginals are.

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 $T(u) = u$ which is a monotonic scalar function, while in the second case it is $T(u) = u^2$, a non-monotonic scalar function.

E.3.2 Identifiability of the general case

As mentioned in section 3, the IMCA model described by equations (100) and (101) generalizes 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); Khemakhem et al. (2020) 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 7. *Assume the following:*

- (I) *The observed data follows the exponential IMCA model of equations (100)-(101).*
- (II) *The mixing function $\mathbf{h} : \mathbb{R}^d \rightarrow \mathbb{R}^d$ is invertible.*
- (III) *The conditional latent distribution $p(\mathbf{z}|\mathbf{y})$ is strongly exponential (definition 7), and its sufficient statistic is differentiable.*
- (IV) *There exist $k + 1$ distinct points $\mathbf{y}^0, \dots, \mathbf{y}^k$ such that the matrix*

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

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

Then the IMCA model is weakly identifiable.

This theorem extends the basic identifiability result of Khemakhem et al. (2020, Theorem 1). It is fundamental as 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 7, we still require that the latent variables are only dependent through the base measure, while still being independently modulated through the auxiliary variable \mathbf{y} . This (and the necessity of having an auxiliary variable) is the price to pay for obtaining identifiability in a nonlinear setting.

E.3.3 Identifiability of the non quasi-location family

The identifiability result of Theorem 7 is weak because of the presence of the linear transformation \mathbf{A} in equation (103). It turns out that by excluding the quasi-location family (definition 8), we can remove this matrix and achieve a stronger form of identifiability. The main technical result of this paper is the following.

Theorem 8. *Assume that the assumptions of Theorem 7 hold. Further assume one of the two following sets of assumptions:*

(V) The sufficient statistic in (101) is twice differentiable and $\dim(\mathbf{T}_l) \geq 2, \forall l$.

(VI) The mixing function \mathbf{h} is a \mathcal{D}^2 -diffeomorphism⁶.

or

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

(VI)' The mixing function \mathbf{h} is a \mathcal{C}^1 -diffeomorphism⁷.

Then the IMCA model is strongly identifiable.

This form of identifiability mirrors the strongest results proven in the nonlinear ICA (Khemakhem et al., 2020, Theorems 2,3), 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 generalizes even existing identifiability theory of the linear case. The mixed case where we have both cases where some sufficient statistics are of dimension greater than 2 and some are univariate and non-monotonic will be studied in future work.

E.4 Estimation of IMCA by self-supervised learning

A recent development in nonlinear ICA is given by Hyvärinen et al. (2019) where the authors assume they observe data $\mathbf{x} = \mathbf{h}(\mathbf{z})$ following a noiseless conditional nonlinear ICA model $p(\mathbf{z}|\mathbf{y}) = \prod_i p_i(z_i|\mathbf{y})$. For estimation, they rely on a self-supervised binary discrimination task based on randomization to learn the unmixing function. More specifically, from a dataset of observations and auxiliary variables pairs $\mathcal{D} = \{\mathbf{x}^{(i)}, \mathbf{y}^{(i)}\}$, they construct a randomized dataset $\mathcal{D}^* = \{\mathbf{x}^{(i)}, \mathbf{y}^*\}$ where \mathbf{y}^* is randomly drawn from the observed distribution of \mathbf{y} . To distinguish between both datasets, a deep logistic regression is used. The last hidden layer of the neural network is a feature extractor whose purpose is to extract the relevant features which will allow to distinguish between the two datasets. Surprisingly, this estimation technique works for IMCA, and is summarized by the following theorem.

