AN ABSTRACT OF THE DISSERTATION OF

Vladyslav Pauk for the degree of Master of Science in Computer Science presented on November 13, 2024.

Title: Post-nonlinear mixture identification via variational auto-encoding

Abstract approved: _____

Xiao Fu

This thesis addresses parameter estimation in unsupervised learning, with a focus on uncovering simplex-constrained latent structures within noisy high-dimensional data under post-nonlinear distortions. Simplex component analysis, which estimates latent variables residing within a probability simplex, is extensively used in fields such as brain signal classification, speech separation, remote sensing, and causal discovery. While traditional linear mixture models are effective for multivariate data analysis, they are inadequate for capturing the nonlinearities often present in real-world systems. Recent advancements have attempted to address these limitations by extending linear mixture models to nonlinear settings, by utilizing autoencoder-based architectures with structured latent spaces. While existing methods effectively remove nonlinearities, they often underperform in noisy environments and fail to uniquely identify latent components without additional observed variables or post-processing. Framed within variational Bayesian inference, probabilistic generative models, on the other hand, naturally accommodate latent structures and noise, and enable unique identification of the latent components from noisy data. Building upon a variational autoencoder framework for linear probabilistic simplex component analysis, this thesis extends this model to accommodate post-nonlinear distortions. The thesis provides rigorous theoretical analysis of the identifiability in the proposed model, and systematic assessment of the theoretical guarantees in numeric simulations.

Post-nonlinear mixture identification via variational auto-encoding

by

Vladyslav Pauk

A DISSERTATION

submitted to

Oregon State University

in partial fulfillment of
the requirements for the
degree of

Master of Science

Presented November 13, 2024
Commencement June 2025

Master of Science dissertation of Vladyslav Pauk presented on November 13, 2024.

APPROVED:

Major Professor, representing Computer Science

Director of the School of Electrical Engineering and Computer Science

---

Dean of the Graduate School

I understand that my dissertation will become part of the permanent collection of Oregon State University libraries. My signature below authorizes release of my dissertation to any reader upon request.

Vladyslav Pauk, Author

TABLE OF CONTENTS

Page

LIST OF FIGURES

Figure Page

LIST OF TABLES

Table Page

LIST OF ALGORITHMS

Chapter 1: Introduction

"I thought of a labyrinth of labyrinths, of one sinuous spreading labyrinth that would encompass the past and the future and in some way involve the stars."

Jorge Luis Borges, The Garden of Forking Paths

Over the past decade, *artificial intelligence* (AI) systems have rapidly evolved and gained widespread adoption across both research and industry, fundamentally reshaping the modern technological and scientific landscape[1]. At the heart of these AI systems is *machine learning* (ML), a computational paradigm that enables systems to learn from examples by identifying underlying patterns within data. Rooted in foundational research on backpropagation [37, 46] and stochastic gradient-based optimization [8], ML enables data-driven modeling in various domains, from natural language processing and computer vision to autonomous systems and signal processing. The proliferation of digital data from sources such as the internet, mobile devices, IoT sensors, and social media has generated vast datasets essential for training complex ML models. High-performance computing hardware, particularly Graphics Processing Units (GPUs) and, more recently, Tensor Processing Units (TPUs), has played a pivotal role in managing massive datasets and complex computations with efficiency. Additionally, the advent of distributed and cloud computing has democratized access to large-scale training resources, enabling researchers and developers to train models that would otherwise be constrained by local hardware limitations.

The technological advances, coupled with development of new algorithmic architectures, such as Convolutional Neural Networks (CNN), Recurrent Neural Networks (RNN), and attention-based Transformers, have unlocked the full potential of *deep learning*. Deep learning leverages *artificial neural networks* (ANNs), complex parametric models with universal approximation capabilities inspired by biological neural systems, to model real-world data with remarkable accuracy. This approach has led to ground-breaking achievements across diverse applications, including healthcare, finance, science, robotics, and entertainment among others.

In a nutshell, training a model entails iteratively adjusting its parameters based on training examples to minimize the discrepancy between the model's output and the ground truth. Also known as *supervised learning*, this paradigm relies on labeled datasets where the input-output relationships are explicitly defined. A key limitation of supervised learning is its dependence on often manually labeled data, which can be both scarce and costly to obtain, hindering its scalability and utility in many applications.

In contrast, *unsupervised learning* focuses on uncovering latent structures and complex patterns within unlabeled input data by relying on intrinsic relationships between the data points. This approach is most notably applied in *density estimation* and *generative modeling*, giving rise to celebrated models like Variational Autoencoders (VAEs) [34], Generative Adversarial

[1]For a comprehensive and recent overview of AI advancements, adoption trends, and their societal impacts, see [42].

Networks [21], and Stable Diffusion [45], where algorithms learn to model the underlying data probability distribution. These models revolutionized the field of generative AI, due to their ability to generate new samples that closely resemble the input data distribution, producing highly realistic synthetic data with remarkable fidelity. Unsupervised learning techniques are particularly effective in representation learning, where they are employed to transform high-dimensional data into more compact and interpretable forms, while capturing its underlying structure in a model-independent way. However, unsupervised learning is inherently challenging. The lack of labeled data means that the evaluation and interpretation of learned representations and the identification of meaningful patterns require more advanced algorithms and techniques. These methods often require intricate design and optimization, making unsupervised learning a sophisticated and computationally intensive process.

Finding meaningful low-dimensional representations for multivariate data is one of central challenges in ML, particularly within the domain of *dimensionality reduction*. Latent Variable Models (LVMs) address this challenge by assuming that the data is generated by an unknown intrinsic process that depends on a small number of hidden (or latent) variables. For example, in many applications, high-dimensional data is generated through a mixing process, where the observations are represented as a linear combination of lower-dimensional latent variables or factors. This class of models is commonly referred to as linear mixture models (LMM). The *unsupervised mixture learning* (UML) encompasses a variety of methods also known in the literature as *blind source separation* and *factor analysis* [24] that aim to recover the latent components from the observed data with the limited prior knowledge of the mixing process or the characteristics of the latent components. UML methods are essential for tasks like audio and speech separation [20], EEG signal denoising [35], image representation learning [2], hyperspectral unmixing [5], and topic mining [19], among others.

An essential aspect of UML is *identifiability*, which pertains to whether the latent components in a model can be uniquely and reliably identified without the need for labeled data or explicit training examples. Identifiability provides a rigorous foundation for meaningful parameter interpretation in LVM. It is well-known, that linear mixture models with Gaussian prior are generally not identifiable because multiple solutions can exist without additional constraints [27]. Incorporation of structural assumptions on the latent variables, such as statistical independence, nonnegativity, boundedness, sparsity, and simplex structures, form the basis for widely adopted models like Independent Component Analysis (ICA) [24], Nonnegative Matrix Factorization [18], Bounded Component Analysis [13], Sparse Component Analysis [54], and Simplex-Structured Matrix Factorization (SSMF) [20, 19]. By leveraging such structural constraints, these UML frameworks can not only render the problem more tractable but also provide meaningful interpretations of the latent components, enhancing their applicability across various domains.

The structural assumptions often arise naturally from the physical or domain-specific characteristics of the underlying problem. For instance, the simplex structure, central to this work, is commonly encountered in compositional data, representing proportions of a whole. Specifically, each data point corresponds to a categorical probability distribution, represented as a vector with non-negative elements that sum to one. This structure is particularly relevant in

applications such as topic mining, hyperspectral unmixing [5], community detection [23], and image representation learning [53], among other.

The LMMs, while effective in many applications, have limitations. In real-world scenarios, the observed data often undergoes unknown nonlinear distortions, which can significantly affect the performance of linear UML methods. This is observed in various applications, including hyperspectral imaging, audio processing, wireless communications, and brain imaging [3, 55]. The nonlinear character of the problem presents significant challenges, particularly when compared to the well-established linear UML setting. As shown in [26, 28], even strong assumptions like statistical independence of the latent components are insufficient to guarantee identifiability in nonlinear mixture models.

The identifiability of latent components in the presence of unknown nonlinear transformations was elusive until recently, when it has gained renewed interest, due to its connection to unsupervised deep learning [26, 28]. In [28], the authors addressed this challenge by approaching it through the lens of nonlinear Independent Component Analysis (nICA), and provided the first nonlinear identifiability guarantees for deep latent-variable models, offering a rigorous framework for recovering independent latent variables under nonlinear mixture model. In many practical scenarios, the assumption of statistical independence between latent components is often overly restrictive, limiting the applicability of ICA models. While some efforts have extended LMMs to handle specific types of nonlinear distortions, such as bi-linear, linear-quadratic, and polynomial transformations [7, 16, 14], addressing general nonlinear mixtures without resorting to statistical independence of the latent factors remains a significant challenge.

The post-nonlinear (PNL) mixture model [50, 6] offers a nonlinear extension to the linear mixture model (LMM) by introducing component-wise nonlinear transformations following the linear mixing process. This model has been effectively applied to complex tasks in signal processing and data analytics, including nonlinear hyperspectral unmixing, image embedding, brain signal classification, speech separation, remote sensing, causal discovery, and nonlinear clustering. A notable advancement in [52] provided rigorous nonlinear identification criteria for the PNL model, demonstrating that unknown post-nonlinear distortions can be removed under specific geometric assumptions on the latent components. These models are based on the premise that the latent components reside in a lower-dimensional subspace, which imposes interdependencies between the nonlinear transformations. The simplex-constrained post-nonlinear mixture (SC-PNM) model, proposed in [52, 40], leverages the assumption that latent components are generated from the probability simplex. Further, [41] demonstrated that under mild conditions, the presence of a nontrivial null space in the underlying mixing system is sufficient to guarantee the identification and removal of unknown nonlinearities.

However, while these methods successfully eliminate nonlinear distortions, they do not fully recover the latent components, requiring additional constraints and linear methods for component extraction. Moreover, since these methods tackle nonlinear distortions in a deterministic manner, they exhibit performance deterioration in the presence of significant noise. Additionally, the use of geometric constraints via constrained optimization introduces computational complexity, as repeated optimization steps are required, affecting scalability of the method.

Probabilistic methods, such as Bayesian inference, are essential for addressing noise in data. In hyperspectral unmixing (HU), Bayesian inference has been applied to manage uncertainty, but the resulting intractable integrals in certain probability density functions often necessitate computationally expensive techniques like Markov chain Monte Carlo (MCMC) sampling [15]. Deep generative models, such as VAE, have gained popularity for their ability to model complex data distributions and generate realistic synthetic data. In these models, data is modeled as a probabilistic function of hidden latent variables, which are sampled from a prior distribution, and the generative process is learned through the gradient-based optimization of the *maximum likelihood* (ML) objective.

In [31], the problem of nonlinear identifiability was tackled by embedding nonlinear ICA into the VAE framework, extending it to handle noisy or incomplete observations within a maximum-likelihood context. This work demonstrated that, for a broad class of deep latent variable models, the true joint distribution of observed and latent variables can be identified, up to simple transformations, enabling robust disentanglement in noisy nonlinear settings. This approach requires a factorized prior distribution over the latent variables, conditioned on an additional observed variable (e.g., class label), and is akin to nonlinear multiview analysis [39], where multiple data views It also connects to identifiable flow-based generative models, which emerge as a special case. This approach relies on a prior factorized over the latent variables, and conditioned on an auxiliary observed variable (e.g., class label), along the lines of nonlinear multiview analysis [39], and contrast learning [31]. Moreover, it connects to identifiable flow-based generative models, which emerge as a special case of this framework.

One advantage of probabilistic models is their ability to naturally incorporate the geometric constraints into the model architecture, by choosing an appropriate prior distribution. Building on this idea, PRISM employs ML inference for noisy linear mixture model with the simplex-distributed prior, and provides linear identifiability guarantees up to a permutation ambiguity. Probabilistic Simplex Component Analysis (PRISM) addresses the intractability of the likelihood function through approximation methods such as importance sampling and variational inference approximation (VIA), which can be computationally expensive for large-scale problems. Expanding on PRISM, VAE-based Simplex Component Analysis (VASCA) addresses optimization of ML inference through the use of a VAE, modeling the variational posterior with ANNs. This enhances the model's expressive power, bypasses computational bottlenecks, and retains the probabilistic structure and identifiability guarantees of the original framework.

This dissertation aims to extend existing research by addressing the limitations of deterministic nonlinear unsupervised mixture learning models through a reformulation within a probabilistic framework, to study nonlinear identifiability in deep generative models on the example of nonlinear simplex component analysis (nSCA). Unlike deterministic approaches, the probabilistic formulation captures both global and local geometric structures of the data, enabling simultaneous removal of nonlinearity and latent component disentanglement. This eliminates the need for pipelining with linear methods or imposing additional structural constraints. The proposed framework offers several practical advantages, including robustness to noise and generative capabilities, making it applicable to a variety of tasks. Specific applications include

portfolio analysis, ECG signal processing, and hyperspectral unmixing, where accurate separation of latent components from noisy, mixed data is crucial for improved decision-making and analysis. By adopting a probabilistic approach, the model is not only suitable for mapping complex data into interpretable forms, but also for scientific and financial simulations, where it can generate realistic nonlinear synthetic data.

As a first step in this agenda, this thesis focuses on the nonlinear simplex component analysis (nSCA) model, where latent components are generated from the probability simplex. Building on the linear VASCA framework, this approach modifies the architecture to handle nonlinear distortions and address optimization challenges. We introduce a nonlinear decoder and redesign the encoder to more effectively capture the relationships between the linearly mixed components. The central contribution of this work is analysis of the nonlinear identifiability of the probabilistic noisy PNL model in nSCA setting, and the development of a scalable latent identification inference algorithm with provable identifiability guarantees. We provide a detailed description of the theoretical foundation and implementation of the algorithm, formulate and prove identifiability of the model, and evaluate it in experiments on synthetic data.

The following manuscript is organized as follows. In Chapter 2, we provide a formal introduction to the theoretical foundations of probability density estimation, variational inference, variational autoencoders, and the concept of identifiability. Chapter 4 presents the core model of this work, alongside the primary theoretical contributions. In Chapter 5, we conduct experiments on synthetic datasets, where we evaluate and analyze the model's performance under various settings. Following the experiments, we discuss the results, highlighting key insights, challenges, and potential avenues for future research. Finally, the dissertation concludes with a summary of the contributions and broader implications of the work. Notations, supplementary reference materials, and technical derivations are provided in the Appendix.

Chapter 2: Theoretical Background

2.1 Notations and Definitions

Our notation follows standard conventions. Column vectors are represented by bold lowercase letters (e.g., $\boldsymbol{x}$), and matrices by capital letters (e.g., A). By default, $\boldsymbol{a}_n$ denotes the n-th column of A. Notations $A^\top$, A^{-1}, and A^+ refer to the transpose, inverse, and pseudo-inverse of A, respectively. The Euclidean norm is denoted by $\|\cdot\|$, and $\mathrm{tr}(\cdot)$ stands for the trace of a matrix. The diagonal matrix with elements x_n is denoted as $\mathrm{diag}(\boldsymbol{x})$. Vectors $\mathbf{0}$ and $\mathbf{1}$ represent an all-zero vector and an all-one vector, respectively, and the identity matrix is represented by $\boldsymbol{I}$.

The symbols $\mathbb{R}$, $\mathbb{R}_+$, and $\mathbb{R}_{++}$ represent the sets of all real numbers, non-negative real numbers, and positive real numbers, respectively. We denote $\boldsymbol{e}_n,\ n = 1, \ldots, N$ as the standard basis vectors in $\mathbb{R}^N$. For $A \in \mathbb{R}^{N\times M}$ and $\boldsymbol{x} \in \mathbb{R}^N$, we define $\overline{A} = [\boldsymbol{a}_1 - \boldsymbol{a}_M, \ldots, \boldsymbol{a}_{M-1} - \boldsymbol{a}_M]$ and $\overline{\boldsymbol{z}} = (z_1, \ldots, z_{M-1})$. A matrix $A \in \mathbb{R}^{N\times M}$ is said to be *affinely independent* if $\overline{A}$ has full column rank.

For any subset $\mathcal{X} \subseteq \mathbb{R}^N$ and any $\boldsymbol{y} \in \mathbb{R}^N$, $\mathcal{X} + \boldsymbol{y}$ is defined as $\{\boldsymbol{x} + \boldsymbol{y} | \boldsymbol{x} \in \mathcal{X}\}$. For a matrix $A \in \mathbb{R}^{N\times M}$, $\mathrm{span}(A)$ represents the span of $\{\boldsymbol{a}_1, \ldots, \boldsymbol{a}_M\}$, and $\mathrm{aff}(A)$ represents the affine hull defined by $\{\boldsymbol{y} = A\boldsymbol{x} | \boldsymbol{x} \in \mathbb{R}^M, \mathbf{1}^\top \boldsymbol{x} = 1\}$. The convex hull is denoted as $\mathrm{conv}(A) = \{\boldsymbol{y} = A\boldsymbol{x} | \boldsymbol{x} \in \mathbb{R}^M_{++}, \mathbf{1}^\top \boldsymbol{x} = 1\}$, indicating the open convex hull on $\mathrm{aff}(A)$. The indicator function for a set X is defined as

$$\mathbf{1}_X(\boldsymbol{x}) = \begin{cases} 1, & \text{if } \boldsymbol{x} \in \mathcal{X} \\ 0, & \text{if } \boldsymbol{x} \notin \mathcal{X} \end{cases}.$$

A random variable $\boldsymbol{x}$ with distribution $\mathcal{D}$ is denoted by $\boldsymbol{x} \sim \mathcal{D}$, and the expectation of $\boldsymbol{x}$ with probability density $p(\boldsymbol{x})$ by $\mathbb{E}_{\boldsymbol{x}\sim p}[\cdot]$. The variance and covariance of $\boldsymbol{x}$ are denoted by $\mathrm{var}(\boldsymbol{x})$ and $\mathrm{cov}(\boldsymbol{x})$, respectively.

We assume all functions under consideration are complex-valued and measurable. Additionally, C^p denotes the space of all continuous p-times differentiable functions. We also use softmax function, that is defined as

$$\mathrm{softmax}(\boldsymbol{z})_i = \frac{e^{z_i}}{\sum_{j=1}^M e^{z_j}}.$$

2.2 Maximum Likelihood Estimation of Probability Density

2.2.1 Probability Density Estimation

The primary goal of probability density estimation (PDE) is to estimate the underlying probability distribution of a random variable from a finite sample of observed data. This task is essential to many areas of statistics and machine learning, where it is used to analyze data distributions, perform anomaly detection, and build generative models.

PDE methods are broadly classified into non-parametric and parametric approaches. Non-parametric methods, such as kernel density estimation, do not assume a specific parametric form for the distribution. Instead, they rely on the data itself to infer the distribution shape. While this flexibility can be advantageous, it comes at the cost of increasing model complexity. As

the dataset grows, the number of parameters required by non-parametric models also grows, often leading to infinite-dimensional models. This increased complexity, along with the curse of dimensionality, makes non-parametric methods computationally expensive and difficult to scale, particularly in high-dimensional spaces.

On the other hand, parametric methods assume that the data follows a specific distribution characterized by a fixed, finite number of parameters, as in Gaussian Mixture Models (GMMs) [1]. In GMMs, the data is modeled as a mixture of several Gaussian distributions, each with its own mean and covariance. The primary task is to estimate both the parameters of the individual Gaussian components and their relative proportions in the mixture, typically using Expectation-Maximization (EM) to maximize the likelihood of the data. Parametric methods are generally more computationally efficient and interpretable, as the number of parameters does not grow with the size of the data. However, due to their inherent assumptions about the distribution structure, they lack the flexibility to capture realistic data distributions.

Recent advances in deep learning, have expanded the scope of parametric models. Techniques such as VAE and GAN have introduced more flexible, neural network-based approaches to modeling distributions. In these models, the parameters of the distribution are learned through neural networks, allowing them to capture intricate nonlinear data patterns. This hybrid approach combines the computational advantages of parametric methods with the flexibility typically associated with non-parametric approaches, making them powerful tools for density estimation in complex, high-dimensional datasets.

2.2.2 Maximum Likelihood Estimation

The *statistical distance* is a key concept in PDE, and is used to quantify the difference between two probability distributions. The commonly used Kullback-Leibler (KL) divergence originated in information theory, as a measure of relative entropy. Given two distributions $q(\boldsymbol{x})$ and $p(\boldsymbol{x})$, the KL divergence is defined as:

$$D_{\mathrm{KL}}(q \,\|\, p) = \mathbb{E}_{\boldsymbol{x}\sim q}\left[\log \frac{q(\boldsymbol{x})}{p(\boldsymbol{x})}\right]. \tag{2.1}$$

which measures the expected logarithmic difference between the distributions. From the perspective of information theory, the KL divergence can be interpreted as the additional bits required to encode samples from $q(\boldsymbol{x})$ using a coding scheme optimized for $p(\boldsymbol{x})$. In this sense, it is closely related to the concepts of entropy and cross-entropy:

$$D_{\mathrm{KL}}(q \,\|\, p) = H[\,q, p\,] - H[\,q\,],$$

where $H[\,q\,] = -\mathbb{E}_{\boldsymbol{x}\sim q}[\log q(\boldsymbol{x})]$ is the entropy of q representing the inherent uncertainty in a distribution, and $H[\,q, p\,] = -\mathbb{E}_{\boldsymbol{x}\sim q}[\log p(\boldsymbol{x})]$ is the cross entropy of p relative to q quantifying the uncertainty in approximating one distribution with another.

In the context of the density estimation, minimizing the statistical distance between the

true distribution $p^*(\boldsymbol{x})$ and its parametric approximation $p_{\boldsymbol{\theta}}(\boldsymbol{x})$ provides an estimator of the underlying data distribution $p^*(\boldsymbol{x})$. The KL-divergence $D_{KL}(p^*\|p_{\boldsymbol{\theta}})$ then quantifies the expected information loss incurred by using the approximate distribution rather than the true distribution. This is often framed as the maximum likelihood (ML) estimation, where the parameters $\boldsymbol{\theta}$ are optimized to maximize the likelihood $\mathcal{L}_T(\boldsymbol{\theta})$ of observing the given data under the approximate model:

$$\boldsymbol{\theta}_{\mathrm{ML}} = \arg\max_{\boldsymbol{\theta}} \mathcal{L}_T(\boldsymbol{\theta}). \tag{2.2}$$

Given a dataset of T independent and identically distributed samples $\{\boldsymbol{x}^{(t)} \sim p^*(\boldsymbol{x})\}$, the empirical log-likelihood function is:

$$\mathcal{L}_T(\boldsymbol{\theta}) = \sum_{t=1}^{T} \log p_{\boldsymbol{\theta}}(\boldsymbol{x}^{(t)}), \tag{2.3}$$

which is related to the cross-entropy between the true distribution and the model distribution $H[\,p^*, p_{\boldsymbol{\theta}}\,] = -\lim_{T\to\infty} \mathcal{L}_T(\boldsymbol{\theta})$ in the population limit. Maximizing (2.3), thus minimizes the divergence betweens the approximate distribution $p_{\boldsymbol{\theta}}(\boldsymbol{x})$ with the true distribution $p^*(\boldsymbol{x})$, effectively aligning them.

ML offers a practical and computationally feasible method to estimate the parameters $\boldsymbol{\theta}$ that best approximate the underlying true distribution $p^*(\boldsymbol{x})$ often used in parametric density estimation. It provides a consistent and asymptotically unbiased estimate of the data distribution under general conditions. By minimizing the KL divergence between $p^*(\boldsymbol{x})$ and $p_\theta(\boldsymbol{x})$, ML identifies the closest approximation, even when the true distribution is not within the model family. This ensures that, in practical applications, the estimated parameters offer the best possible fit within the constraints of the model, even if the model is imperfect and the data is finite. ML estimation serves as the foundation for VIA approaches, where approximating complex posterior distributions is necessary, such as in the evidence lower bound (ELBO) optimization used in VAE, central to this work.

2.3 Variational Inference in Latent Variable Models

2.3.1 Latent Variable Models

Latent Variable Models (LVMs) explain correlations between observed variables by assuming a common underlying generative process, governed by a set of hidden factors. These hidden factors, or latent variables, typically reside in a space with much lower dimensionality than the observed data, meaning that the data effectively lies on a low-dimensional manifold within the high-dimensional observation space. This property is often leveraged in dimensionality reduction techniques, where such manifolds reveal the intrinsic structure of the data. In practice, real-world data points are not perfectly confined to this smooth, low-dimensional manifold, as noise introduces deviations. These departures from the ideal manifold can be interpreted as perturbations or noise around the true generative structure. Formally, we represent observed

data $\boldsymbol{x}$ as being generated from latent variables $\boldsymbol{z}$ through a deterministic mapping function $\boldsymbol{f}$ and an added noise component $\boldsymbol{v}$:

$$\boldsymbol{x} = \boldsymbol{f}(\boldsymbol{z}) + \boldsymbol{v}.$$

In the simplest case, where both the latent variables and noise are Gaussian distributed and the mapping is linear, this generative model aligns with principal component analysis (PCA) and related approaches like factor analysis [9].

The hidden variables in a probabilistic model need not, however, have any explicit physical interpretation but may be introduced simply to allow a more complex joint distribution to be constructed from simpler components. If we define a joint distribution over observed and latent variables, the corresponding distribution of the observed variables alone is obtained by marginalization. This allows relatively complex marginal distributions over observed variables to be expressed in terms of more tractable joint distributions over the expanded space of observed and latent variables. Specifically, they all employ latent variable models, and each postulates a different latent prior—independent Gaussian for probabilistic PCA, independent non-Gaussian for probabilistic ICA, simplex-uniform for PRISM, and Dirichlet mixture for DECA.

This leads naturally to a generative view of such models in which we first select a point within the manifold according to some latent variable distribution and then generate an observed data point by adding noise, drawn from some conditional distribution of the data variables given the latent variables:

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

where $p_{\boldsymbol{\theta}}(\boldsymbol{x}|\boldsymbol{z})$ is the likelihood of the observed data given the latent variables, and $p(\boldsymbol{z})$ is the prior distribution over $\boldsymbol{z}$. The Bayesian framework provides a powerful structure for learning

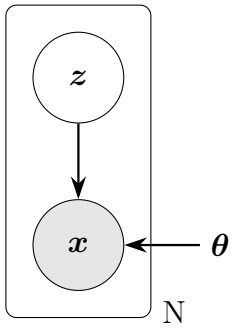

Figure 2.1: Ditected graphical model representation of a Latent Variable Model (LVM). The observed data is represented by the shaded node $\boldsymbol{x}$, and the latent variables by the unshaded node $\boldsymbol{z}$. Solid lines denote the generative model $p(z)p_\theta(x|z)$, with the generative model parameters θ.

LVMs from data, formalizing the process of updating our beliefs about the latent variables based on observed data. Within the Bayesian framework, learning about the latent variable $\boldsymbol{z}$ involves inferring the posterior distribution $p_{\boldsymbol{\theta}}(\boldsymbol{z}|\boldsymbol{x})$, which represents the probability of latent states given observed data. This is formalized by Bayes' theorem, which expresses the posterior as

$$p_{\boldsymbol{\theta}}(\boldsymbol{z}|\boldsymbol{x}) = \frac{p_{\boldsymbol{\theta}}(\boldsymbol{x}|\boldsymbol{z})p(\boldsymbol{z})}{p_{\boldsymbol{\theta}}(\boldsymbol{x})}, \tag{2.4}$$

where $p_{\boldsymbol{\theta}}(\boldsymbol{x})$ is the marginal likelihood or evidence of the data under the model. The corresponding distribution over the visible parameters $p_{\boldsymbol{\theta}}(\boldsymbol{x})$ can be obtained by marginalizing (**??**) over the latent variables:

$$p_{\boldsymbol{\theta}}(\boldsymbol{x}) = \int p_{\boldsymbol{\theta}}(\boldsymbol{x}|\boldsymbol{z})p(\boldsymbol{z})d\boldsymbol{z}. \tag{2.5}$$

Except for very specific forms of the $p_{\boldsymbol{\theta}}(\boldsymbol{x}, \boldsymbol{z})$ and $p(\boldsymbol{z})$, this marginalization is in general not analytically tractable, as it may involve, for example, an exponential in the dimensionality of the latent space number of terms [9].

2.3.2 Evidence-Lower Bound

Variational inference [29], widely used in modern machine learning, builds on foundational principles of variational calculus, which has its roots in the 18th-century work of Euler and Lagrange. This calculus provides a method to find extrema of functionals, i.e., mappings from a function space to real numbers, by examining how the functional responds to small variations in the input functions. Such functionals often take the form of integrals over functions and their derivatives; examples include expectations, like entropy and KL divergence. Many complex problems involving optimizing over function spaces can be framed as finding extremum of a functional. This approach is ubiquitous in scientific applications; some examples include finite element methods for differential equations [30], the maximum entropy principle in statistical mechanics [44], and path integrals in quantum mechanics [17].

Variational inference leverages these principles to approximate intractable posterior distributions, method also known in the literature as variational inference approximation (VIA) or amortized inference. The core idea is to select a approximation of the posterior from a tractable parametric family and adjust it to be as close as possible to the true posterior. This process turns inference into an optimization task of minimizing the KL divergence between the approximate and the target distribution. By introducing a surrogate posterior $q_{\boldsymbol{\phi}}(\boldsymbol{z})$ parametrized by $\boldsymbol{\phi}(\boldsymbol{x})$, we find a tractable lower bound for $\log p_{\boldsymbol{\theta}}(\boldsymbol{x})$, and maximize this bound to approximate the maximum likelihood. To emphasize this connection we can obtain the ELBO by rewriting the log-likelihood using the Jensen inequality. Given an arbitrary measurable probability density function over the latent variables $q_\phi(\boldsymbol{z})$, we utilize the Jensen inequality to obtain:

$$\log \mathbb{E}_{\boldsymbol{z}\sim q_\phi(\boldsymbol{z})}\left[\frac{p_\theta(\boldsymbol{x}, \boldsymbol{z})}{q_\phi(\boldsymbol{z})}\right] \geq \mathbb{E}_{\boldsymbol{z}\sim q_\phi(\boldsymbol{z})}\left[\log \frac{p_\theta(\boldsymbol{x}, \boldsymbol{z})}{q_\phi(\boldsymbol{z})}\right]. \tag{2.6}$$

The quantity on the right side of (2.6) is known as the Evidence Lower Bound (ELBO):

$$\ell_{\boldsymbol{\theta},\boldsymbol{\phi}}(\boldsymbol{x}) = \mathbb{E}_{\boldsymbol{z}\sim q_{\boldsymbol{\phi}}(\boldsymbol{z})}\left[\log \frac{p_\theta(\boldsymbol{x}, \boldsymbol{z})}{q_{\boldsymbol{\phi}}(\boldsymbol{z})}\right]. \tag{2.7}$$

The difference between the evidence and the ELBO is the KL divergence between the variational and the true posteriors:

$$\log p_\theta(\boldsymbol{x}) = \ell_{\boldsymbol{\theta},\boldsymbol{\phi}}(\boldsymbol{x}) + D_{KL}(q_{\boldsymbol{\phi}}(\cdot) \,\|\, p_\theta(\cdot|\boldsymbol{x})). \tag{2.8}$$

Hence, ELBO approximates the evidence when $q_{\boldsymbol{\phi}}(\boldsymbol{z}) = p_\theta(\boldsymbol{z}|\boldsymbol{x})$. Using ELBO, we can reframe the ML maximization problem in (2.3) as

$$\theta_{ML}, \phi_{ML} = \max_{\boldsymbol{\theta},\boldsymbol{\phi}} \frac{1}{T} \sum_{t=1}^{T} \ell_{\boldsymbol{\theta},\boldsymbol{\phi}}(\boldsymbol{x}^{(t)}). \tag{2.9}$$

Due to linearity of expectations ELBO exhibits different linear decompositions. By explicitly balancing the trade-offs within the loss function, one can adjust the model to emphasize different properties of the learned representation. A commonly used form is expressed as a difference between a reconstruction term and a KL divergence with respect to the prior $p(\boldsymbol{z})$:

$$\ell_{\boldsymbol{\theta},\boldsymbol{\phi}}(\boldsymbol{x}) = \mathbb{E}_{\boldsymbol{z}\sim q_{\boldsymbol{\phi}}(\boldsymbol{z}|\boldsymbol{x})}[\log p_{\boldsymbol{\theta}}(\boldsymbol{x}|\boldsymbol{z})] - D_{KL}[\, q_{\boldsymbol{\phi}} \,\|\, p\,]. \tag{2.10}$$

Maximizing ELBO simultaneously attempts to keep $q_{\boldsymbol{\phi}}(\cdot|\boldsymbol{x})$ close to $p(\boldsymbol{z})$ and concentrate $q_{\boldsymbol{\phi}}(\cdot|\boldsymbol{x})$ on those $\boldsymbol{z}$ that likely generated $\boldsymbol{x}$.

We also introduce the following form of the objective:

$$\ell_{\boldsymbol{\theta},\boldsymbol{\phi}}(\boldsymbol{x}) = \mathbb{E}_{\boldsymbol{z}\sim q_{\boldsymbol{\phi}}(\boldsymbol{z}|\boldsymbol{x})}[\log p_{\boldsymbol{\theta}}(\boldsymbol{x}|\boldsymbol{z}) + \log p(\boldsymbol{z}) + h_\phi(\boldsymbol{z})]. \tag{2.11}$$

where $h_\phi(\boldsymbol{z}) = -\log q_{\boldsymbol{\phi}}(\boldsymbol{z})$ is the point-wise differential entropy of the variational distribution.

Higher entropy encourages $q_{\boldsymbol{\phi}}(\boldsymbol{z})$ to be more spread out, avoiding overfitting by ensuring the posterior doesn't collapse into a very narrow distribution. This term encourages exploration, promoting broader, less certain posterior distributions that can prevent overfitting. Entropy provides a more interpretable measure of uncertainty in the posterior approximation. Higher entropy directly corresponds to more uncertainty in the latent representation, while lower entropy corresponds to a more concentrated and certain posterior. This interpretability can be useful for model diagnostics and understanding how the model represents uncertainty in the data.

2.4 Stochastic Optimization of Variational Autoencoders

2.4.1 Variational Autoencoder

Variational Autoencoders (VAEs) [34] are a class of generative models that combine variational inference with deep learning. VAEs are designed to learn a generative model of data shown in Figure 2.2, by optimizing the lower bound on the log-likelihood of the data, framed as the ELBO loss. The model consists of two neural networks: an encoder network $q_{\boldsymbol{\phi}}(\boldsymbol{z}|\boldsymbol{x})$ that maps data points $\boldsymbol{x}$ to a distribution over latent variables $\boldsymbol{z}$, and a decoder network $p_{\boldsymbol{\theta}}(\boldsymbol{x}|\boldsymbol{z})$ that maps latent variables back to the data space.

Figure 2.3 illustrates the architecture of a VAE, comprising an encoder, a latent sampling step, and a decoder. Starting with the input $\boldsymbol{x}$, the encoder network (shown in blue) processes the data to produce two outputs: μ_t (mean) and $\log \sigma_t$ (log standard deviation). These parameters define the Gaussian distribution from which the latent variable $z^{(r)}$ is sampled using the reparameterization trick: $z^{(r)} = g(\mu + \epsilon^{(r)} \odot \sigma)$, where $\epsilon^{(r)} \sim \mathcal{N}(0, I)$ introduces stochasticity

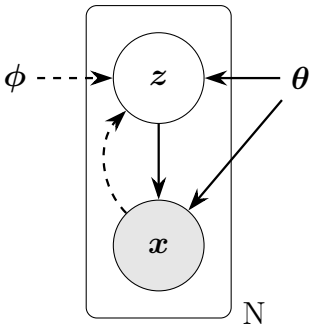

Figure 2.2: Graphical model representation of a Variational Autoencoder (VAE). Solid lines denote the generative model $p_\theta(z)p_\theta(x|z)$, dashed lines denote the inference model, represented by the variational approximation $q_\phi(z|x)$ to the intractable posterior $p_\theta(z|x)$. The variational parameters ϕ are learned jointly with the generative model parameters θ.

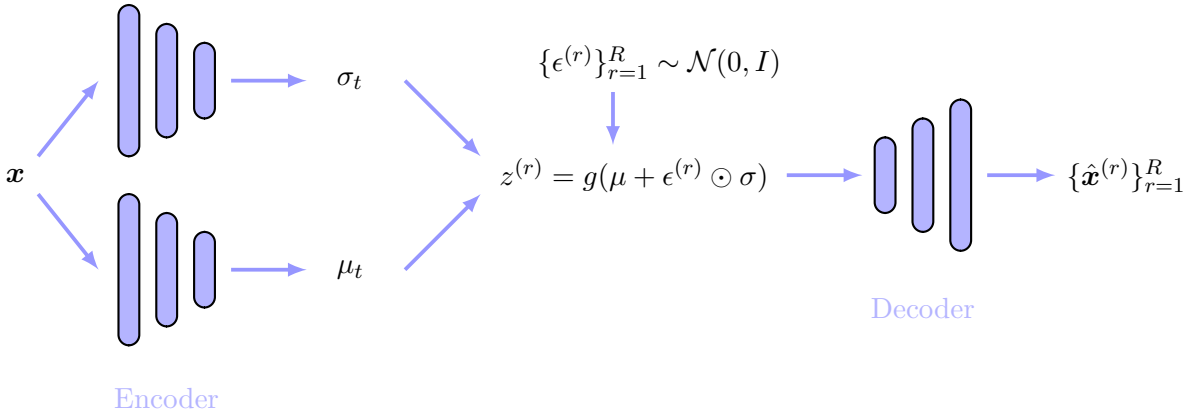

Figure 2.3: Architecture of a Variational Autoencoder (VAE). The input $\boldsymbol{x}$ is processed by an encoder represented by stacked blue layers denoting the neural netwrorks. The encoder outputs the mean μ_t and log standard deviation $\log \sigma_t$, defining a Gaussian distribution. The reparameterization trick $z^{(r)} = \mu + \epsilon^{(r)} \odot \sigma$, where $\epsilon^{(r)} \sim \mathcal{N}(0, I)$, allows for sampling of the latent variable $z^{(r)}$. The sampled latent variable then flows through the decoder layers, reconstructing the input as a sample $\{\hat{\boldsymbol{x}}^{(r)}\}_{r=1}^R$.

into the sampling process. The sampled latent variable $z^{(r)}$ is then passed to the decoder, which reconstructs the original input, generating the output $\{\hat{\boldsymbol{x}}^{(r)}\}_{r=1}^R$ as an approximation of $\boldsymbol{x}$. This structure enables the VAE to learn a compressed representation of the input while capturing probabilistic variations in the latent space.

2.4.2 Stochastic Variational-Bayes Estimator

Optimization with gradient-based methods requires differentiation of expectations over $\boldsymbol{z} \sim q_{\boldsymbol{\phi}}(\boldsymbol{z}|\boldsymbol{x})$ with respect to the variational parameters $\boldsymbol{\phi}$. To address this, we employ the reparameterization trick, which enables the construction of a differentiable estimator. The key idea is to express the latent variable $\boldsymbol{z}$ as a deterministic transformation of a noise variable $\boldsymbol{\epsilon}$, which is independent of the parameters $\boldsymbol{\phi}$. Specifically, we reparameterize $\boldsymbol{z}$ as $\boldsymbol{z} = g_{\boldsymbol{\phi}}(\boldsymbol{\epsilon}, \boldsymbol{x})$, where $\boldsymbol{\epsilon}$ is

drawn from a fixed distribution $p(\boldsymbol{\epsilon})$. This allows us to rewrite the original expectation as:

$$\mathbb{E}_{\boldsymbol{z}\sim q_{\boldsymbol{\phi}}(\boldsymbol{z}|\boldsymbol{x})}[f(\boldsymbol{z})] = \int q_{\boldsymbol{\phi}}(\boldsymbol{z}|\boldsymbol{x}) f(\boldsymbol{z})\, d\boldsymbol{z} = \int p(\boldsymbol{\epsilon}) f(g_{\boldsymbol{\phi}}(\boldsymbol{\epsilon}, \boldsymbol{x}))\, d\boldsymbol{\epsilon}.$$

Since $\boldsymbol{\epsilon}$ is independent of $\boldsymbol{\phi}$, the function $f(g_{\boldsymbol{\phi}}(\boldsymbol{\epsilon}, \boldsymbol{x}))$ is differentiable with respect to $\boldsymbol{\phi}$. We can thus approximate the expectation using Monte Carlo sampling as:

$$\mathbb{E}_{q_{\boldsymbol{\phi}}(\boldsymbol{z}|\boldsymbol{x})}[f(\boldsymbol{z})] \approx \frac{1}{R}\sum_{l=1}^{R} f(g_{\boldsymbol{\phi}}(\boldsymbol{\epsilon}^{(r)}, \boldsymbol{x})) \quad \text{where } \boldsymbol{\epsilon}^{(r)} \sim p(\boldsymbol{\epsilon}),$$

which provides a differentiable estimator that can be optimized using gradient-based methods.