Theorem 9. *Self-supervised nonlinear ICA estimation algorithms presented in Hyvärinen and Morioka (2016); Hyvärinen et al. (2019) work for the estimation of IMCA.*

E.5 Proofs

E.5.1 Proof of Theorem 7

Consider two different sets of parameters $(\mathbf{h}, \mathbf{T}, \boldsymbol{\lambda}, \mu)$ and $(\tilde{\mathbf{h}}, \tilde{\mathbf{T}}, \tilde{\boldsymbol{\lambda}}, \tilde{\mu})$, defining two conditional latent densities $p(\mathbf{z}|\mathbf{y})$ and $\tilde{p}(\mathbf{z}|\mathbf{y})$. Suppose that the density of the observations arising from these two different models are equal:

$$p(\mathbf{x}|\mathbf{y}) = \tilde{p}(\mathbf{x}|\mathbf{y}) \quad (105)$$

$$\log p(\mathbf{g}(\mathbf{x})|\mathbf{y}) - \log |\det \mathbf{J}_{\mathbf{h}}^{-1}(\mathbf{x})| = \log p(\tilde{\mathbf{g}}(\mathbf{x})|\mathbf{y}) - \log |\det \mathbf{J}_{\tilde{\mathbf{g}}}(\mathbf{x})| \quad (106)$$

$$\begin{aligned} \log \mu(\mathbf{g}(\mathbf{x})) + \mathbf{T}(\mathbf{g}(\mathbf{z}))^T \boldsymbol{\lambda}(\mathbf{y}) - \Gamma(\mathbf{y}) - \log |\det \mathbf{J}_{\mathbf{g}}(\mathbf{x})| = \\ \log \tilde{\mu}(\tilde{\mathbf{g}}(\mathbf{x})) + \tilde{\mathbf{T}}(\tilde{\mathbf{g}}(\mathbf{z}))^T \tilde{\boldsymbol{\lambda}}(\mathbf{y}) - \tilde{\Gamma}(\mathbf{y}) - \log |\det \mathbf{J}_{\tilde{\mathbf{g}}}(\mathbf{x})| \end{aligned} \quad (107)$$

Let $\mathbf{y}_0, \dots, \mathbf{y}_k$ be the points provided by assumption (IV) 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{y}_l in (107) to obtain $k + 1$ such equations. Then, we subtract the first equation for \mathbf{y}_0 from the remaining k equations to get for $l = 1, \dots, k$:

$$\mathbf{T}(\mathbf{z})^T (\boldsymbol{\lambda}(\mathbf{y}_l) - \boldsymbol{\lambda}(\mathbf{y}_0)) - G(\mathbf{y}_l) = \tilde{\mathbf{T}}(\mathbf{z})^T (\tilde{\boldsymbol{\lambda}}(\mathbf{y}_l) - \tilde{\boldsymbol{\lambda}}(\mathbf{y}_0)) - \tilde{G}(\mathbf{y}_l) \quad (108)$$

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

Most importantly, both base measure terms disappear after taking the differences, which is the key enabler of identifiability in the IMCA framework.

⁶invertible, all second order cross-derivatives of the function and its inverse exist but aren't necessarily continuous

⁷invertible, all partial derivatives of the function and its inverse exist and are continuous

The rest of the proof is similar to the proof of [Khemakhem et al. \(2020, Theorem 1\)](#). The only difference is that we don't restrict the sufficient statistics to have equal dimensions, and so we can't use the proof technique from [Khemakhem et al. \(2020, Theorem 1\)](#) without any modification. We present an alternative technique in the proof of Theorem 4, which we refer too for more details. We then conclude that

$$\mathbf{T}(\mathbf{h}^{-1}(\mathbf{x})) = \mathbf{A}\tilde{\mathbf{T}}(\tilde{\mathbf{h}}^{-1}(\mathbf{x})) + \mathbf{b} \quad (109)$$

which implies that the model is weakly identifiable. \square

E.5.2 Proof of Theorem 8

The conclusion of Theorem 7 is the same as the conclusion of [Khemakhem et al. \(2020, Theorem 1\)](#). Since we make the same assumptions as [Khemakhem et al. \(2020, Theorems 2,3\)](#), the proof to Theorem 8 is similar to the proof of these theorems, which we refer too for more details. The IMCA model is strongly identifiable under the assumptions of Theorem 8. \square

E.5.3 Proof of Theorem 9

We will first quickly summarize the method proposed in [Hyvärinen et al. \(2019\)](#), and then show how it works for IMCA.