A concrete example of this approach is the univariate Gaussian distribution $\boldsymbol{z} \sim \mathcal{N}(\boldsymbol{\mu}, \Sigma)$. We can reparameterize z as $\boldsymbol{z} = \boldsymbol{\mu} + \Sigma^{1/2}\boldsymbol{\epsilon}$, where $\boldsymbol{\epsilon} \sim \mathcal{N}(0, 1)$. The expectation becomes:

$$\mathbb{E}_{\mathcal{N}(\boldsymbol{z};\boldsymbol{\mu},\Sigma)}[f(\boldsymbol{z})] = \mathbb{E}_{\mathcal{N}(\boldsymbol{\epsilon};0,1)}[f(\boldsymbol{\mu} + \Sigma^{1/2}\boldsymbol{\epsilon})].$$

Using Monte Carlo sampling, we can approximate this expectation as:

$$\mathbb{E}_{\mathcal{N}(\boldsymbol{z};\boldsymbol{\mu},\sigma^2)}[f(\boldsymbol{z})] \approx \frac{1}{R}\sum_{l=1}^{R} f(\boldsymbol{\mu} + \Sigma^{1/2}\boldsymbol{\epsilon}^{(r)}) \quad \text{where } \boldsymbol{\epsilon}^{(r)} \sim \mathcal{N}(0, 1),$$

which is now differentiable with respect to both $\boldsymbol{\mu}$ and Σ.

The reparameterization trick is not limited to Gaussian distributions. It can be applied to any distribution where a suitable transformation of the form $\boldsymbol{z} = g_{\boldsymbol{\phi}}(\boldsymbol{\epsilon}, \boldsymbol{x})$ can be found. For example, for distributions in the location-scale family (such as the Laplace or Student's t-distribution), we can reparameterize using the standard form of the distribution as the noise variable $\boldsymbol{\epsilon}$. Additionally, for some distributions, such as the Log-Normal or Gamma distributions, reparameterization can be achieved through composition of simpler transformations. In cases where reparameterization is not straightforward, approximate methods such as inverse CDF transformations or numerical approximations can be employed to achieve similar results. This flexibility makes the reparameterization trick a powerful tool for constructing differentiable Monte Carlo estimators, enabling the efficient optimization of complex models involving stochastic components.

By applying this technique to the variational lower bound we obtain the Stochastic Gradient Variational Bayes (SGVB) estimator of the evidence lower bound (ELBO) in (2.10). The SGVB estimator is given by:

$$\tilde{\ell}(\boldsymbol{\theta}, \boldsymbol{\phi}; \boldsymbol{x}^{(t)}) = \frac{1}{R}\sum_{r=1}^{R} \log p_{\boldsymbol{\theta}}(\boldsymbol{x}^{(t)}, \boldsymbol{z}^{(t,r)}) - \log q_{\boldsymbol{\phi}}(\boldsymbol{z}^{(t,r)}|\boldsymbol{x}^{(t)}), \tag{2.12}$$

where $\boldsymbol{z}^{(t,r)} = g_{\phi}(\boldsymbol{\epsilon}^{(t,r)}, \boldsymbol{x}^{(t)})$ and $\boldsymbol{\epsilon}^{(t,r)} \sim p(\boldsymbol{\epsilon})$. This formulation facilitates computation of gradients with respect to both the generative model parameters $\boldsymbol{\theta}$ and the variational parameters $\boldsymbol{\phi}$, enabling the use of stochastic gradient descent for optimization.

2.5 Probabilistic Simplex Component Ananlysis

2.5.1 Probability Simplex

The concept of a *simplex* is foundational in convex geometry and serves as the basis for simplex component analysis (SCA) in data science and signal processing. A simplex is defined as the convex hull $\text{conv}(A) = \{\boldsymbol{y} = A\boldsymbol{x} | \boldsymbol{x} \in \mathbb{R}^M_{++}, \mathbf{1}^\top \boldsymbol{x} = 1\}$ of an affinely independent set of points, known as its vertices. Let $A = \{\boldsymbol{a}_1, \ldots, \boldsymbol{a}_M\}$ represent the vertices of a simplex, where affine independence ensures that the modified matrix $\overline{A} = [\boldsymbol{a}_1 - \boldsymbol{a}_M, \ldots, \boldsymbol{a}_{M-1} - \boldsymbol{a}_M]$, obtained by shifting all vertices relative to one of them, has full column rank. A simplex is considered full-dimensional in $\mathbb{R}^N$ if $N = M - 1$, meaning it is spanned by $N + 1$ points in N-dimensional space. The volume of a full-dimensional simplex defined by $A \in \mathbb{R}^{N \times M}$ can be generalized to rectangular matrices with $N > M$:

$$\text{vol}(A) = \frac{\left(\det \overline{A}^\top \overline{A}\right)^{1/2}}{(M-1)!}. \tag{2.13}$$

This definition will be instrumental in integrating over simplexes residing in a lower-dimensional space in further chapters.

A *probability simplex* is a geometric construct representing the space of probability distributions over a finite set of mutually exclusive events, often referred to as categories. Each point in a probability simplex corresponds to a unique distribution across M categories, where M denotes the dimensionality of the simplex. Each coordinate of a point represents the probability of a particular category, such that the probabilities across all categories sum to one, defined over the unit M-simplex Δ^M:

$$\Delta^M = \{\boldsymbol{z} \in \mathbb{R}^M_+ \mid \mathbf{1}^\top \boldsymbol{z} = 1\}. \tag{2.14}$$

The *Dirichlet distribution* is commonly used for modeling random variables on the unit simplex. For a random variable $\boldsymbol{z}$ distributed over a unit simplex Δ^M, the Dirichlet distribution with concentration parameter $\alpha \in \mathbb{R}^N_+$ has the probability density function (PDF):

$$D(\boldsymbol{z}; \alpha) = \frac{1}{B(\alpha)} \prod_{i=1}^{N} z_i^{\alpha_i - 1},$$

where

$$B(\alpha) = \frac{\prod_{i=1}^N \Gamma(\alpha_i)}{\Gamma(\sum_{i=1}^N \alpha_i)}$$

is a normalization constant derived from the Gamma function Γ. The parameter vector α dictates the distribution's shape, making it versatile for modeling diverse data structures. Similarly to how the uniform Gaussian distribution $\mathcal{N}(0, I)$ is used to model the prior in the AEVB variant of VAE, the uniform Dirichlet density:

$$\mathcal{D}(\boldsymbol{z}; \mathbf{1}) = (N-1)! \cdot \mathbb{1}_{\overline{\Delta}}(\boldsymbol{z}) \tag{2.15}$$

is used as a prior in the SCA models. More details on the Dirichlet distribution and its properties are provided in Appendix A.

Multivariate location-scale distributions describe random variables whose transformation is distributed according to a multivariate normal distribution with the diagonal covariance matrix. This can be expressed as the following random variable model:

$$\boldsymbol{z} = g(\boldsymbol{\mu} + \boldsymbol{\sigma} \odot \boldsymbol{\epsilon}), \tag{2.16}$$

where $\boldsymbol{\epsilon} \sim \mathcal{N}(0, \boldsymbol{I})$, and $\boldsymbol{g}$ is a differentiable transformation. The general form of the location-scale family is given by:

$$p_{\boldsymbol{g},\boldsymbol{\sigma},\boldsymbol{\mu}}(\boldsymbol{z}) = \prod_{m=1}^{M} p(z_m) = J_g^{-1} \prod_{m=1}^{M} \frac{1}{\sigma_m} \phi\left(\frac{g^{-1}(z_m) - \mu_m}{\sigma_m}\right), \tag{2.17}$$

where $\phi(\cdot)$ is the standard normal distribution, μ_m and σ_m are the location and scale parameters, respectively, $g(\cdot)$ is the transformation and J_g is the Jacobian of the transformation. The location-scale family provides a flexible and expressive parametric distribution that allows for easy sampling and stochastic optimization.

The logistic-normal (LN) distribution is another example of a probability distribution over the probability simplex obtained by applying the additive logistic transformation $g(\boldsymbol{z}) = \text{softmax}([\boldsymbol{z}; 0])$ to a Gaussian distribution, or equivalently as the probability distribution of a random variable whose multinomial logit is a normal distribution. Explicitly:

$$\begin{aligned} g_i(\boldsymbol{z}) &= \frac{e^{z_i}}{1 + \sum_{j=1}^{M-1} e^{z_j}} \quad \text{for } i = 1, 2, \ldots, M-1, \\ g_M(\boldsymbol{z}) &= \frac{1}{1 + \sum_{j=1}^{M-1} e^{z_j}}. \end{aligned}$$

The probability density function of the LN distribution is:

$$q_{\boldsymbol{\phi}}(\boldsymbol{z}|\boldsymbol{x}) = |2\pi\boldsymbol{\Sigma}|^{-1/2} \left(\prod_{m=1}^{M} z_m\right)^{-1} \exp\left(-\frac{1}{2}\left\{\log\left(\frac{\boldsymbol{z}_{-M}}{z_M}\right) - \boldsymbol{\mu}\right\}^{\top} \boldsymbol{\Sigma}^{-1} \left\{\log\left(\frac{\boldsymbol{z}_{-M}}{z_M}\right) - \boldsymbol{\mu}\right\}\right), \tag{2.18}$$

where $\boldsymbol{z}_{-M} = [z_1, \ldots, z_{M-1}]$ and $z_M = 1 - \sum_{m=1}^{M-1} z_m$, the location parameter is given by the Gaussian mean and the diagonal scale matrix is the covariance matrix. The LN distribution is commonly used as a posterior parameterization in simplex component analysis models, as it provides a flexible and tractable distribution over the probability simplex.

2.5.2 Probabilistic Simplex Component Analysis

The probabilistic simplex component analysis (PRISM) [51] addresses the problem of identifying simplex vertices from a noisy set of data points, where each observation $\boldsymbol{x}^{(t)} \in \mathbb{R}^N$ is modeled

as:

$$\boldsymbol{x}^{(t)} = \boldsymbol{y}^{(t)} + \boldsymbol{v}^{(t)}, \quad \boldsymbol{y}^{(t)} = A^* \boldsymbol{z}^{(t)} \in \operatorname{conv}(A^*),$$

where $\boldsymbol{y}^{(t)}$ is the noise-free component, $A^* \in \mathbb{R}^{N \times M}$ is the true vertex matrix of the simplex, $\boldsymbol{z}^{(t)} \in \Delta$ represents latent variables, and $\boldsymbol{v}^{(t)}$ is Gaussian noise with zero mean and covariance $\sigma^2 I$. It is assumed that each $\boldsymbol{z}^{(t)}$ is drawn independently from a uniform distribution over the unit simplex, represented by a 1-Dirichlet distribution. Furthermore, the matrix A^* has affinely independent columns, ensuring a well-defined simplex structure.

The objective in PRISM is to estimate the vertices A through ML inference:

$$\hat{A} \in \arg\max_{A \in \mathbb{R}^{M \times N}} \mathcal{L}(A) \coloneqq \frac{1}{T} \sum_{t=1}^{T} \log p_A(\boldsymbol{x}^{(t)}), \tag{2.19}$$

where $p_A(\boldsymbol{x})$ is the likelihood of an observation under parameter A. For the given model, this likelihood is expressed as:

$$p_A(\boldsymbol{x}) = (M-1)! \int \phi_\sigma(\boldsymbol{x} - A\boldsymbol{z})\, 1_{\bar{\Delta}}(\boldsymbol{z})\, d\mu(\boldsymbol{z}),$$

where $\phi_\sigma(\boldsymbol{x})$ represents the multivariate Gaussian distribution.

The integral in the likelihood expression lacks a closed-form solution, making direct ML inference challenging. One possible alternative is to maximize the joint likelihood over both A and the latent variables $\boldsymbol{z}_1, \ldots, \boldsymbol{z}_T$. This leads to a *simplex-structured matrix factorization* (SSMF) problem:

$$\min_{A \in \mathbb{R}^{N \times M}, Z \in \Delta^T} \|X - AZ\|^2,$$

where X denotes the matrix of observed data points, and Z is a matrix containing simplex-constrained latent variables. However, SSMF suffers from an identifiability issue: if R is any invertible matrix in Δ^N, then (AR^{-1}, RZ) is also a solution to the problem. This lack of uniqueness implies that even noise-free solutions may not recover the true vertices accurately.

In the noiseless case, the log likelihood simplifies to:

$$\log p_A(\boldsymbol{x}) = -\log \operatorname{vol}(A) + \log(1_{\operatorname{conv}(A)}(\boldsymbol{x})),$$

where

$$\log(1_{\operatorname{conv}(A)}(\boldsymbol{x})) = \begin{cases} 0, & \boldsymbol{x} \in \operatorname{conv}(A) \\ -\infty, & \boldsymbol{x} \notin \operatorname{conv}(A). \end{cases}$$

Thus, the ML problem reduces to:

$$\min_{A \in \mathcal{A}} \log \operatorname{vol}(A) \quad \text{s.t.} \quad \boldsymbol{x}^{(t)} \in \operatorname{conv}(A), \quad t = 1, \ldots, T.$$

This objective seeks the minimum-volume simplex $\operatorname{conv}(A)$ that encloses all data points $\boldsymbol{x}^{(t)}$, aligning PRISM with the simplex volume minimization (SVMin) approach in the noiseless case.

Let q be any probability density function (PDF) measurable on $\{s \in \mathbb{R}^N \mid 1^\top s = 1\}$ with

support $\bar{\Delta}$. Define $p_A(\boldsymbol{x}, \boldsymbol{z}) = p_A(\boldsymbol{x}|\boldsymbol{z})p(\boldsymbol{z})$. The log likelihood of y under A is bounded by the ELBO:

$$\log p_A(\boldsymbol{x}) \geq \mathbb{E}_{\boldsymbol{z}\sim q}\left[\log\left(\frac{p_A(\boldsymbol{x}, \boldsymbol{z})}{q(\boldsymbol{z})}\right)\right] =: \hat{\ell}(A, q; \boldsymbol{x}), \tag{2.20}$$

where equality holds if and only if

$$q(\boldsymbol{z}) \propto p(\boldsymbol{x}, \boldsymbol{z}; A) \propto \phi_\sigma(\boldsymbol{x} - A\boldsymbol{z})\, 1_{\bar{\Delta}}(\boldsymbol{z}),$$

i.e., q is a Gaussian distribution truncated to the unit simplex. This condition is equivalent to:

$$q(\boldsymbol{z}) = \frac{p_A(\boldsymbol{x}, \boldsymbol{z})}{\int p_A(\boldsymbol{x}, \boldsymbol{z})\, d\mu(\boldsymbol{z})} = p_A(\boldsymbol{z}|\boldsymbol{x}).$$

Using this result, we can reformulate the maximum likelihood (ML) problem in (2.19) as:

$$\max_{A\in\mathbb{R}^{N\times M},\, q_t\in\mathcal{D}\,\forall t} \hat{L}_T(A, \{q^{(t)}\}) \coloneqq \frac{1}{T}\sum_{t=1}^{T} \hat{\ell}(A, q^{(t)}; \boldsymbol{x}^{(t)}),$$

where $\mathcal{D}$ is the family of all distributions with support $\bar{\Delta}$.

PRISM frames VIA of (**??**) by restricting the family $\mathcal{D}$ of all $\bar{\Delta}$-supported distributions to the Dirichlet family:

$$\mathcal{D}(\alpha) = \{q = D(\cdot; \alpha) \mid \alpha \in \mathbb{R}^N_{++}\}. \tag{2.21}$$

This restriction yields a lower-bound approximation of the ML objective, motivated by two factors. Under $q = \mathcal{D}(\alpha)$, the function $\hat{\ell}(A, q; \boldsymbol{x})$ in (**??**) simplifies to

$$-\hat{\ell}(A, q; \boldsymbol{x}) \propto \frac{1}{2\sigma^2}\mathbb{E}[\|\boldsymbol{x} - A\boldsymbol{z}\|^2] - H(\boldsymbol{z}) = \frac{1}{2\sigma^2}\left(\|\boldsymbol{x} - A\mathbb{E}[\boldsymbol{z}]\|^2 + \mathrm{tvar}(A\boldsymbol{z})\right) - H(\boldsymbol{z}), \tag{2.22}$$

where we denote $\mathbb{E}_{\boldsymbol{z}\sim q}[\cdot] = \mathbb{E}[\cdot]$ for brevity; $H(\boldsymbol{z}) = \mathbb{E}[-\log q(\boldsymbol{z})]$ is the entropy; and $\mathrm{tvar}(x) = \sum_{i=1}^{N} \mathrm{var}(x_i) = \mathrm{tr}(\mathrm{Cov}(x))$. All $\mathbb{E}[\boldsymbol{z}]$, $\mathrm{Cov}(\boldsymbol{z})$, and $H(\boldsymbol{z})$ have explicit expressions, making the ML problem under the Dirichlet restriction (2.21), or VIA-ML, tractable. It can be solved using alternating optimization, where the latent variables $\boldsymbol{z}$ are updated using the optimal q in (**??**), and the vertex matrix A is updated using the latent variables.

The structure of the Dirichlet distribution enables analytical expression of expectations and resulting in a lightweight algorithm. This structure simplifies inference, but the Dirichlet assumption is often restrictive for broader applications. For more flexible distributions, the Importance Sampling Approximation (ISA) can be applied to approximate $\hat{\ell}(A, q_t; \boldsymbol{x}_t)$ with a large number of independent samples:

$$\hat{\ell}(A, q_t; \boldsymbol{x}_t) \approx \frac{1}{R}\sum_{r=1}^{R}\left(\log p_A(\boldsymbol{x}_t, \boldsymbol{z}_t^{(r)}) - \log q_t(\boldsymbol{z}_t^{(r)})\right), \tag{2.23}$$

where $\{\boldsymbol{z}_t^{(r)}\}_{r=1}^R$ are samples drawn from the distribution q_t, typically chosen as $q_t \propto \phi_\sigma(\boldsymbol{x}_t - A\boldsymbol{z}_t)\, 1_\Delta(\boldsymbol{z}_t)$. Sampling can be implemented using rejection sampling or Markov Chain Monte

Carlo (MCMC) methods [47, 51]. However, generating effective samples under the structural constraints of SCA poses significant challenges. For instance, in rejection sampling over 99.9% of samples may be discarded, when $M > 10$ [51], rendering this method impractical for moderate M.

2.5.3 Variational Simplex Component Analysis

The VAE-based Simplex Component Analysis (VASCA) approaches SCA problem by framing it within VAE framework. It employs the variational ML framework by utilizing the LN distribution as a variational posterior, and leveraging neural networks to model its parameters. As an LN distribution belongs to the location-scale family, it permits a straightforward reparameterization trick, which enables efficient, gradient-based optimization. This setup integrates the required probabilistic structure for SCA while allowing for expressive and scalable posterior approximations.

Specifically, the logistic-normal variational posterior $q_{\boldsymbol{\phi}} \in \mathcal{LN}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$ is parameterized by the neural networks $\boldsymbol{\mu}(\boldsymbol{x})$ and $\boldsymbol{\sigma}(\boldsymbol{x})$, where $\boldsymbol{\Sigma}(\boldsymbol{x}) = \mathrm{Diag}(\boldsymbol{\sigma}(\boldsymbol{x})^2)$. Given (**??**) and (2.18), the SGVB estimator (2.12) can be written as

$$\tilde{\ell}(\boldsymbol{\theta}, \boldsymbol{\phi}; \boldsymbol{x}) = \frac{1}{R} \sum_{r=1}^{R} \left[\log p_{\boldsymbol{\theta}}(\boldsymbol{x}|\boldsymbol{z}^{(r)}) + \log p(\boldsymbol{z}^{(r)}) - \log q_{\boldsymbol{\phi}}(\boldsymbol{z}^{(r)}|\boldsymbol{x}) \right], \tag{2.24}$$

where $\boldsymbol{z}^{(r)} = g(\boldsymbol{\mu}(\boldsymbol{x}) + \boldsymbol{\sigma}(\boldsymbol{x}) \odot \boldsymbol{\epsilon}^{(r)})$, and $\boldsymbol{\epsilon}^{(r)} \sim \mathcal{N}(0, \boldsymbol{I})$. The uniform prior in (**??**) yields a constant $\log p(\boldsymbol{z}) = \log \Gamma(M)$.

Substituting these expressions back into equation (2.24) yields expression for the optimization objective:

$$\tilde{\ell}(\boldsymbol{\theta}, \boldsymbol{\phi}; \boldsymbol{x}) = \frac{1}{R} \sum_{r=1}^{R} \left[-\frac{1}{2\sigma^2} \ell_{rec} + h_{\phi}(\boldsymbol{z}^{(r)}; \boldsymbol{x}) \right] + C, \tag{2.25}$$

where the point-wise entropy is

$$h_{\phi}(\boldsymbol{z}; \boldsymbol{x}) = \frac{1}{2} \tilde{z}^{\top} \mathrm{Diag}(\boldsymbol{\sigma}(\boldsymbol{x}))^{-1} \tilde{z} + \frac{1}{2} \sum_{i=1}^{M-1} \log \sigma_i(\boldsymbol{x}) + \sum_{i=1}^{M} \log z_i, \tag{2.26}$$

the reconstruction loss is

$$\ell_{rec} = \sum_{i=1}^{M} \left(x_i - \sum_{j=1}^{N} A_{ij} z_j \right)^2. \tag{2.27}$$

and constant term C is given by

$$C = \frac{1}{2} \log(2\pi) + \log \Gamma(M) - \frac{N}{2} \log(2\pi\sigma^2).$$

Chapter 3: State of the Art

3.1 Identifiability in Latent Variable Models

Identifiability is an essential property of statistical models, which guarantees interpretability of estimated parameters and latent variables, independently of estimation method. This property implies that distinct parameter values correspond to distinct distributions over the observations, allowing for the unique recovery of model parameters and latent variables from data. Formally, a model $\mathcal{P} = \{p_\theta : \theta \in \Theta\}$ defined over a set of observations $x \in \mathcal{X}$ is identifiable if the mapping $\theta \mapsto p_\theta(x)$ is injective, in other words:

$$p_\theta(x) = p_{\theta^*}(x) \implies \theta = \theta^*, \quad \forall x \in \mathcal{X}.$$

In practical scenarios, we are often interested in models that are identifiable up to a certain class of transformations.