We consider that we observe data (\mathbf{x}, \mathbf{y}) that follows the exponential IMCA model of equations (4)-(5). Following [Hyvärinen et al. \(2019\)](#) we start by constructing new data from the observations \mathbf{x} and \mathbf{y} to obtain two datasets

$$\tilde{\mathbf{x}} = (\mathbf{x}, \mathbf{y}) \quad (110)$$

$$\tilde{\mathbf{x}}^* = (\mathbf{x}, \mathbf{y}^*) \quad (111)$$

where \mathbf{y}^* is a random value from the distribution of \mathbf{y} and independent of \mathbf{x} . We then proceed by defining a multinomial 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 randomized dataset or not as labels. In particular, we train a deep neural network using multinomial 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 randomized dataset $\tilde{\mathbf{x}}^*$. The final layer of the network is simply linear, and the regression function takes the form

$$r(\mathbf{x}, \mathbf{y}) = \mathbf{s}(\mathbf{x})^T \mathbf{v}(\mathbf{y}) + \mathbf{a}(x) + \mathbf{b}(u) \quad (112)$$

We state now the main result.

Theorem 9 ([Hyvärinen et al. \(2019\)](#), adapted). *Assume that the assumptions of Theorem 7, and the assumptions (V)-(VI) of Theorem 8 hold. Further assume that we train a nonlinear logistic regression with universal approximation capability to discriminate between $\tilde{\mathbf{x}}$ in (110) and $\tilde{\mathbf{x}}^*$ in (111) with the regression function in (112), where the feature extractor has dimension d .*

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

Proof. The proof of this theorem is inspired by [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:

$$\begin{aligned} \sum_{i=1}^d s_i(\mathbf{x}) v_i(\mathbf{y}) + a(x) + b(u) &= \log p_{\tilde{\mathbf{x}}}(\mathbf{x}, \mathbf{y}) - \log p_{\tilde{\mathbf{x}}^*}(\mathbf{x}, \mathbf{y}^*) \\ &= \log p(\mathbf{z}, \mathbf{y}) + \log |\det \mathbf{J}_{\mathbf{h}}^{-1}(\mathbf{x})| - \log p(\mathbf{z}) p(\mathbf{y}) - \log |\det \mathbf{J}_{\mathbf{h}}^{-1}(\mathbf{x})| \\ &= \log p(\mathbf{z}|\mathbf{y}) - \log p(\mathbf{z}) \\ &= \log \mu(\mathbf{z}) - \log Z(\mathbf{y}) + \sum_{i=1}^d \mathbf{T}_i(z_i)^T \boldsymbol{\lambda}_i(\mathbf{y}) - \log p(\mathbf{z}) \end{aligned} \quad (113)$$

where $\mathbf{J}_{\mathbf{h}}^{-1}(\mathbf{x})$ is the Jacobian matrix of \mathbf{h}^{-1} at point \mathbf{x} . Let $\mathbf{y}_0, \dots, \mathbf{y}_k$ be the point provided by assumption (iv). We plug each of those \mathbf{y}_k in (113) to obtain $k + 1$ such equations. We subtract the first equation for \mathbf{y}_0 from the remaining k equations to get for $l = 1, \dots, k$:

$$\sum_{i=1}^d s_i(\mathbf{x})(v_i(\mathbf{y}_l) - v_i(\mathbf{y}_0)) + (\mathbf{b}(\mathbf{y}_l) - \mathbf{b}(\mathbf{y}_0)) - \log \frac{Z(\mathbf{y}_l)}{Z(\mathbf{y}_0)} = \sum_{i=1}^d \mathbf{T}_i(z_i)^T (\boldsymbol{\lambda}_i(\mathbf{y}_l) - \boldsymbol{\lambda}_i(\mathbf{y}_0)) \quad (114)$$

Interestingly, the term $\log \mu(\mathbf{z})$ cancels out. The rest of the proof is similar to Theorems 4a and 4b. The only minor difference is that the matrix \mathbf{A} will not be square, but it is still full rank, and can be used to prove that $\mathbf{s} \circ \mathbf{h}$ is a point-wise nonlinearity. \square