Definition 1. *Let $\sim$ be an equivalence relation on the parameter space Θ. We say that a model is identifiable up to $\sim$ (or $\sim$-identifiable) on $\mathcal{X}$ if*

$$p_\theta(x) = p_{\theta^*}(x) \implies \theta \sim \theta^*, \quad \forall x \in \mathcal{X}, \tag{3.1}$$

where $p_\theta(x)$ denotes the probability distribution parameterized by θ. The elements of the quotient space $\Theta/\sim$ are referred to as the identifiability classes.

In the following, we will encounter various types of equivalence relations, specific to the generative model under consideration. Linear equivalence relations appear in deterministic nonlinear models, where identifiability is only guaranteed up to a linear transformation of the latent variables. A stricter form of equivalence arises when latent components can be uniquely recovered up to permutation and scaling transformations. For example, in the factor analysis model,

$$\boldsymbol{x} = A\boldsymbol{z} + \boldsymbol{v}, \tag{3.2}$$

identifiability implies the equivalence relation on the estimated mixing matrix A.

Definition 2. *Let $\sim$ be an equivalence relation on $\mathbb{R}^{N\times M}$ defined as follows: $A \sim A^*$ if and only if there exists a matrix $W \in \mathbb{R}^{N\times N}$, such that*

$$\boldsymbol{\theta} = W\boldsymbol{\theta}^* \quad \forall \boldsymbol{x} \in \mathcal{X},$$

If W is invertible, we denote this relation by $\sim_U$. If W is a block permutation matrix, we denote it by $\sim_P$, if W is a scaled permutation matrix, we denote it by $\sim_S$.

However, when the model assumes invariance in latent distribution $p(\boldsymbol{z})$, identifiability breaks down. For instance, in Gaussian factor analysis,

$$\boldsymbol{x} = A\boldsymbol{z} + \boldsymbol{v}, \quad \boldsymbol{v} \sim \mathcal{N}(0, \sigma^2\boldsymbol{I}), \quad \boldsymbol{z} \sim \mathcal{N}(0, \boldsymbol{I}), \tag{3.3}$$

the $p(\boldsymbol{z})$ is spherically symmetric, i.e. any rotation of $\boldsymbol{z}$ preserves the distribution. For example, if $p(\boldsymbol{z})$ follows a spherical Gaussian distribution, any orthogonal transformation of $\boldsymbol{z}$ will leave $p(\boldsymbol{z})$ unchanged, which implies $\sim_U$ equivalence relation on A. While $p_A(\boldsymbol{x})$ remains constant, these transformations change $p_A(\boldsymbol{z}|\boldsymbol{x})$, making it impossible to uniquely recover the true posterior distribution and the original latent structure in the model, also known as factor rotation indeterminacy [25]. Similarly, optimizing the marginal likelihood of observed data in VAE does not inherently guarantee that the correct joint distribution over observed and latent variables will be learned. This limitation arises in many VAE implementations, which often result in non-identifiable models [38]. Incorporating structural assumptions, such as conditional or constrained priors, helps break this symmetry and permits unique recovery of the model's parameters and latent variables.

Identifiability typically refers to the ability to uniquely determine the model parameters, but it can be extended to the latent variables, though defining it rigorously can be challenging. In a noise-free factor analysis model, knowing the mixing matrix A allows retrieval of the latent components $\boldsymbol{z}$ via the (pseudo)inverse of A. However, in the presence of noise, full recovery of $\boldsymbol{z}$ from A alone is not possible, instead identifiability refers to the ability to recover the posterior $p_A(\boldsymbol{z}|\boldsymbol{x})$ [32].

In the following, we discuss linear and nonlinear identifiability in key ICA and VAE models, focusing on the role of structural assumptions, and general methodology for establishing identifiability. We also provide a high-level proofs of identifiability emphasizing key techniques. We begin by discussing identifiability in ICA, a foundational framework for recovering statistically independent, non-Gaussian latent components from observed linear mixtures. ICA's identifiability principles [12], serve as a blueprint, introducing essential techniques for identifiability proofs. Next, we discuss the nonlinear ICA (nICA) [28] leverages auxiliary observations to disentangle latent sources. This model establishes a framework for handling nonlinear identifiability.

We further explore latent variable models with simplex-structured latent spaces, which introduce an alternative approach to identifiability based on geometric constraints on the prior. In the Simplex-Constrained Post-Nonlinear Model (SC-PNM), the identifiability is achieved by constraining the latent space to a unit simplex. This model leverages the geometry of the simplex to establish identifiability for a specific class of nonlinear mappings, and introduces techniques that allow to convert simplex constraints into equivalence relations.

Finally, we examine identifiability in noisy generative models. Formally, in the presence of noise we cannot identify latents from the observed data, even if we know the true parameters. In this case, identifiability of the model is framed as matching the true posterior and the conditional likelihood. We first discuss the auxiliary-variable-based VAE (iVAE) that utilizes auxiliary variables to structure the latent space and condition the prior, which enables nonlinear identifiability. This model introduces important techniques for establishing equivalence relations in noisy models. The Probabilistic Simplex Component Analysis (PRISM) further adapts these probabilistic techniques to simplex-structured LVMs, applying the framework to cases with geometrically constrained latent variables. Through techniques such as conditional priors and structured variational inference, iVAE and PRISM establish identifiability in complex

probabilistic models where traditional approaches would struggle.

3.2 Deterministic Models

3.2.1 Independent Component Analysis

ICA is closely related to the Gaussian factor model. It assumes that the observed vector $\boldsymbol{x} \in \mathbb{R}^N$ is a linear combination of latent, statistically independent sources $\boldsymbol{z} = [z_1, \ldots, z_N]^\top$, represented as

$$\boldsymbol{x} = A\boldsymbol{z}, \tag{3.4}$$

where $A \in \mathbb{R}^{N \times N}$ is an invertible mixing matrix [**?**, **?**, **?**, **?**, **?**]. Here, $\boldsymbol{z} \in \mathbb{R}^N$ represents independent sources that are non-Gaussian, zero-centered, and with unit variance. ICA seeks to estimate both the matrix A and the sources $\boldsymbol{z}$ from observed data $\boldsymbol{x}$ by leveraging the non-Gaussianity of the sources, which is essential for identifiability.

The non-Gaussian assumption fundamentally distinguishes ICA from classical factor analysis, and enables identifiability. It allows exploiting additional information contained in higher-order statistical properties [**?**, **?**, **?**], such as kurtosis, to break the rotational symmetry and achieve identifiability. ICA typically assumes that the number of observed variables matches the number of independent sources, i.e., $M = N$, so that A is square and invertible. In practice, the dimension is often reduced by PCA as a preprocessing step. Hence, ICA can be seen as a "factor rotation", considering the principal components as estimates of factors [25]. Te lack of a noise term in ICA is compensated by the high number of components, which together account for all data variance, implicitly capturing both noise and signal.

ICA identifies latent components by finding an invertible transformation that maximizes the non-Gaussianity of the resulting outputs. The intuition behind this approach stems from the Central Limit Theorem, which implies that a sum of independent random variables is generally more Gaussian than each individual variable, especially if they share the same distribution. Therefore, among possible transformations, the one producing the least Gaussian (most non-Gaussian) component aligns with a single source variable, maximizing the non-Gaussianity of each transformed component. For further insights, refer to [**?**, **?**].

However, ICA involves two fundamental ambiguities: first, the ordering of components remains undefined, and second, each component can only be estimated up to arbitrary scale and sign, as multiplying a component by a constant with a corresponding adjustment in the columns of A leaves the data distribution unchanged. Conventionally, setting component variances to unity reduces this scale ambiguity, standardizing the components to a *white* form, though the exact variance of the sources themselves remains unidentifiable.

In terms of the definition in (**??**), we say that ICA is identifiable up to the equivalence relation $\sim_S$, i.e.

$$p_A(\boldsymbol{x}) = p_{A^*}(\boldsymbol{x}) \implies A = SA^*, \quad \forall \boldsymbol{x} \in \mathbb{R}^N, \tag{3.5}$$

where S is a scaled permutation matrix. In the ICA context, identifiability implies that if two

different mixing matrices A and A^* yield the same distribution of $\boldsymbol{x}$, then they must be equivalent up to scaling and permutation of columns. The following theorem provides a formal statement and the high-level proof of identifiability for ICA:

Theorem 1. *Assume that the independent components z_i have finite variance and that their log-pdfs $\log p_{z_i}(z_i)$ are twice continuously differentiable. Then, for a linear mixture model $\boldsymbol{x} = A\boldsymbol{z}$ with non-Gaussian z_i, the mixing matrix A is $\sim_S$-identifiable, i.e. has exactly one non-zero entry in each row and column.*

Proof. Without loss of generality, A is assumed orthogonal, which can be achieved by whitening $\boldsymbol{x}$ beforehand. Whitening standardizes the covariance of $\boldsymbol{z}$ and forces A to be orthogonal, simplifying the proof.

The identifiability proof in ICA proceeds in three steps: First, $p(\boldsymbol{x})$ is expressed in terms of the densities of the independent, non-Gaussian latent components $\boldsymbol{z}$, leveraging their independence to factorize $p(\boldsymbol{x})$. Next, second derivatives of $\log p(\boldsymbol{x})$ with respect to $\boldsymbol{x}$ are taken, where independence implies that cross-terms vanish, leading to a matrix equation that isolates each z_i's influence. Finally, this matrix equation forms an eigenvalue decomposition (EVD). The orthogonality of A and distinct values from the non-Gaussianity of $\boldsymbol{z}$ ensure a unique EVD up to permutation and sign changes, establishing the identifiability of $\boldsymbol{z}$.

To express the probability density function of $\boldsymbol{x}$ as a function of the densities of z_i, we apply the change of variables formula:

$$p(\boldsymbol{x}) = \prod_{i=1}^{n} p_{z_i}(y_i) \left|\det(A^\top)\right|, \tag{3.6}$$

where

$$y_i = \sum_{j=1}^{n} a_{ji} x_j$$

and p_{z_i} denotes the pdf of each component z_i. Taking the logarithm of $p(\boldsymbol{x})$, we obtain

$$\log p(\boldsymbol{x}) = \sum_{i=1}^{n} \log p_{z_i}(y_i). \tag{3.7}$$

Since the observed variables x_i are assumed to be independent, we can decompose $\log p(\boldsymbol{x}) = \sum_{i=1}^{n} f_i(x_i)$ for some functions f_i, indicating that all second-order cross-derivatives must be zero:

$$\frac{\partial^2 \log p(\boldsymbol{x})}{\partial x_k \partial x_l} = \sum_{i=1}^{n} a_{ki} a_{li} \left(\log p_{z_i}(y_i)\right)'' = 0, \quad \forall k \neq l. \tag{3.8}$$

In matrix form, this system can be expressed as the eigenvalue decomposition (EVD):

$$A^\top \operatorname{diag}\left(\log p_{z_i}(y_i)\right)'' A = \operatorname{diag}(c_i(\boldsymbol{x}; A)) \tag{3.9}$$

where $c_i(\boldsymbol{x}; A)$ are unknown scalar functions, and which is valid for all $\boldsymbol{x}$.

The uniqueness of EVD here shows why Gaussian components lack identifiability. For Gaussian densities, $\log p_{z_i}$ is quadratic, yielding a constant second derivative $(\log p_{z_i})''$ and reducing the left side of equation (3.9) to an identity matrix, permitting any orthogonal transformation of A. In contrast, for non-Gaussian densities, the second derivative of $\log p_{z_i}$ is non-constant, allowing to choose points $\boldsymbol{x}$ such that each diagonal entry is distinct. By the uniqueness of EVD, this requirement enforces that the components of A are identifiable up to permutation and scaling of columns, concluding the proof. □

An alternative approach to the proof utilizes the Fourier domain by working with characteristic functions $\hat{p}$. Here, $p(x)$ and $p_{z_i}(z_i)$ are replaced by their characteristic functions, eliminating the need for the Jacobian in the initial equations. This approach replaces the smoothness assumption of the probability density functions with the requirement of continuous second derivatives for the characteristic functions $\hat{p}_{z_i}$, which is slightly more restrictive than assuming finite variances.

3.2.2 Nonlinear ICA with Auxiliary Variables

The fundamental concept of ICA is extended here to the nonlinear setting, where the observed vector $\boldsymbol{x} \in \mathbb{R}^N$ is generated by a nonlinear, invertible mixing function $\boldsymbol{f} : \mathbb{R}^N \to \mathbb{R}^N$, yielding model

$$\boldsymbol{x} = \boldsymbol{f}(\boldsymbol{z}). \tag{3.10}$$

This mixing function $\boldsymbol{f}$ is assumed to be smooth, with continuous second derivatives, and it need not follow any specific functional form. Consequently, $\boldsymbol{f}$ can be modeled by a neural network, and empirical results suggest that it generally maintains invertibility without explicit constraints.

To estimate the model, a contrastive learning approach is employed, which distinguishes between real and randomized datasets. Specifically, two datasets are defined as $\tilde{x} = (x, u)$ and $\tilde{x}^* = (x, u^*)$, where u^* is a randomly selected, independent sample drawn from the distribution of u, generated by permuting the empirical observations of u. The estimation procedure uses a nonlinear logistic regression model (e.g., a neural network) with a regression function defined as

$$r(x, u) = \sum_{i=1}^{N} \psi_i(h_i(x), u),$$

which yields the posterior probability of the original class as $(1 + \exp(-r(x, u)))^{-1}$. This contrastive approach effectively leverages the auxiliary variable u for enhanced separation of the latent structure.

For identifiability, the approach assumes that each latent component z_i in the vector $\boldsymbol{z} = [z_1, \ldots, z_N]^\top$ is conditionally dependent on an observed m-dimensional auxiliary variable $\boldsymbol{u}$, while

being conditionally independent of the other components:

$$\log p(\boldsymbol{z}|\boldsymbol{u}) = \sum_{i=1}^{n} q_i(z_i, \boldsymbol{u}), \tag{3.11}$$

for some functions q_i. The auxiliary variable $\boldsymbol{u}$ is application-specific and may represent temporal information (e.g., previous time steps in time series), spatial indices (e.g., pixel indices for image data), or other contextual variables such as class labels. This formulation generalizes the independence assumption in linear ICA by introducing conditional dependencies modulated by $\boldsymbol{u}$, which is essential for separating the components in nonlinear settings.

Theorem 2 ([28]). *Assume:*

1. *The observed data $\boldsymbol{x}$ follows the nonlinear ICA model with auxiliary variables in equations* (3.10), (3.11), *where $f : \mathbb{R}^N \to \mathbb{R}^N$ is an invertible, smooth mixing function, and $\boldsymbol{z} = [z_1, \ldots, z_N]^\top$ is the latent vector.*

2. *The latent components z_i are conditionally dependent on an auxiliary variable $\boldsymbol{u}$, with conditional independence among the components given $\boldsymbol{u}$:*

$$\log p(\boldsymbol{z}|\boldsymbol{u}) = \sum_{i=1}^{n} q_i(z_i, \boldsymbol{u}), \tag{3.12}$$

   where q_i are smooth functions.

3. *The auxiliary variable $\boldsymbol{u}$ is diverse enough to satisfy the* Assumption of Variability*: For any $y \in \mathbb{R}^N$, there exist $2n+1$ values of $\boldsymbol{u}$, denoted by $\boldsymbol{u}_j$, $j = 0, \ldots, 2n$, such that the set of vectors*

$$\{w(y, \boldsymbol{u}_j) - w(y, \boldsymbol{u}_0) \mid j = 1, \ldots, 2n\}$$

   is linearly independent, where

$$w(y, \boldsymbol{u}) = \left(\frac{\partial q_1}{\partial z_1}(y_1, \boldsymbol{u}), \ldots, \frac{\partial q_N}{\partial z_N}(y_N, \boldsymbol{u}), \frac{\partial^2 q_1}{\partial z_1^2}(y_1, \boldsymbol{u}), \ldots, \frac{\partial^2 q_N}{\partial z_N^2}(y_N, \boldsymbol{u}) \right).$$

4. *The regression function in the learning model has universal approximation capability and is trained to discriminate between pairs $(\boldsymbol{x}, \boldsymbol{u})$ and $(\boldsymbol{x}, \boldsymbol{u}^*)$ with $\boldsymbol{u}^*$ permuted independently of $\boldsymbol{x}$.*

Then, as the amount of data $T \to \infty$, the functions $h_i(\boldsymbol{x})$ in the learned regression function recover the independent components z_i, up to invertible scalar transformations.

This approach builds on the ICA proof's foundation and leverages variability in auxiliary variables to resolve the linear system formed by cross-derivative equations.

Proof. According to [22], after convergence with infinite data, the learned regression function approximates the difference in log-densities between two classes. Specifically, the regression

function can be expressed as:

$$\sum_{i=1}^{n} \psi_i(h_i(x), u) = \sum_{i=1}^{n} q_i(g_i(x), u) + \log p(u) - \log p_s(g(x))$$

where $g = f^{-1}$, p_s is the marginal density over the latent components integrated over u, and $h(x)$ and $v(y) = g(h^{-1}(y))$ are derived from a change of variables. In this equation, the Jacobian determinants and the marginal $\log p(u)$ terms cancel. This expression parallels the ICA framework, where the density of x is expressed as a product of the densities of independent components. However, here the inclusion of auxiliary variables u introduces additional structure, which will later facilitate identifiability.

The second step involves taking first and second derivatives of the transformed density expression to analyze the dependencies between components. Differentiating both sides of the equation with respect to y_j yields:

$$\psi_j'(y_j, u) = \sum_{i=1}^{n} q_i'(v_i(y), u) v_j^i(y) - \bar{q}_j(y)$$

where $v_j^i(y) = \frac{\partial v_i}{\partial y_j}$ and $\bar{q}(y) = \log p_s(v(y))$. A subsequent derivative with respect to $y_{j'}$ (for $j' \neq j$) introduces cross-derivative terms:

$$\sum_{i=1}^{n} q_i''(v_i(y), u) v_j^i(y) v_{j'}^i(y) + q_i'(v_i(y), u) v_{jj'}^i(y) - \bar{q}_{jj'}(y) = 0$$

where $v_{jj'}^i(y)$ represents second-order cross-derivatives. This step mirrors the cross-derivative analysis in the linear ICA proof, where the independence of components implies vanishing cross-terms. However, here, the introduction of auxiliary variables and nonlinearity results in new terms, requiring a more complex analysis to establish that each component v_i depends on only one y_i.

The final step utilizes the variability in auxiliary variables u to resolve the linear system formed by these cross-term equations. By collecting the cross-derivative conditions into a matrix $M(y)$ with size $n(n-1)/2 \times 2n$, the equations can be expressed as:

$$M(y)w(y, u) = c(y)$$

with $w(y, u)$ representing terms involving the first and second derivatives of q_i. Auxiliary variables u are selected with values $u_0, u_1, \ldots, u_{2n}$, yielding:

$$M(y)(w(y, u_1) - w(y, u_0), \ldots, w(y, u_{2n}) - w(y, u_0)) = 0$$

The linear independence assumption on the columns of w ensures that $M(y)$ must be zero, forcing $a_i(y) = 0$ and $b_i(y) = 0$ for each i. This confirms that each row of the Jacobian of v has only one non-zero entry, meaning each v_i is exclusively a function of a single y_i.

To conclude, the continuity and invertibility of the Jacobian imply that each v_i consistently depends on one y_i, as any deviation would introduce singularities. Thus, each v_i is an invertible function of a single y_i, completing the identifiability proof.

□

3.2.3 Simplex Constrained Post-Nonlinear Mixture

Another way to break the symmetry of the latent space is by constraining the latent variables domain to a lower-dimensional manifold. The Simplex-Constrained Post-Nonlinear Mixture (SC-PNM) model is a nonlinear generalization of the linear latent component model, where the identifiability is enabled by constraining the latent space to the unit simplex [52, 40]. This model applies to scenarios where the data generation process can be decomposed into a linear mixing stage followed by unknown nonlinear scalar distortions on each observation channel. The model can be expressed as follows:

$$\boldsymbol{x} = \boldsymbol{f}(\mathbf{A}\boldsymbol{z}), \tag{3.13}$$

where $\boldsymbol{f}(\boldsymbol{y}) = [f_1(y_1), \ldots, f_N(y_N)]^\top$ is a component-wise nonlinear continuous function, and $\boldsymbol{z} \in \Delta^M$.

The SC-PNM problem is formulated as constructing a transformation $\boldsymbol{g} : \mathbb{R}^N \to \mathbb{R}^N$ that inversely identifies the nonlinear functions $\boldsymbol{f}(\cdot)$, satisfying the following conditions:

$$\sum_{n=1}^{N} g_n(x_n) = 1, \quad \forall \boldsymbol{x} \in X, \tag{3.14}$$

where $X = \{\boldsymbol{x} \in \mathbb{R}^N | \boldsymbol{x} = \boldsymbol{f}(A\boldsymbol{z}), \forall \boldsymbol{z} \in \Delta^M\}$, and each $g_n : \mathbb{R} \to \mathbb{R}$ is an invertible, scalar-valued function.

To solve for the nonlinear functions $\boldsymbol{g}(\cdot)$, SC-PNM model leverages a neural network-based autoencoder framework. The objective is to reconstruct each observation $\boldsymbol{x}_t$ through an invertible transformation and a subsequent nonlinear mapping. This can be formalized as the following optimization problem:

$$\min_{\boldsymbol{\theta}} \frac{1}{N} \sum_{t=1}^{N} \|\boldsymbol{q}(\boldsymbol{g}(\boldsymbol{x}_t)) - \boldsymbol{x}_t\|_2^2, \tag{3.15}$$

where $\boldsymbol{q}(\cdot) = [q_1(\cdot), \ldots, q_N(\cdot)]^\top$ is a neural decoder. The reconstruction error is minimized under the constraint:

$$\mathbf{1}^\top \boldsymbol{g}(\boldsymbol{x}_t) - 1 = 0, \quad \forall t \in [T], \tag{3.16}$$

where this condition ensures that the transformed data $\boldsymbol{g}(\boldsymbol{x}_t)$ remains on the simplex.

Before proceeding to the identifiability theorem we provide two technical lemmas, that will be used in our formalism in Chapter 3.

Fact 1. *Assume $M \geq 3$. Consider $\boldsymbol{z} = [z_1, \ldots, z_M]^\top \in \overline{\Delta}^M$. Then, $\partial z_i / \partial z_j = 0$ for $i \neq j$ where $i, j \in [M-1]$.*

Proof. For $\boldsymbol{z} \in \overline{\Delta}^M$, we have $M-1$ free variables, e.g., z_i for $i = 1, \ldots, M-1$. Assume

$i, j \in [M-1]$ and $i \neq j$. For any fixed z_i, z_j can take any values within a nonempty continuous domain. Thus, treating z_i as a function of z_j implies $\partial z_i / \partial z_j = 0$ for $\boldsymbol{z} \in \overline{\Delta}^M$. □

Theorem 3. *Assume that:*

1. *The (full-rank?) mixing matrix $A \in \mathbb{R}^{N \times M}$ is drawn from a continuous distribution.*
2. *Each $h_n = g_n \circ f_n$ is twice differentiable and invertible.*
3. *The dimension constraints $3 \leq M \leq N \leq \frac{M(M-1)}{2}$ hold.*

Then, almost surely, any solution $\boldsymbol{g} = [g_1, \ldots, g_N]^\top$ that satisfies (3.14) ensures that each $h_n(y) = g_n \circ f_n(y)$ is affine, specifically:

$$h_n(y) = c_n y + d_n, \quad c_n \neq 0, \quad d_n \in \mathbb{R}. \tag{3.17}$$

Moreover, if $\sum_{n=1}^N d_n \neq 1$, we have $\boldsymbol{h}(A\boldsymbol{z}) = \hat{A}\boldsymbol{z}$, where $\hat{A} = DA$ with a full-rank diagonal matrix D, i.e. $\boldsymbol{h}$ is a linear function on a simplex $\mathcal{A} = \{A\boldsymbol{z} \in \mathbb{R}^M | \sum_{m=1}^M z_m = 1, z_m \geq 0\}$, or the model is $\sim_A$-identifiable on $\mathcal{A}$.

Proof. Solving criterion (3.14) leads to

$$\sum_{i=1}^N \hat{h}_i \left(a_{i,1} z_1 + \cdots + a_{i,M} \left(1 - \sum_{j=1}^{M-1} z_j \right) \right) = 1,$$

using $z_M = 1 - \sum_{j=1}^{M-1} z_j$, and Fact 1 ensures that $\partial z_i / \partial z_j = 0$ for $i \neq j$. Taking second-order derivatives with respect to z_i and z_j for $i, j \in [M-1]$ results in the system:

$$G\hat{\boldsymbol{h}}'' = \begin{bmatrix} (b_1 \odot b_1)^\top \\ \vdots \\ (b_{M-1} \odot b_{M-1})^\top \\ (b_1 \odot b_2)^\top \\ \vdots \\ (b_{M-2} \odot b_{M-1})^\top \end{bmatrix} \begin{bmatrix} \hat{h}_1'' \\ \vdots \\ \hat{h}_N'' \end{bmatrix} = 0, \tag{3.18}$$

where $\boldsymbol{b}_i = [a_{1,i} - a_{1,M}, \ldots, a_{N,i} - a_{N,M}]^\top$ for $i = 1, \ldots, M-1$, and $B = [\boldsymbol{b}_1, \ldots, \boldsymbol{b}_{M-1}]$. Here, G has dimensions $\frac{M(M-1)}{2} \times N$, and we aim to establish that $\operatorname{rank}(G) = N$.

This rank condition can be shown by constructing a specific case where an $N \times N$ submatrix of G has full rank. Consider a scenario in which B is a Vandermonde matrix, such that $b_i = [1, z_i, z_i^2, \ldots, z_i^{N-1}]^\top$ with $z_i \neq z_j$. Such a matrix B can be constructed by ensuring that the first $N-1$ rows of $A^\top$ form a Vandermonde matrix, with the last row containing all zeros. By selecting $\tilde{M}$ columns from this Vandermonde matrix B, with $\tilde{M} \leq M-1$, we satisfy the inequality $N \leq \frac{\tilde{M}(\tilde{M}+1)}{2}$. For simplicity, we assume $N = \frac{\tilde{M}(\tilde{M}+1)}{2}$ in what follows. Now, consider

the structure of the submatrix in G formed by the selected rows. This submatrix takes the form:

$$\begin{bmatrix} 1 & z_1^2 & \dots & z_1^{2(N-1)} \\ \vdots & \vdots & & \vdots \\ 1 & z_{\tilde{M}}^2 & \dots & z_{\tilde{M}}^{2(N-1)} \\ 1 & z_1 z_2 & \dots & (z_1 z_2)^{N-1} \\ \vdots & \vdots & & \vdots \\ 1 & z_{\tilde{M}-1} z_{\tilde{M}} & \dots & (z_{\tilde{M}-1} z_{\tilde{M}})^{N-1} \end{bmatrix}.$$

If we can find a particular case where an $N \times N$ submatrix of G has a non-zero determinant, then by continuity, this property holds almost everywhere for A. One can construct a sequence of values for z_i, such as $z_1 = 1$, $z_2 = 1.1$, $z_3 = 1.11$, and so forth, ensuring that the resulting matrix has full rank. Since the determinant of any $N \times N$ submatrix of G is a polynomial in the entries of A, it follows that this polynomial is either identically zero or non-zero almost everywhere [11] Thus, G has full column rank N almost surely.

Next, we follow the linearity arguments presented in [52], Remark 1, to conclude that $\hat{\boldsymbol{h}}(A\boldsymbol{z})$ is linear in $A\boldsymbol{z}$ if $\sum_{n=1}^{N} d_n \neq 1$. Define

$$T_h(X) = \Big[\boldsymbol{h}(\boldsymbol{x}_1), \quad \boldsymbol{h}(\boldsymbol{x}_2), \quad \cdots, \quad \boldsymbol{h}(\boldsymbol{x}_T)\Big],$$

where $X = [\boldsymbol{x}_1, \boldsymbol{x}_2, \cdots, \boldsymbol{x}_T] \in \mathbb{R}^{M \times T}$. Given that $\boldsymbol{h}$ is affine, we can write

$$T_h(X) = DX + \boldsymbol{c}\mathbf{1}_T^\top,$$

where $D = \mathrm{diag}(d_1, \dots, d_M)$ is a diagonal matrix and $\boldsymbol{c} = [c_1, c_2, \dots, c_M]^\top$. To establish that T_h is linear, consider the row sum of $T_h(X)$:

$$\mathbf{1}_M^\top T_h(X) = \mathbf{1}_M^\top DX + \mathbf{1}_M^\top \boldsymbol{c}\mathbf{1}_T^\top.$$

Condition in 3.16 yields

$$\mathbf{1}_M^\top T_h(X) = \mathbf{1}_T^\top,$$

which implies

$$\mathbf{1}_T^\top = \frac{\mathbf{1}_M^\top DX}{1 - \mathbf{1}_M^\top \boldsymbol{c}}.$$

Consequently, $T_h(X)$ is linear, and we can write

$$T_h(X) = \left(\mathbf{I} + \frac{\boldsymbol{c}\mathbf{1}_M^\top}{1 - \mathbf{1}_M^\top \boldsymbol{c}}\right) DX,$$

establishing that $\boldsymbol{h}$ is linear, given that $\sum_{n=1}^{N} c_n \neq 1$ (this is assumed to be true given that $\square$

Theorem 3 establishes that identifiability in SC-PNM models is guaranteed only if $N \leq \frac{M(M-1)}{2}$. This condition is counterintuitive from a conventional LMM perspective, where adding more channels N typically increases flexibility and improves performance. However, in SC-PNM

learning, having more channels can complicate the model, as each additional $g_n(\cdot)$ introduces unknown nonlinearity that must align with the overall solution structure in (3.18). In cases where N exceeds this threshold, SC-PNM's identifiability cannot be guaranteed with criterion 3.14, as shown in Theorem 4:

Theorem 4. *If $M > \frac{K(K-1)}{2}$, then solutions f_m satisfying criterion* (3.14) *can lead to $h_m = g_m \circ f_m$ functions that are not affine.*

This finding highlights that criterion 3.14 alone is insufficient when M is large relative to K. Fortunately, a modification to the criterion resolves this: instead of applying it globally, one can impose the constraint segment-by-segment as follows:

$$\mathbf{1}^\top f\left([\boldsymbol{x}]_{(p-1)K+1:pK}\right) = 1, \quad p \in \left[\frac{M}{K}\right],$$

assuming M/K is an integer. For non-integer cases, overlapping segments provide a practical extension, ensuring the SC-PNM model remains identifiable and scalable for higher-dimensional data.

Finding a feasible solution for Problem (3.15) is challenging, as feasibility requires satisfying all equality constraints. Since any Karush–Kuhn–Tucker (KKT) point is feasible, efficient KKT-point searching algorithms from nonlinear programming can be leveraged. SC-PNM we utilizes the augmented Lagrangian $L(\boldsymbol{\theta}, \boldsymbol{\lambda})$, defined by:

$$L(\boldsymbol{\theta}, \boldsymbol{\lambda}) = \frac{1}{N}\sum_{\ell=1}^{N} J_{\boldsymbol{\theta}}(\boldsymbol{x}_\ell) + \frac{1}{N}\sum_{\ell=1}^{N} \lambda_\ell C_{\boldsymbol{\theta}}(\boldsymbol{x}_\ell) + \frac{\rho}{2N}\sum_{\ell=1}^{N} |C_{\boldsymbol{\theta}}(\boldsymbol{x}_\ell)|^2,$$

where $J_{\boldsymbol{\theta}}(\boldsymbol{x}_\ell) = \|q(\boldsymbol{f}(\boldsymbol{x}_\ell)) - \boldsymbol{x}_\ell\|_2^2$ and $C_{\boldsymbol{\theta}}(\boldsymbol{x}_\ell) = \mathbf{1}^\top \boldsymbol{f}(\boldsymbol{x}_\ell) - 1$. The dual variables $\boldsymbol{\lambda} = [\lambda_1, \ldots, \lambda_N]$ and $\rho > 0$ control constraint adherence.

The algorithm iteratively updates $\boldsymbol{\theta}$ and $\boldsymbol{\lambda}$ as follows:

$$\boldsymbol{\theta}^{t+1} \leftarrow \arg\min_{\boldsymbol{\theta}} L(\boldsymbol{\theta}, \boldsymbol{\lambda}^t) \quad \text{(inexact minimization)}, \tag{17a}$$

$$\lambda_\ell^{t+1} \leftarrow \lambda_\ell^t + \rho^t C_{\boldsymbol{\theta}^{t+1}}(\boldsymbol{x}_\ell), \quad \rho^{t+1} \leftarrow \kappa\rho^t, \tag{17b}$$

where $\kappa > 1$ is a fixed parameter. This method belongs to augmented Lagrangian techniques [4]. Classic results ensure that if $\|\nabla L(\boldsymbol{\theta}^{t+1}, \boldsymbol{\lambda}^t)\|_2^2 \le \epsilon_t \to 0$, then all limit points are KKT points.

In practice, SGD-based optimizers like Adam are suitable for updating neural network parameters $\boldsymbol{f}$ and q. As $\rho_t \to \infty$ may cause instability, practical alternatives include incrementing ρ_t gradually or keeping it fixed, which both perform well in simulations. Nevertheless, the algorithm exhibits slow convergence, particularly for large-scale problems as it requires multiple passes over the data to ensure constraint satisfaction.

3.3 Probabilistic Models

3.3.1 Variational Autoencoders with Conditional Prior

In probabilistic terms, identifiability implies that any two different parameter choices, $\boldsymbol{\theta}$ and $\boldsymbol{\theta}'$, yielding the same marginal density $p_{\boldsymbol{\theta}}(\boldsymbol{x})$, must have identical joint distributions $p_{\boldsymbol{\theta}}(\boldsymbol{x}, \boldsymbol{z})$. This equivalence means that if we find a parameter $\boldsymbol{\theta}$ such that $p_{\boldsymbol{\theta}}(\boldsymbol{x}) = p_{\boldsymbol{\theta}^*}(\boldsymbol{x})$, then $p_{\boldsymbol{\theta}}(\boldsymbol{x}, \boldsymbol{z}) = p_{\boldsymbol{\theta}^*}(\boldsymbol{x}, \boldsymbol{z})$, indicating we have recovered the correct prior, $p_{\boldsymbol{\theta}}(\boldsymbol{z}) = p_{\boldsymbol{\theta}^*}(\boldsymbol{z})$, and posterior, $p_{\boldsymbol{\theta}}(\boldsymbol{z}|\boldsymbol{x}) = p_{\boldsymbol{\theta}^*}(\boldsymbol{z}|\boldsymbol{x})$. For VAEs, this identifiability allows us to use an inference model $q_{\boldsymbol{\phi}}(\boldsymbol{z}|\boldsymbol{x})$ to efficiently infer the latent sources $\boldsymbol{z}^*$ from the data $\boldsymbol{x}$.

Achieving identifiability in VAEs thus requires modifying the latent structure to overcome the indeterminacies that arise with unconditional latent priors. For example, using a latent prior that is conditioned on additional observations, such as labels or time indices, can establish identifiability by introducing a structured latent space [31]. Under these conditions, if the learned marginal distribution $p_{\boldsymbol{\theta}}(\boldsymbol{x})$ matches the true data distribution, then the learned joint distribution $p_{\boldsymbol{\theta}}(\boldsymbol{x}, \boldsymbol{z})$ will also match the true joint distribution. This model is closely related to the nICA formalism discussed above, but framed as a probabilistic generative model. It allows for guaranteed removal of nonlinear mixing, up to component-wise transformations.

The proposed model assumes a noisy nonlinear model:

$$\boldsymbol{x} = \boldsymbol{f}(\boldsymbol{z}) + \boldsymbol{v}, \tag{3.19}$$

with an independent noise variable $\boldsymbol{v}$, distributed according to $p_{\boldsymbol{v}}(\boldsymbol{v})$, and a conditionally factorized prior distribution over the latent variables $p_{\boldsymbol{\theta}}(\boldsymbol{z}|\boldsymbol{u})$, where $\boldsymbol{u}$ is an additional observed variable. This auxiliary variable $\boldsymbol{u}$ could represent, for example, a time index in a time series, a noisy class label, or a concurrent observation. Formally, given observed variables $\boldsymbol{x} \in \mathbb{R}^N$ and $\boldsymbol{u} \in \mathbb{R}^N$, a latent variable $\boldsymbol{z} \in \mathbb{R}^M$ (with $M \leq N$), and model parameters $\boldsymbol{\theta} = (\boldsymbol{f}, \boldsymbol{T}, \boldsymbol{\lambda})$, the conditional generative model is defined by

$$p_{\boldsymbol{\theta}}(\boldsymbol{x}, \boldsymbol{z}|\boldsymbol{u}) = p_{\boldsymbol{f}}(\boldsymbol{x}|\boldsymbol{z}) p_{\boldsymbol{T},\boldsymbol{\lambda}}(\boldsymbol{z}|\boldsymbol{u}), \tag{3.20}$$

where

$$p_{\boldsymbol{f}}(\boldsymbol{x}|\boldsymbol{z}) = p_{\boldsymbol{v}}(\boldsymbol{x} - \boldsymbol{f}(\boldsymbol{z})). \tag{3.21}$$

The function $\boldsymbol{f} : \mathbb{R}^M \to \mathbb{R}^N$ is assumed injective and may be arbitrarily complex, often implemented as a neural network in practice.

The latent variable prior $p_{\boldsymbol{\theta}}(\boldsymbol{z}|\boldsymbol{u})$ exhibits a conditionally factorial structure such that each component z_i follows a univariate exponential family distribution conditioned on $\boldsymbol{u}$:

$$p_{T,\lambda}(\boldsymbol{z}|\boldsymbol{u}) = \prod_i Q_i(z_i) Z_i(\boldsymbol{u}) \exp\left(\sum_{j=1}^{J} T_{i,j}(z_i) \lambda_{i,j}(\boldsymbol{u}) \right), \tag{3.22}$$

where Q_i is the base measure, $Z_i(\boldsymbol{u})$ the normalizing constant, $T_i = (T_{i,1}, \ldots, T_{i,J})$ the sufficient

statistics, and $\lambda_i(\boldsymbol{u}) = (\lambda_{i,1}(\boldsymbol{u}), \ldots, \lambda_{i,J}(\boldsymbol{u}))$ the parameters dependent on $\boldsymbol{u}$ through an arbitrary function, e.g., a neural network.

Let $Z \subset \mathbb{R}^M$, $X \subset \mathbb{R}^N$, and $U \subset \mathbb{R}^L$ represent the domain, image of $\boldsymbol{f}$, and the support of $\boldsymbol{u}$, respectively. We denote by $\boldsymbol{T}(\boldsymbol{z}) := (T_1(\boldsymbol{z}_1), \ldots, T_n(\boldsymbol{z}_N))$ and $\boldsymbol{\lambda}(\boldsymbol{u}) := (\lambda_1(\boldsymbol{u}), \ldots, \lambda_n(\boldsymbol{u}))$.

Theorem 5. *Assume that data is generated from the model defined in Eqs.* (3.21)-(3.22) *with parameters* $\boldsymbol{\theta} = (\boldsymbol{f}, \boldsymbol{T}, \boldsymbol{\lambda})$. *Under the following conditions:*

1. *The characteristic function* $\phi_{\boldsymbol{v}}$ *of the noise distribution* $p_{\boldsymbol{v}}$ *is non-zero almost everywhere.*
2. *The function* $\boldsymbol{f}$ *is injective.*
3. *The sufficient statistics* $T_{i,j}$ *are differentiable almost everywhere and linearly independent on any subset of* Z *of positive measure.*
4. *There exist* $MJ + 1$ *distinct points* $\boldsymbol{u}_0, \ldots, \boldsymbol{u}_{MJ}$ *such that the matrix* $L = (\boldsymbol{\lambda}(\boldsymbol{u}_1) - \boldsymbol{\lambda}(\boldsymbol{u}_0), \ldots, \boldsymbol{\lambda}(\boldsymbol{u}_{MJ}) - \boldsymbol{\lambda}(\boldsymbol{u}_0))$ *is invertible,*

the parameters $\boldsymbol{\theta}$ *are identifiable up to a linear transformation.*

Theorem 5 establishes a basic identifiability for the generative model in equation (3.20). Assuming data is generated by parameters (f^*, T^*, λ^*), and a learning algorithm provides a consistent estimate of parameters (f, T, λ), the true parameters are $(f^*, T^*, \lambda^*) \sim_A (f, T, \lambda)$. In the absence of noise, this implies that the learned transformation $\tilde{f}$ reconstructs the latent variables z from observations x, up to a linear transformation A and nonlinear mappings T and $\tilde{T}$. With noise, identifiability holds analogously for the posterior distribution of the latents.

The proof proceeds in three steps, establishing identifiability by transforming the noisy observation model to a noiseless equivalent, isolating the sufficient statistics, and proving invertibility.

Proof. We start by assuming two sets of parameters $\boldsymbol{\theta} = (\boldsymbol{f}, \boldsymbol{T}, \boldsymbol{\lambda})$ and $\boldsymbol{\theta}^* = (\boldsymbol{f}^*, \boldsymbol{T}^*, \boldsymbol{\lambda}^*)$ such that:

$$p_{\boldsymbol{\theta}}(\boldsymbol{x}|\boldsymbol{u}) = p_{\boldsymbol{\theta}^*}(\boldsymbol{x}|\boldsymbol{u}), \quad \forall(\boldsymbol{x}, \boldsymbol{u}). \tag{3.23}$$

Expanding this equality with the noise model from Eq. (3.21), we have

$$\int p_{T,\lambda}(\boldsymbol{z}|\boldsymbol{u}) p_{\boldsymbol{v}}(\boldsymbol{x} - f(\boldsymbol{z})) d\boldsymbol{z} = \int p_{\boldsymbol{T}^*,\lambda^*}(\boldsymbol{z}|\boldsymbol{u}) p_{\boldsymbol{v}}(\boldsymbol{x} - f^*(\boldsymbol{z})) d\boldsymbol{z}. \tag{3.24}$$

Applying a change of variable $\bar{\boldsymbol{x}} = f(\boldsymbol{z})$ and using the Fourier transform $\mathcal{F}$ to both sides, we obtain

$$\mathcal{F}[p_{T,\lambda,f,\boldsymbol{u}}](\omega)\phi_{\boldsymbol{v}}(\omega) = \mathcal{F}[p_{\boldsymbol{T}^*,\lambda^*,f^*,\boldsymbol{u}}](\omega)\phi_{\boldsymbol{v}}(\omega), \tag{3.25}$$

where $\phi_{\boldsymbol{v}}$ is the characteristic function of $p_{\boldsymbol{v}}$. Since $\phi_{\boldsymbol{v}}(\omega) \neq 0$ almost everywhere, we conclude

$$p_{T,\lambda,f,\boldsymbol{u}} = p_{\boldsymbol{T}^*,\lambda^*,f^*,\boldsymbol{u}}, \tag{3.26}$$

which implies equality in the noise-free probability densities.

Taking the logarithm of both sides, we substitute $p_{\boldsymbol{T},\boldsymbol{\lambda}}(\boldsymbol{z}|\boldsymbol{u})$ from Eq. (3.22):

$$\log|\det J_{f^{-1}}(\boldsymbol{x})| + \sum_{i=1}^{n}\left(\log Q_i(f_i^{-1}(\boldsymbol{x})) - \log Z_i(\boldsymbol{u}) + \sum_{j=1}^{J} T_{i,j}(f_i^{-1}(\boldsymbol{x}))\lambda_{i,j}(\boldsymbol{u})\right)$$
$$= \log|\det J_{f^{*-1}}(\boldsymbol{x})| + \sum_{i=1}^{n}\left(\log {Q^*}_i({f^*}_i^{-1}(\boldsymbol{x})) - \log {Z^*}_i(\boldsymbol{u}) + \sum_{j=1}^{J} \boldsymbol{T}^*_{i,j}({f^*}_i^{-1}(\boldsymbol{x}))\lambda^*_{i,j}(\boldsymbol{u})\right). \quad (3.27)$$

Subtracting the equation for $\boldsymbol{u}_0$ from the remaining equations for $\boldsymbol{u}_1, \ldots, \boldsymbol{u}_{MJ}$, we isolate the sufficient statistics:

$$L^T T(\boldsymbol{f}^{-1}(\boldsymbol{x})) = L^T \boldsymbol{T}^*(\boldsymbol{f}^{*-1}(\boldsymbol{x})) + \boldsymbol{b}, \quad (3.28)$$

where $\boldsymbol{b}$ represents constants independent of $\boldsymbol{x}$.

We define the Jacobian $J_{\boldsymbol{T}}$ of $\boldsymbol{T}$ and consider points $\bar{\boldsymbol{x}}_1, \ldots, \bar{\boldsymbol{x}}_J$ where the Jacobian matrix $Q = (J_{\boldsymbol{T}}(\bar{\boldsymbol{x}}_1), \ldots, J_{\boldsymbol{T}}(\bar{\boldsymbol{x}}_J))$ is full rank, ensuring linear independence. Differentiating the prior equality over all points $\bar{\boldsymbol{x}}_i$, we find:

$$Q = AQ^*. \quad (3.29)$$

Since Q and Q^* are invertible, it follows that A is also invertible, thereby concluding that $(\boldsymbol{f}, \boldsymbol{T}, \boldsymbol{\lambda}) \sim (\boldsymbol{f}^*, \boldsymbol{T}^*, \boldsymbol{\lambda}^*)$. □

The first step of this proof provides a blueprint for transforming the noisy model to an equivalent noiseless model in identifiability proofs for probabilistic models with additive Gaussian noise, and reducing the probabilistic identifiability problem to a deterministic one.

3.3.2 Probabilistic Simplex Component Analysis

Recall that the PRISM concerns with the following model (see Section 2.5.2):,

$$\boldsymbol{x} = A\boldsymbol{z} + \boldsymbol{v}, \quad \boldsymbol{z} \in \Delta^K, \quad \boldsymbol{v} \sim \mathcal{N}(\boldsymbol{0}, \sigma^2\mathbf{I}), \quad (3.30)$$

where $A \in \mathbb{R}^{N\times M}$ is affinely independent mixing matrix. Assuming $T \to \infty$, by the law of large numbers, the log likelihood function $\mathcal{L}_T$ as defined in (2.19) converges to

$$\mathcal{L}(A) = \int_{\mathbb{R}^N} p_{A^*}(\boldsymbol{x}) \log p_A(\boldsymbol{x})\, d\boldsymbol{x}. \quad (3.31)$$

By the Kullback-Leibler divergence, we have $\mathcal{L}(A^*) \geq \mathcal{L}(A)$, with equality if and only if

$$p_{A^*}(\boldsymbol{x}) = p_A(\boldsymbol{x}), \quad \forall \boldsymbol{x} \in \mathcal{X}, \quad (3.32)$$

suggesting that A^* is identifiable if there exists no non-trivial choice of A satisfying this equality.

Theorem 6. *Equation* (3.32) *holds if and only if* $A = A^*\Pi$*, where* Π *is a permutation matrix. Consequently,* A *is a solution to the maximization of ML in* (3.31) *if and only if* $A = A^*\Pi$.

Proof. To demonstrate the identifiability of the model parameter A, we consider the specific

case $M = N - 1$ with A being affinely independent. In this scenario, the noise-free components $\boldsymbol{x}_t$ in Equation (7) follow a uniform distribution over the simplex formed by the columns of A. This allows us to express the density function $p_A(\boldsymbol{x})$ as follows:

$$p_A(\boldsymbol{x}) = \frac{1}{\text{vol}(A)} \mathbf{1}_{\text{conv}(A)}(\boldsymbol{x}),$$

where $\mathbf{1}_{\text{conv}(A)}(\boldsymbol{x})$ is the indicator function for the convex hull of A, denoting that $p_A(\boldsymbol{x})$ is non-zero only within this convex hull. Applying this uniform simplex distribution to the observed model yields:

$$p_A(\boldsymbol{y}) = \int_{\mathbb{R}^{N-1}} \phi_\sigma(\boldsymbol{y} - \boldsymbol{x}) p_A(\boldsymbol{x})\, d\boldsymbol{x}.$$

To proceed, we leverage the Fourier transform (FT) for simplifying the convolution structure of this integral. By defining the Fourier transform $\hat{f}(\xi)$ of a function $f : \mathbb{R}^n \to \mathbb{R}$ as

$$\hat{f}(\xi) = \int_{\mathbb{R}^n} f(\boldsymbol{x}) e^{-j2\pi\xi^\top \boldsymbol{x}} d\boldsymbol{x},$$

we obtain the FT representation of $p(\boldsymbol{y}; A)$, which implies:

$$p_A(\boldsymbol{x}) = p_{A^*}(\boldsymbol{x}) \quad \forall \boldsymbol{x} \Rightarrow \hat{\phi}_\sigma(\xi)\hat{p}_A(\xi) = \hat{\phi}_\sigma(\xi)\hat{p}_{A^*}(\xi) \quad \forall \xi.$$

Since $\hat{\phi}_\sigma(\xi)$ is always non-zero for any ξ due to its Gaussian form $\hat{\phi}_\sigma(\xi) = e^{-2\pi^2\|\xi\|^2}$, we can simplify to:

$$\hat{p}_A(\xi) = \hat{p}_{A^*}(\xi) \quad \forall \xi.$$

Taking the inverse FT on both sides yields:

$$p_A(\boldsymbol{x}) = p_{A^*}(\boldsymbol{x}) \quad \forall \boldsymbol{x},$$

which implies that the convex hulls of A and A^* must coincide:

$$\text{conv}(A) = \text{conv}(A^*).$$

Thus, the vertices of the convex hulls, representing the columns of A and A^*, must be identical up to permutation, giving us

$$\{a_1, \ldots, a_N\} = \{a_{0,1}, \ldots, a_{0,N}\}.$$

This completes the intuitive proof.

For completeness, it is important to address a technical detail: when $p_A(\boldsymbol{x})$ is discontinuous, $\hat{p}_A(\xi)$ may not be integrable, and thus, its inverse FT might not exist in a strict sense. This, along with the generalization to arbitrary M, is formally resolved in the proof provided in [51]. □

Chapter 4: Methodology

4.1 Probabilistic Post-Nonlinear Simplex Component Analysis

The main objective of this study is to addresses a gap in the literature on nonlinear identifiability by introducing a probabilistic generative framework for post-nonlinear SCA. Limitations of current methods for post-nonlinear SCA are mainly due to two factors: inability to recover the latent space without additional linear SCA methods, and deterioration of performance in noisy conditions. Drawing on methodologies from [40, 52], we establish the first identifiability guarantees for nonlinear SCA in noisy settings. We show that our method guarantees simultaneous removal of nonlinear distortions and identifiability of the latent components.

We model the observed random vector $\boldsymbol{x} \in \mathbb{R}^N$ using a *noisy post-nonlinear mixture model*:

$$\boldsymbol{x} = \boldsymbol{f}(A\boldsymbol{z}) + \boldsymbol{v}, \tag{4.1}$$

where $\boldsymbol{z} \in \Delta^M$ is a latent variable representing the mixture proportions. Specifically, $\boldsymbol{z}$ follows a uniform Dirichlet distribution $\boldsymbol{z} \sim \mathcal{D}(\mathbf{1})$, enforcing a simplex constraint that each component of $\boldsymbol{z}$ is non-negative and sums to one, thus ensuring interpretability as fractional contributions. The term $\boldsymbol{v} \in \mathbb{R}^N$ represents additive Gaussian noise, modeled as $\boldsymbol{v} \sim \mathcal{N}(0, \sigma^2 I)$, where σ^2 controls the noise variance. The model parameters consist of a *mixture matrix* $A \in \mathbb{R}^{N \times M}$, assumed to be of full column rank, which maps the latent space to the observed space, and a *component-wise nonlinear function* $\boldsymbol{f}(\boldsymbol{x}) = [f_1(x_1), \ldots, f_N(x_N)]^\top$, where each $f_i : \mathbb{R} \to \mathbb{R}$ models nonlinear distortions in the observed mixture.

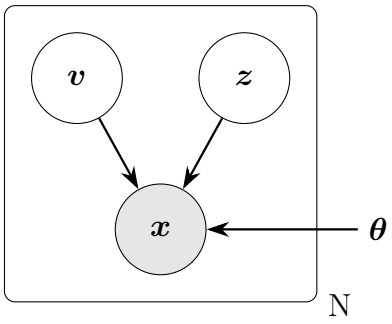

Figure 4.1: Ditected graphical model representation of a noisy Latent Variable Model (nLVM). The observed data is represented by the shaded node $\boldsymbol{x}$, the latent and noise variables by the unshaded nodes $\boldsymbol{z}$ and $\boldsymbol{v}$ respectively. Solid lines denote the generative model $p(z)p_\theta(x|z)$, with the generative model parameters θ.

The generative process in (4.1) allows us to define the probabilistic model by specifying the joint distribution over $\boldsymbol{x}$ and $\boldsymbol{z}$:

$$p_{\boldsymbol{\theta}}(\boldsymbol{x}, \boldsymbol{z}) = p_{\boldsymbol{\theta}}(\boldsymbol{x}|\boldsymbol{z})p(\boldsymbol{z}), \tag{4.2}$$

where $\boldsymbol{\theta} = (\boldsymbol{f}, A)$ denotes the parameters of the model. Here,

$$p(\boldsymbol{z}) = (M-1)!\, \mathbb{1}_{\Delta^M}(\boldsymbol{z}), \tag{4.3}$$

specifies the uniform prior distribution on the latent variable $\boldsymbol{z}$. The term

$$p_{\boldsymbol{\theta}}(\boldsymbol{x}|\boldsymbol{z}) = \phi_\sigma(\boldsymbol{x} - \boldsymbol{f}(A\boldsymbol{z})), \tag{4.4}$$

represents the conditional distribution of $\boldsymbol{x}$ given $\boldsymbol{z}$, with $\phi_\sigma(\boldsymbol{x}) = e^{-\|\boldsymbol{x}\|_2^2/2\sigma^2}/(\sqrt{2\pi}\sigma)^N$ as the Gaussian density function, centered around the nonlinear transformation $\boldsymbol{f}(A\boldsymbol{z})$.

To obtain the marginal likelihood of the observed data $\boldsymbol{x}$, we integrate out the latent variable $\boldsymbol{z}$ from the joint distribution:

$$p_{\boldsymbol{\theta}}(\boldsymbol{x}) = \int p_{\boldsymbol{\theta}}(\boldsymbol{x} \mid \boldsymbol{z}) p(\boldsymbol{z})\, d\boldsymbol{\mu}(\boldsymbol{z}). \tag{4.5}$$

This integration is performed over the simplex support of the Dirichlet prior, which is expressed as a Lebesgue integral over the unit simplex. For practical purpose, this integral can be reformulated in terms of the coordinates $\bar{\boldsymbol{z}} = [z_1, \ldots, z_{M-1}]$ as follows:

$$\int f(\boldsymbol{z})\, d\boldsymbol{\mu}(\boldsymbol{z}) = \int_{\mathbb{R}^{M-1}} f(\bar{\boldsymbol{z}}, 1 - \mathbf{1}^\top \bar{\boldsymbol{z}})\, d\bar{\boldsymbol{z}}, \tag{4.6}$$

where $z_M = 1 - \mathbf{1}^\top \bar{\boldsymbol{z}}$ ensuring that $\boldsymbol{z}$ satisfies the simplex constraint.

4.2 Identifiability Guarantees

Next, we examine the identifiability of the model (4.1). We define the equivalence relations and derive conditions under which the model parameters can be estimated from the observed data. Our primary objectives are twofold: first, to determine a transformation that effectively inverts the nonlinear distortions present in the data, and second, to accurately identify the underlying latent components.

4.2.1 Equivalence Relations

To this end, we introduce the following equivalence relation for the model in (4.1):

Definition 3. *Let $\sim$ be an equivalence relation on Θ defined as follows: $(f, A) \sim (f^*, A^*)$ if and only if there exist a matrix U and a vector $\boldsymbol{w}$ such that*

$$A^+ \boldsymbol{f}^{-1}(\boldsymbol{x}) = U A^{*+} \boldsymbol{f}^{*-1}(\boldsymbol{x}) + \boldsymbol{w}, \quad \forall \boldsymbol{x} \in \mathcal{X},$$

where U is an $n \times n$ matrix and $\boldsymbol{w}$ is a vector. If U is invertible, we denote this relation by $\sim_U$. If U is a block permutation matrix, we denote it by $\sim_P$, if U is a scaled permutation matrix, we denote it by $\sim_S$.

Building on techniques discussed in the previous chapters, we demonstrate that the model (4.5) is $\sim_U$-identifiable under mild regularity assumptions. This result is formalized in the following theorem, which constitutes the central theoretical contribution of this dissertation.

4.2.2 Nonlinearity Removal

We will first show that maximizing the likelihood 4.5 guarantees that the nonlinear transformation is removed from the model.

Theorem 7 (Nonlinearity Removal)**.** *The model* (4.5) *is* $\sim_U$*-identifiable, if the following assumptions hold:*

Assumption 7.1. *Functions* $f_1, \dots, f_N$ *are twice differentiable, and invertible.*

Assumption 7.2. *The matrix* $A \in \mathbb{R}^{N \times M}$ *has a full column rank.*

Assumption 7.3. *Dimensions of the problem satisfy* $3 \leq M \leq N \leq M(M-1)/2$.

Assumption 7.4. *The set* $\boldsymbol{x} \in \mathcal{X} | \phi_\sigma(\boldsymbol{x}) = 0$ *has measure zero, where* ϕ_σ *is defined in* (4.4).

Proof. The proof of the theorem is based on three technical lemmas provided in Section 4.3. Along the lines of the identifiability proofs in Section 3.3, we prove the theorem in three steps.

First, suppose we have two sets of parameters $(\boldsymbol{f}, \boldsymbol{A})$ and $(\boldsymbol{f}^*, A^*)$ such that their likelihoods are equal:

$$p_{\boldsymbol{f},\boldsymbol{A}}(\boldsymbol{x}) = p_{\boldsymbol{f}^*,A^*}(\boldsymbol{x}) \quad \text{for all } \boldsymbol{x} \in \mathbb{R}^N. \tag{4.7}$$

According to Lemma 2, given that Assumption 7.4 is valid, (4.7) implies equality of the noiseless distributions

$$\tilde{p}_{\boldsymbol{f},\boldsymbol{A}}(\boldsymbol{x}) = \tilde{p}_{\boldsymbol{f}^*,A^*}(\boldsymbol{x}) \quad \text{for all } \boldsymbol{x} \in \mathbb{R}^N, \tag{4.8}$$

where

$$\tilde{p}_{\boldsymbol{f},\boldsymbol{A}}(\boldsymbol{x}) = p(A^+ \boldsymbol{f}^{-1}(\boldsymbol{x})) \operatorname{vol}(\boldsymbol{A})^{-1} \operatorname{vol}(J_{\boldsymbol{f}^{-1}}(\boldsymbol{x})) \mathbb{1}_{\mathcal{X}}(\boldsymbol{x}). \tag{4.9}$$

Next, we introduce $\boldsymbol{y} = \boldsymbol{f}^{-1}(\boldsymbol{x})$. Equation (4.9) yields

$$\frac{\mathbb{1}_{\overline{\operatorname{conv}}(\boldsymbol{A})}(\boldsymbol{y})}{\operatorname{vol} A \operatorname{vol} \boldsymbol{J}_{\boldsymbol{f}}(\boldsymbol{y})} = \frac{\mathbb{1}_{\overline{\operatorname{conv}}(A^\star)}(\boldsymbol{h}(\boldsymbol{y}))}{\operatorname{vol} A^\star \operatorname{vol} \boldsymbol{J}_{\boldsymbol{f}^\star}(\boldsymbol{h}(\boldsymbol{y}))},$$

where we used the inverse function theorem. Given that the denominator is finite and non-zero, we obtain

$$\mathbb{1}_{\overline{\operatorname{conv}}(\boldsymbol{A})}(\boldsymbol{y}) = \mathbb{1}_{\overline{\operatorname{conv}}(A^\star)}(\boldsymbol{h}(\boldsymbol{y})), \quad \forall \boldsymbol{y} \in \mathbb{R}^N. \tag{4.10}$$

Using Lemma 3, we can write

$$\mathbf{1}^\top A^{\star +} \boldsymbol{h}(A\boldsymbol{z}) = 1, \quad \forall \boldsymbol{z} \in \overline{\Delta}^M. \tag{4.11}$$

Be denoting $\boldsymbol{b} = \mathbf{1}^\top A^{*+}$ we can rewrite this funcitonl equation as

$$\boldsymbol{b}\boldsymbol{h}(\boldsymbol{y}) = 1, \quad \forall \boldsymbol{y} \in \mathbb{R}^N. \tag{4.12}$$

Given that Assumptions 7.1, 7.2, and 7.3 are satisfied, according to Lemma 4, $\boldsymbol{h}$ is a linear transformation. Given that $\boldsymbol{h}$ is a component-wise linear transformation, this amounts to $\sim_S$

identifiability, i.e. the post-nonlinear distortions are identified up to rescaling. This completes the proof.

□

Theorem 7 establishes a nonlinear form of identifiability for the generative model defined by equations (4.5). Specifically, consider data generated from an original parameter set $(\boldsymbol{f}^*, A^*)$, and let $(\boldsymbol{f}, \boldsymbol{A})$ represent parameters estimated by a consistent learning algorithm in the population limit. The theorem guarantees that these estimated parameters are equivalent to the true parameters up to a linear transformation, $(\boldsymbol{f}, \boldsymbol{A}) \sim_U (\boldsymbol{f}^*, A^*)$. This result implies that, in the absence of noise, the learned transformation $\boldsymbol{f}$ would map observations to latent mixtures $\boldsymbol{y} = \boldsymbol{f}^{-1}(\boldsymbol{x})$ that are a linear transformation of the true mixtures $\boldsymbol{y}^* = \boldsymbol{f}^{*-1}(\boldsymbol{x})$. In other words, $\boldsymbol{y} = h(\boldsymbol{y}^*)$, and $\boldsymbol{h} = \boldsymbol{f}^{-1} \circ \boldsymbol{f}^*$ is a linear function.

4.2.3 Latent Variables Identification

Unlike the models in Section 3.2, where the latent variables are related to the observed data by a deterministic transformation, in the probabilistic setting, this relationship is stochastic, due to the presence of noise. As a result, the latent variables cannot be uniquely identified from the observed data, even if the model parameters are known. In this case identifiability refers to the unique recovery of the posterior distribution over the latent variables, given the observed data. We this in mind, we define the following equivalence relation:

Definition 4 (Probability Equivalence)**.** *Let $\sim$ be an equivalence relation on Θ defined as follows: $p_\theta \sim_U p_{\theta^*}$ if only if there exists a matrix U and a vector $\boldsymbol{c}$ such that*

$$p_\theta(\boldsymbol{x}) = \det U p_{\theta^*}(U^{-1}(\boldsymbol{x} - \boldsymbol{w})), \quad \forall \boldsymbol{x} \in \mathbb{R}^N, \tag{4.13}$$

where $U \in \mathbb{R}^{N \times N}$ and $\boldsymbol{w} \in \mathbb{R}^N$.

Lemma 1 (Means of Equivalent Distributions)**.** *Let $p_\theta \sim_U p_{\theta^*}$ be two equivalent distributions, then the means of the distributions are related by*

$$\mathbb{E}_{p_\theta}[\boldsymbol{x}] = U\mathbb{E}_{p_{\theta^*}}[\boldsymbol{x}] + \boldsymbol{w}.$$

Proof. The proof is trivial. Given the equivalence relation (4.13), we have

$$\begin{aligned}
\mathbb{E}_{p_\theta}[\boldsymbol{x}] &= \int \boldsymbol{x} p_\theta(\boldsymbol{x}) d\boldsymbol{x} \\
&= \int \boldsymbol{x} \det U p_{\theta^*}(U^{-1}(\boldsymbol{x} - \boldsymbol{w})) d\boldsymbol{x} \\
&= \det U \int J_{U^{-1}}(\boldsymbol{x})(U\boldsymbol{x} + w) p_{\theta^*}(\boldsymbol{x}) d\boldsymbol{x} \\
&= U\mathbb{E}_{p_{\theta^*}}[\boldsymbol{x}] + \boldsymbol{w}.
\end{aligned}$$

The proof is complete. □

Theorem 3 guarantees that the estimated posterior distribution is equivalent to the true posterior up to a linear transformation. This result is formalized in the following corollary.

Corollary 1 (Latent Identifiability)**.** *Assuming assumptions of Theorem 7, the posterior distributions over latent variables $p_\theta(\boldsymbol{z}|\boldsymbol{x})$ are $\sim_P$-identifiable.*

Proof. Given that the ELBO criterion is optimized, the KL divergence term in (2.8) vanishes, and the variational posterior matches the model posterior which is consistent with the estimated likelihood and the prior, according to the Bayes rule.

Given that the Theorem 7 is valid, the residual nonlinearity is a linear function, $\boldsymbol{h}(\boldsymbol{x}) = H\boldsymbol{x}$. Substituting $\boldsymbol{h}$ into (4.11), gives $A^{*+}HA\boldsymbol{z} \in \Delta^M$ with $\boldsymbol{z} \in \Delta^M$. Therefore, $A^{*+}HA = \Pi$ is a permutation matrix, and the true and estimated posterior distributions are $\sim_P$-equivalent.

□

By Corollary 1 and Lemma 1, the latent variables $\boldsymbol{z}$ can be identified up to permutation by sampling the posterior distribution and computing the mean.

4.3 Technical Lemmas

Here, we prove the technical lemmas that support the results discussed earlier.

Lemma 2. *Suppose $\theta = (\boldsymbol{f}, \boldsymbol{A})$ and $\theta^* = (\boldsymbol{f}^*, A^*)$ are the approximated and true parameters of the model* (4.5)*. If the marginal distributions are equal,*

$$p_\theta(\boldsymbol{x}) = p_{\theta^*}(\boldsymbol{x}), \quad \forall \boldsymbol{x} \in \mathcal{X}$$

then the noiseless distributions are equal almost everywhere:

$$\tilde{p}_\theta(\boldsymbol{x}) = p(A^+\boldsymbol{f}^{-1}(\boldsymbol{x}))\,\mathrm{vol}(\boldsymbol{A})^{-1}\,\mathrm{vol}(J_{\boldsymbol{f}^{-1}}(\boldsymbol{x}))\mathbb{1}_{\mathcal{X}}(\boldsymbol{x}), \quad \forall \boldsymbol{x} \in \mathcal{X}$$

Proof. The likelihood of model (4.5) can be written as

$$p_{\boldsymbol{\theta}}(\boldsymbol{x}) = \int_{\Delta^M} \phi_\sigma(\boldsymbol{x} - \boldsymbol{f}(A\boldsymbol{z}))p(\boldsymbol{z})d\mu(\boldsymbol{z}). \tag{4.14}$$

By leveraging (B.2) and the change of variables $\bar{\boldsymbol{x}} = \boldsymbol{f}(\boldsymbol{A}\boldsymbol{z})$, given Assumptions 7.1 and 7.2 are valid, we can rewrite the integral as

$$p_{\boldsymbol{\theta}}(\boldsymbol{x}) = \int_{\mathcal{X}} \phi_\sigma(\boldsymbol{x} - \bar{\boldsymbol{x}})p(A^+\boldsymbol{f}^{-1}(\bar{\boldsymbol{x}}))\,\mathrm{vol}(\boldsymbol{A})^{-1}\,\mathrm{vol}(J_{\boldsymbol{f}^{-1}}(\bar{\boldsymbol{x}}))d\mu(\bar{x}),$$

where $\mathcal{X}$ is the image of Δ^M under $\boldsymbol{f} \circ \boldsymbol{A}$, and A^+ is the left pseudo-inverse of $\boldsymbol{A}$. Given that $\boldsymbol{A}$ is the full-rank matrix with independent columns, $A^+ = (A^\top \boldsymbol{A})^{-1}A^\top$. Introducing the shorthand notation for the transformed distribution,

$$\tilde{p}_{\boldsymbol{f},\boldsymbol{A}}(\boldsymbol{x}) = p(A^+\boldsymbol{f}^{-1}(\boldsymbol{x}))\,\mathrm{vol}(\boldsymbol{A})^{-1}\,\mathrm{vol}(J_{\boldsymbol{f}^{-1}}(\boldsymbol{x}))\mathbb{1}_{\mathcal{X}}(\boldsymbol{x}),$$

we can write:

$$p_{\boldsymbol{\theta}}(\boldsymbol{x}) = \int_{\mathbb{R}^N} \phi_\sigma(\boldsymbol{x} - \bar{\boldsymbol{x}}) \tilde{p}_{\boldsymbol{f},\boldsymbol{A}}(\bar{\boldsymbol{x}}) d\mu(\bar{\boldsymbol{x}}). \tag{4.15}$$

By applying the Fourier Transform (FT) on both sides, and using the convolution property, we obtain:

$$\mathcal{F}[p_{\boldsymbol{\theta}}](\boldsymbol{\omega}) = \mathcal{F}[\tilde{p}_{\boldsymbol{f},\boldsymbol{A}}](\boldsymbol{\omega}) \mathcal{F}[\phi_\sigma](\boldsymbol{\omega}), \text{ for all } \boldsymbol{\omega}, \tag{4.16}$$

where $\mathcal{F}[\phi_\sigma](\boldsymbol{\omega}) = e^{-\frac{1}{2}\sigma^2 \|\boldsymbol{\omega}\|_2^2} \neq 0$ for all $\boldsymbol{\omega}$.

We have

$$p_\theta(\boldsymbol{x}) = p_{\theta^*}(\boldsymbol{x}). \tag{4.17}$$

By Assumption 7.4, $\mathcal{F}[\phi_\sigma](\omega)$ is non-zero almost everywhere, as a result equation (4.17) implies that

$$\mathcal{F}[\tilde{p}_{\boldsymbol{f},\boldsymbol{A}}](\boldsymbol{\omega}) = \mathcal{F}[\tilde{p}_{\boldsymbol{f}^*,A^*}](\boldsymbol{\omega}). \tag{4.18}$$

Given the uniqueness of the Fourier transform, according to Fact 2, it follows that:

$$\tilde{p}_{\boldsymbol{f},\boldsymbol{A}}(x) = \tilde{p}_{\boldsymbol{f}^*,A^*}(x) \tag{4.19}$$

This result indicates that the noise-free distributions must be identical if the noisy distributions are identical. □

Fact 2 (Proposition 3.8.6 [10])**.** *If two bounded Borel measures have equal Fourier transforms, then they coincide. In particular, two integrable functions with equal Fourier transforms are equal almost everywhere.*

Lemma 3. *If the following equation holds*

$$\mathbb{1}_{\overline{\text{conv}}(\boldsymbol{A})}(\boldsymbol{y}) = \mathbb{1}_{\overline{\text{conv}}(B)}(\boldsymbol{h}(\boldsymbol{y})), \quad \forall \boldsymbol{y} \in \mathbb{R}^N. \tag{4.20}$$

then

$$\mathbf{1}^\top B^+ \boldsymbol{h}(A\boldsymbol{z}) = 1, \quad \forall \boldsymbol{z} \in \overline{\Delta}^M. \tag{4.21}$$

Proof. Equation

$$\mathbb{1}_{\overline{\text{conv}}(\boldsymbol{A})}(\boldsymbol{y}) = \mathbb{1}_{\overline{\text{conv}}(B)}(\boldsymbol{h}(\boldsymbol{y})), \quad \forall \boldsymbol{y} \in \mathbb{R}^N. \tag{4.22}$$

implies that $\boldsymbol{h}(\boldsymbol{y})$ is in the convex hull of A^*, if and only if $\boldsymbol{y}$ is in the convex hull of A^*. In other words,

$$\boldsymbol{y} \in \overline{\text{conv}}(\boldsymbol{A}) \leftrightarrows \boldsymbol{h}(\boldsymbol{y}) \in \overline{\text{conv}}(B), \quad \forall \boldsymbol{y} \in \mathbb{R}^N. \tag{4.23}$$

The two sides in (4.23) are equivalent to the following conditions:

$$\boldsymbol{y} = \boldsymbol{A}\boldsymbol{z}, \quad \boldsymbol{z} \in \overline{\Delta}^M, \tag{4.24}$$

$$\boldsymbol{h}(\boldsymbol{y}) = A^* \boldsymbol{z}^*, \quad \boldsymbol{z}^* \in \overline{\Delta}^M, \tag{4.25}$$

and we can rewrite it as

$$\boldsymbol{z} \in \Delta^M \leftrightarrows B^+ \boldsymbol{h}(A\boldsymbol{z}) \in \overline{\Delta}^M. \tag{4.26}$$

As a result, we obtain a functional equation

$$\mathbf{1}^\top B^+ \boldsymbol{h}(A\boldsymbol{z}) = 1, \quad \forall \boldsymbol{z} \in \overline{\Delta}^M. \tag{4.27}$$

□

Fact 3 (Lemma 1 [48])**.** *Consider the Khatri–Rao (column-wise Kronecker) product defined as*

$$B \circledast A := [\boldsymbol{b}_1 \otimes \boldsymbol{a}_1, \cdots, \boldsymbol{b}_N \otimes \boldsymbol{a}_N],$$

where $A \in \mathbb{R}^{K\times N}$, $B \in \mathbb{R}^{L\times N}$, $\otimes$ stands for the Kronecker product, and $\boldsymbol{b}_n$, $\boldsymbol{a}_n$ are the columns of B and A. If the following condition holds:

$$\operatorname{krank}(A) + \operatorname{krank}(B) \geq N + 1,$$

then the matrix $B \circledast A$ has full column rank N.

Lemma 4. *Assume that Assumptions 7.1, 7.2, and 7.3 of the Theorem 7 are valid. Given that the following functional equation holds,*

$$\boldsymbol{b}\boldsymbol{h}(A\boldsymbol{z}) = 1, \quad \forall \boldsymbol{z} \in \overline{\Delta}^M, \tag{4.28}$$

$\boldsymbol{h}(\boldsymbol{z})$ is almost surely a linear transformation for $\boldsymbol{z} \in \overline{\Delta}^M$.

Proof. We first use $z_M = 1 - \sum_{i=1}^{M-1} z_i$ to rewrite the functional equation (4.28) as

$$\sum_{n=1}^{N} b_n \, h_n \left(\sum_{m=1}^{M-1} (a_{nm} - a_{nM}) z_m + a_{nM} \right) = 1.$$

By Fact 1, we know that for $\boldsymbol{z} \in \overline{\Delta}^M$, $\frac{\partial z_i}{\partial z_j} = 0$ for $i, j = 1, \ldots, M-1$ and $i \neq j$. Hence, differentiating with respect to z_l (where $l \in [M-1]$), and using the chain rule, this becomes:

$$\sum_{n=1}^{N} b_n (a_{nl} - a_{nM}) h_n' \left(\sum_{m=1}^{M-1} (a_{nm} - a_{nM}) z_m + a_{nM} \right) = 0, \tag{4.29}$$

Taking the second derivative with respect to z_k (for $k \in [M-1]$):

$$\sum_{n=1}^{N} b_n (a_{nl} - a_{nM})(a_{nk} - a_{nM}) h_n'' \left(\sum_{m=1}^{M-1} (a_{nm} - a_{nM}) z_m + a_{nM} \right) = 0. \tag{4.30}$$

Now, we can express the system of equations in (4.30) in matrix form:

$$b \circledast G\boldsymbol{h}'' = \mathbf{0}, \tag{4.31}$$

where $\boldsymbol{h}'' = \begin{bmatrix} h_1'' & h_2'' & \cdots & h_M'' \end{bmatrix}^\top$, and G is the matrix with dimensions $M(M-1)/2 \times N$, constructed as follows:

$$G = \begin{bmatrix} (\bar{\boldsymbol{a}}_1 \odot \bar{\boldsymbol{a}}_1)^\top \\ (\bar{\boldsymbol{a}}_2 \odot \bar{\boldsymbol{a}}_2)^\top \\ \vdots \\ (\bar{\boldsymbol{a}}_{M-1} \odot \bar{\boldsymbol{a}}_{M-1})^\top \\ (\bar{\boldsymbol{a}}_1 \odot \bar{\boldsymbol{a}}_2)^\top \\ \vdots \\ (\bar{\boldsymbol{a}}_{M-2} \odot \bar{\boldsymbol{a}}_{M-1})^\top \end{bmatrix}, \tag{4.32}$$

where each vector $\bar{\boldsymbol{a}}_i$ is defined as:

$$\bar{\boldsymbol{a}}_l = \begin{bmatrix} a_{1l} - a_{1N}, & a_{2l} - a_{2N}, & \ldots, & a_{Nl} - a_{NM} \end{bmatrix}^\top, \tag{4.33}$$

with $l = 1, \ldots, M-1$, and $\bar{A} = [\bar{\boldsymbol{a}}_1, \ldots, \bar{\boldsymbol{a}}_{M-1}]$. Here, operator $\odot$ denotes the Hadamard product, and $\circledast$ is the Khatri-Rao product.

If $b \circledast G$ has full column rank, then (4.31) yields $\boldsymbol{h}''(A\boldsymbol{z}) = 0$, which implies that $\boldsymbol{h}$ must be an affine function. To demonstrate it, we first establish that $\text{rank}(G) = N$ by identifying an $N \times N$ full-rank submatrix of G. As shown in the proof of Theorem 3 in Section 3.2.3, G almost surely has full Kruskal rank $\min(M(M-1), N)$. According to Fact 3, the Khatri-Rao product $b \circledast G$ achieves full column rank provided that $\text{krank}(b) + \text{krank}(G) \geq N + 1$. The rank of b is 1, as it is a non-zero vector, therefore $b \circledast G$ has full column rank, if $\text{krank}(G) \geq N$, i.e. $M(M-1)/2 \geq N$, which is satisfied by Assumption 7.2.

Finally, given that $\boldsymbol{h}$ is affine we can follow the arguments in the last part proof of Theorem 3 in Section 3.2.3, to conclude that $\boldsymbol{h}$ is a linear transformation.

□

4.4 Algorithm Design

In this section, we propose a learning criterion and the optimization scheme to remove the nonlinear transformations and identify the latent components in model (4.1). We consider the ML optimization design based on the VASCA model [36], i.e. we leverage a variational posterior from the LN family and optimize the ELBO loss using the VAE architecture. The posterior is parameterized by the encoder neural networks $\boldsymbol{\sigma}(\boldsymbol{x}) : \mathbb{R}^N \to \mathbb{R}^{M-1}$ and $\boldsymbol{\mu}(\boldsymbol{x}) : \mathbb{R}^N \to \mathbb{R}^{M-1}$:

$$q_{\boldsymbol{\phi}}(\boldsymbol{z}) = \frac{1}{\sqrt{2\pi}} \left(\prod_{m=1}^{M} z_m \right)^{-1} |\operatorname{diag}(\boldsymbol{\sigma}(\boldsymbol{x}))|^{-1^1/_2} \exp\left(-\frac{1}{2} \tilde{\boldsymbol{z}}^\top \operatorname{diag}(\boldsymbol{\sigma}(\boldsymbol{x}))^{-1} \tilde{\boldsymbol{z}} \right), \tag{4.34}$$

where $\tilde{z} = \log\left(\frac{\boldsymbol{z}_{-M}}{z_M}\right) - \mu(\boldsymbol{x})$ with $\boldsymbol{z}_{-M} = [z_1, \ldots, z_{M-1}]$ and $z_M = 1 - \sum_{m=1}^{M-1} z_m$, the location parameter is given by the Gaussian mean and the diagonal scale matrix is the covariance matrix. This posterior belongs to the location-scale family, generated by the additive logistic transformation $g(\cdot) = \text{softmax}([\cdot, 0])$. The difference with VASCA architecture lies in using a

nonlinear decoder. Our decoder consists of a linear transformation followed by component-wise nonlinear distortions, parameterized by neural networks $f_n(x),\ n = 1, \ldots, N$, and is optimized jointly with the encoder using *Adam* optimizer [33].

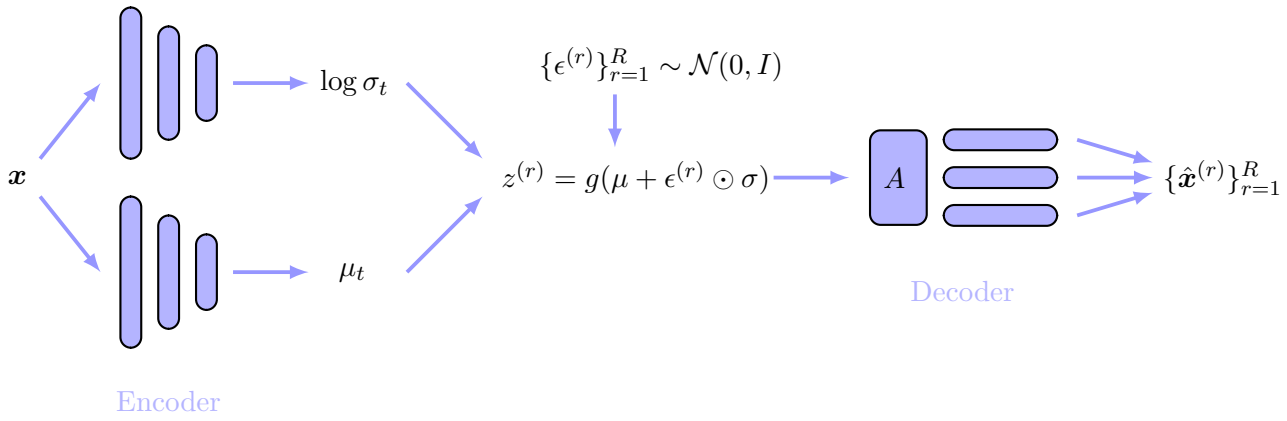

Figure 4.2: This figure illustrates the architecture of the variational autoencoder for the Post-Nonlinear Simplex Component Analysis (NISCA) model. The notation follows Figure 2.3. In this configuration, the decoder includes a linear layer A, which represents the linear mixing, followed by neural networks that apply nonlinear distortions to each component. The transformation function $g(\cdot) = \text{softmax}([\cdot, 0])$ is an additive logistic transformation, facilitating simplex-structured posterior and prior distributions.

We define the ELBO loss according to (2.11), drop the constant terms and normalize so that it has a well-behaved noiseless limit. This yields the following minimization objective:

$$\ell_{\theta,\phi}(\boldsymbol{x}) = \sum_{r=1}^{R} \ell_\theta^{rec}(\boldsymbol{x}, \boldsymbol{z}^{(r)}) + \sigma_v^2 h_\phi(\boldsymbol{z}^{(r)}), \tag{4.35}$$

where $\boldsymbol{z}^{(r)}$ is sampled according to

$$\boldsymbol{z}^{(r)} = g(\boldsymbol{\mu}(\boldsymbol{x}) + \boldsymbol{\sigma}(\boldsymbol{x}) \odot \boldsymbol{\epsilon}^{(r)}), \quad \boldsymbol{\epsilon}^{(r)} \sim \mathcal{N}(0, \boldsymbol{I}).$$

The reconstruction loss and the pointwise entropy terms are defined as:

$$\ell_{\boldsymbol{f},A}^{rec}(\boldsymbol{x}, \boldsymbol{z}) = \|\boldsymbol{x} - \boldsymbol{f}(A\boldsymbol{z})\|^2, \tag{4.36}$$

$$h_\phi(\boldsymbol{z}) = \tilde{z}^\top \text{diag}(\boldsymbol{\sigma}(\boldsymbol{x}))^{-1} \tilde{z} + \sum_{i=1}^{M-1} \log \sigma_i(\boldsymbol{x}) + 2 \sum_{i=1}^{M} \log z_i. \tag{4.37}$$

This loss function corresponds with the architecture as shown in Figure 4.2, and is optimized by Algorithm 1.

In contrast to deterministic nonlinear approaches, our loss function incorporates an additional entropy term, which acts as a regularizer. This term encourages the approximate posterior distribution to remain close to the prior, thereby preventing overfitting and prevent degenerate solutions, also known as the posterior collapse. By imposing this regularization, we ensure that the estimated posterior properly fills the latent space and captures an accurate approximation

of the true posterior distribution.

Algorithm 1 Post-Nonlinear Simplex Component Analysis (NISCA).

Input: Data tensor X, latent dimension M, noise variance σ_v^2
Initialize: Parameters A, $\boldsymbol{f}$, $\boldsymbol{\mu}$, and $\boldsymbol{\sigma}$
for epoch $= 1, \ldots,$ num_epochs **do**
    for $i = 1, \ldots,$ num_batches **do**
        $X_i \leftarrow$ i-th minibatch of X
        Draw R samples $\{\boldsymbol{\epsilon}^{(r)}\}_{r=1}^{R}$ from $\mathcal{N}(0, \boldsymbol{I})$
        Compute reconstructed samples $\hat{X}_i^{(r)} = \boldsymbol{f}(A\boldsymbol{z}^{(r)})$
        Calculate loss $\frac{1}{R}\sum_{r=1}^{R} \ell_{A,\boldsymbol{f},\boldsymbol{\mu},\boldsymbol{\sigma}}^{(r)}(X_i, \hat{X}_i^{(r)})$
        Update $A, \boldsymbol{f}, \boldsymbol{\mu}, \boldsymbol{\sigma}$ using *Adam* optimizer
    end for
end for
Output: Optimized parameters $A, \boldsymbol{f}, \boldsymbol{\mu}, \boldsymbol{\sigma}$

Chapter 5: Experiments

In this chapter, we evaluate the performance of the proposed model in numerical experiments. We leverage synthetic data to systematically validate the model's theoretical guarantees. Synthetic data enables controlled data generation, allowing us to specify the form of nonlinear distortions and noise levels. Moreover, having ground truth available supports precise quantitative evaluation of the estimated model parameters and latent components. Knowing the ground truth also directly reveals essential model attribute, such as the latent space dimension. In real-world applications, however, the latent dimension is typically determined through cross-validation.

The experiments are implemented in Python using the PyTorch Lightning framework on a MacBook Pro with an Apple M3 Pro chip, 12 cores at 4.05 GHz, and 18 GB of RAM. The source code is publicly available at `https://github.com/paukvlad/nisca`.

5.1 Experiment Design

5.1.1 Data Generation

The data is generated according to the model specified in (4.1). The mixing matrix A^* is sampled from a normal unit distribution, scaled by factor 10 for better visualization. Sampling from the normal distribution ensures linear independence of the columns. This is because the set of matrices with linearly dependent columns forms a lower-dimensional subspace within the space of all matrices, which has measure zero; thus, a randomly drawn matrix from a continuous distribution will lie outside this subspace with probability 1. The nonlinear functions f_n are chosen as variants of $\exp(\cdot)$, $\text{sigmoid}(\cdot)$, and $\tanh(\cdot)$ to ensure invertibility and reduce computational load. Latent components $\boldsymbol{z}^{(t)}$ are drawn from a uniform Dirichlet distribution $\mathcal{D}(\mathbf{1})$ and combined using the matrix A^*, followed by the nonlinear functions f_n, to produce noiseless observations $\boldsymbol{x}^{(t)} = \boldsymbol{f}(A\boldsymbol{z}^{(t)})$.

Gaussian noise is then added to the generated data to achieve the desired signal-to-noise ratio (SNR). The variance of the noise is set according to

$$\sigma_v^2 = \frac{1}{\text{SNR}} \sum_{t=1}^{T} \|\boldsymbol{x}^{(t)}\|_2^2,$$

where $\boldsymbol{v}^{(t)}$ is the noise vector for the t-th sample. This formulation maintains a noise level that scales with the total signal power, ensuring that the SNR remains consistent across all generated data samples. SNR is commonly expressed in decibels (dB), calculated as

$$\text{SNR}_{\text{dB}} = 10 \log_{10}(\text{SNR}) \, \text{dB}.$$

For simplicity and ease of visualization, we restrict the model to two independent degrees of freedom, setting both the latent space dimension and the observed space dimension to $N = M = 3$. An example sample of the data vectors $\boldsymbol{x}$ is shown in Figure 5.1.

Figure 5.1: A sample of the generated data vectors $\boldsymbol{x}$.

5.1.2 Algorithm Settings

Unless otherwise specified the algorithm settings for our model are as follows and are kept consistent across all experiments. The decoder architecture employs an independent fully-connected neural network (FCN) for each observed component. For our simulation settings, we find that a single-hidden-layer network with 128 neurons and ReLU activation provides an effective balance between performance and computational efficiency. While increasing network depth enhances both performance and convergence speed, it also raises computational complexity. By contrast, increasing network width does not yield significant improvements. For more complex real-world experiments, network architecture design can benefit from practices such as cross-validation and other deep learning optimization techniques. Additionally, we include a single linear layer with $N = 3$ neurons, matching the target space dimension, and an input dimension of $M = 3$, without any activation function, to model the linear combination of latent components that precedes the nonlinear transformations. We use the same decoder architecture throughout the experiments. The encoder is implemented as a fully-connected neural network with input dimension $N = 3$ matching the observed space dimension, and output dimension $M = 3$. Similarly to the decoder, a single-layer network with 128 neurons and ReLU activation is sufficient for our simulation settings.

To optimize parameters, we use the Adam optimizer [33] with a learning rate of 10^{-3} for the decoder and 10^{-2} for the encoder. The optimizer processes mini-batches of 100 data points, randomly sampled from the dataset, to estimate gradients for updating model parameters. Training runs for up to 5000 epochs, and stops if the latent MSE improvement between checks falls below 10^{-4} for 100 epochs.

5.1.3 Baselines and Benchmarks

VASCA serves as our baseline, representing a linear version of our model. We use identical to our model encoder architecture, and the decoder is represented by a single linear layer without activation to model the linear mixture of latent components. The model is optimized using Adam with the same learning rates as our model, and the batch size of 1000 samples.

We use the constrained neural autoencoder (CNAE) [40] method as our main benchmark for the nonlinear experiments. This method relies on the same nonlinear mixture model as the proposed method in the noiseless limit. The CNAE model, which only addresses nonlinear distortion removal, relies on subsequent application of the MVES algorithm to unmix latent components after nonlinearity cancellation. The generated latent components are sufficiently scattered in Δ^M according to the definition in [18], thus are provably identifiable up to permutation ambiguities by MVES given that the nonlinearity is removed. CNAE only considers the component-wise transformation and has the equal input and output dimensions. The decoder in this case is similar to our model, with the exception that the linear layer is omitted. The encoder

has the same architecture as the decoder. CNAE is trained using the augmented Lagrangian optimization. We run up to 100 epochs of Adam for solving the primal subproblem with the 'importance' of the constraint $\rho = 10^2$, with the learning rate set to 10^{-3} for both the encoder and decoder, and the batch size set to 100 samples.

5.1.4 Metrics

The model is evaluated by leveraging three primary metrics. Nonlinearity removal is assessed through *coefficient of determination*, denoted by R^2, also referred to as the R-square metric. It provides a measure of how well observed outcomes are replicated by the model, based on the proportion of total variation of outcomes explained by the model. Applied to residual nonlinearity $\boldsymbol{h} = \boldsymbol{f}^{*-1} \circ \boldsymbol{f}$, it quantifies the alignment between the residual nonlinearity and its linear approximation, and provides a direct measure of the model's effectiveness in compensating for nonlinear distortions. Specifically, we use the R^2 metric between the residual nonlinearity and its linear fit $\tilde{\boldsymbol{h}} = C\boldsymbol{y}$, over the estimated linear mixtures $\boldsymbol{y}^{(t)} = A\boldsymbol{z}^{(t)}$ for $t = 1, \ldots, T$:

$$R^2 = 1 - \frac{\sum_{t=1}^{T} \|\tilde{h}^{(t)} - h^{(t)}\|_2^2}{\sum_{t=1}^{T} \|\tilde{h}^{(t)} - \bar{h}\|_2^2},$$

where $\bar{h} = \frac{1}{T}\sum_{i=1}^{T} \tilde{h}^{(t)}$. If the residual nonlinearity is perfectly linear, $R^2 = 1$.

Another metric, that provides a measure of the model's ability to remove nonlinearity, is the *subspace distance* (SD) between the true and estimated latent spaces. SD measures the nonlinear distortions in the latent space, providing a quantitative measure of the model's ability to remove nonlinearity under noisy conditions. According to Theorem 7,

$$H = \boldsymbol{h}(AZ) \approx DAZ,$$

where $Z = [\boldsymbol{z}^{(1)}, \ldots, \boldsymbol{z}^{(T)}]$, is expected under the proposed learning criterion, where D is a full rank diagonal matrix. Hence, we adopt the subspace distance $\text{dist}(Z, \hat{Z}) = \|P_{\boldsymbol{z}}^{\perp} Q_{\boldsymbol{h}}\|_2$ with $Z = \text{range}(Z^\top)$ and $\hat{Z} = \text{range}(H^\top)$ as the performance metric, where $Q_{\boldsymbol{h}}$ is the orthogonal basis of $\text{range}(H^\top)$ and $P_{\boldsymbol{z}}^{\perp} = I - Z(Z^\top Z)^{-1}Z^\top$ is the orthogonal projector onto the complement of $\text{range}(Z^\top)$. This performance metric, which is bounded within the interval $[0, 1]$, serves as an indicator of how well the subspace $\hat{Z}$ approximates the subspace Z. A value of 0 indicates perfect alignment between the subspaces, representing the best possible outcome, while larger values suggest greater deviations and, consequently, less effective nonlinearity removal.

The accuracy of latent space recovery is quantified by the mean square error (MSE) between the true and estimated latent variables, particularly under noisy and nonlinear settings. This metric assesses the fidelity of the recovered latent space, indicating how closely the model can approximate the true latent variables despite the presence of noise. The latent space identification

is assessed by the *mean square error* (MSE) between the true and estimated latent components,

$$\mathrm{MSE} = \min_{\boldsymbol{\pi} \in \Pi_M} \frac{1}{M} \sum_{m=1}^{M} \left\| \boldsymbol{z}^*_{m,:} - \boldsymbol{z}_{\pi_m,:} \right\|_2^2, \tag{5.1}$$

where Π_M is the set of all permutations of $\{1, \ldots, M\}$, $\boldsymbol{z}^*_{m,:}$ and $\boldsymbol{z}_{m,:}$ are the ground truth and estimated m-th row of Z, respectively, and the m-th row in Z represents the m-th latent component. The permutation matrix reflects an intrinsic row permutation ambiguity in the estimated latent components that cannot be removed without additional prior knowledge.

5.2 Results

5.2.1 Linear Mixture

The experimental setup and algorithm details are as described in Section 5.1.2 and Section 5.1.1, except that the data generation does not include nonlinear distortions, and the batch size is 1000 for both VASCA and our model. First, we confirm that our model correctly identifies nonlinear distortions as linear functions, as shown in Figure 5.2. This result is expected, but nontrivial, as the model is randomly initialized and the decoder neural networks are not linear at the beginning of training. Figure 5.3 demonstrates correlation between the true and the estimated

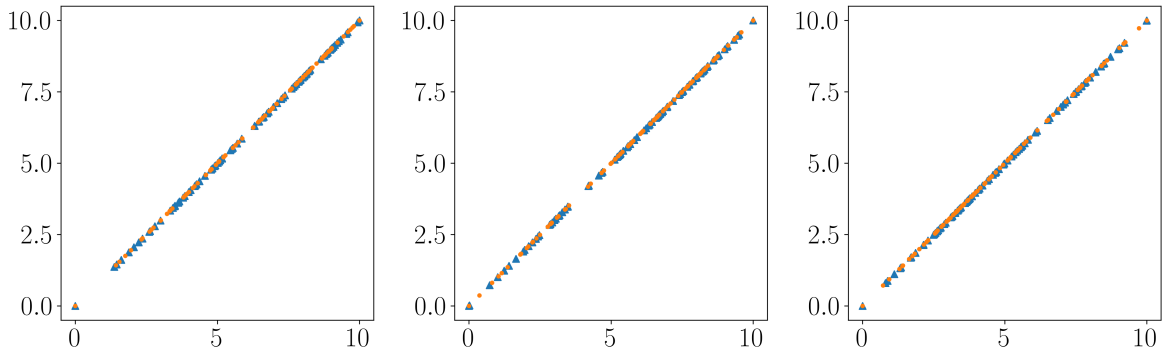

Figure 5.2: Nonlinearity removal in the linear case. Each plot shows the true (orange dots) and estimated (blue triangles) nonlinearities for each of the observed components (rescaled for better visualization).

latent components. The model correctly identifies the latent space and unmixes the latent components, providing a noisy approximation of the true latent variables, represented by the points scattered along the identity line.

Table 5.1 and Table 5.2 show the MSE and SD for different noise levels, averaged over 5 random trials. The proposed method shows comparable performance to VASCA benchmark in terms of both metrics, and is correctly recovering the latent components. In contrast to VASCA, our model does not identify the mixture matrix. This is because of the ambiguity introduced by the learned linear post-mixture transformation in the decoder, which is identity in the VASCA model.

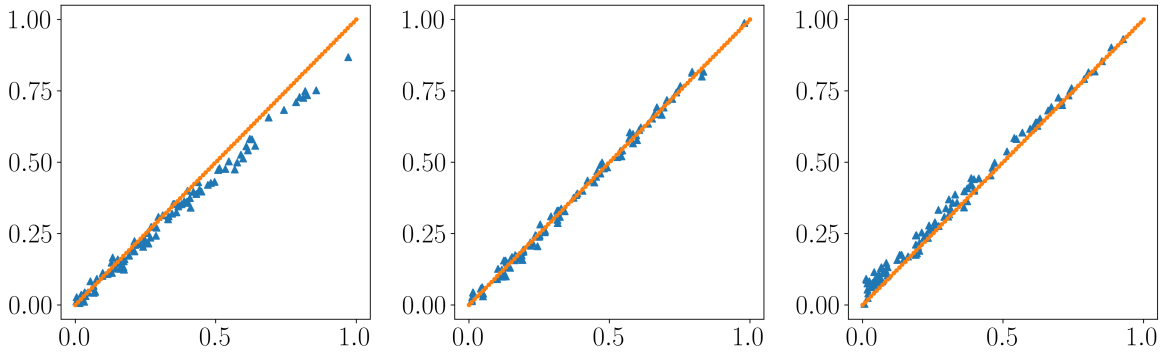

Figure 5.3: Latent component identification in the linear case with 30 dB SNR. Each plot shows correlation between the true and estimated latent component in a random sample of training data, with the true components residing on the x-axis and the estimated components on the y-axis. The orange line represents identity.

Table 5.1: MSE of the estimated latent components under various SNR in linear mixture.

Model	**10dB**	**20dB**	**30dB**
Proposed	$2.82 \cdot 10^{-2}$	$5.43 \cdot 10^{-3}$	$5.63 \cdot 10^{-3}$
VASCA	$2.93 \cdot 10^{-2}$	$5.47 \cdot 10^{-3}$	$5.03 \cdot 10^{-3}$

5.2.2 Nonlinearity Removal

Next, we assess the model's ability to identify nonlinear distortions. The noise level is set to 30dB as before. The nonlinear distortions applied to each dimension are:

$$
\begin{aligned}
f_1(x_1) &= 5\,\mathrm{sigmoid}(x_1) + 0.3x_1, \\
f_2(x_2) &= 3\tanh(x_2)0.2x_2, \\
f_3(x_3) &= 0.4\exp(x_3).
\end{aligned}
$$

It is important to note that these functions are not disclosed to the learning algorithm and are solely used for visualization purposes after the learning process is completed. We run trainer for up to 5000 epochs, and use the same stopping criterion as in the linear case.

Figure 5.4 shows the true and estimated nonlinearities, scaled to fill the same range for better visualization. As we can see, nonlinearities learned by the proposed method visually align with the true distortions. The R^2 values are 0.9931, 0.9898, and 0.9912 for the first, second, and third components, respectively.

5.2.3 Latent Space Identification

Figure 5.5 shows correlation between the true and estimated latent components, corresponding with the estimated nonlinearities in Figure 5.4. The estimates are visually well aligned with the ground truth, demonstrating that our model correctly recovers the latent space and achieves unmixing in the presence of noise and nonlinear distortions.

Table 5.2: SD between the estimated latent components under various SNR in linear mixture.

Model	**10dB**	**20dB**	**30dB**
Proposed	0.331	0.130	0.132
VASCA	0.357	0.131	0.128

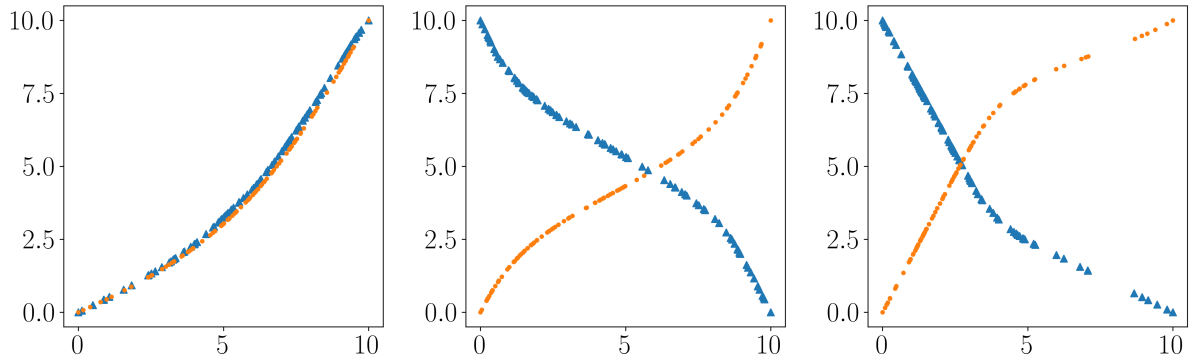

Figure 5.4: Nonlinearity removal in noisy post-nonlinear mixture data. Each plot shows the true (orange dots) and estimated (blue traingles) nonlinearities for each of the observed components (rescaled for better visualization).

For the given experimental settings, our model yields the latent MSE of $5.23 \cdot 10^{-3}$, with SD 0.131. Compared to the linear case, the values are higher, which is expected due to the presence of nonlinear distortions. Nevertheless, the model is able to unmix the latent components to a degree comparable to CNAE/MVES pipeline, which yields the SD of 0.123 for similar experimental settings. This is analogous to around only 2 degrees of misalignment between two vectors.

5.2.4 Impact of Noise

Finally, we systematically evaluate the impact of noise and compare the performance of our model with the benchmark CNAE/MVES pipeline and the baseline, under different SNR values. Settings are the same as before. The SNR is set to 10, 20, and 30 dB, and the results are averaged over 5 random trials.

From Table 5.3, one can see that the nonlinear models by far outperform the baseline, especially at higher SNR levels. At lower SNR levels, the performance gap is smaller, as the

Table 5.3: R^2 of the estimated latent components under various SNR in nonlinear mixture.

Model	**10dB**	**20dB**	**30dB**
VASCA	0.671	0.561	0.715
CNAE/MVES	0.772	0.945	0.991
Proposed	0.897	0.971	0.994

noise dominates the signal, especially for the latent space metrics MSE and SD in Tables 5.4, and 5.5. At the same time, our model shows superior performance in removing nonlinear distor-

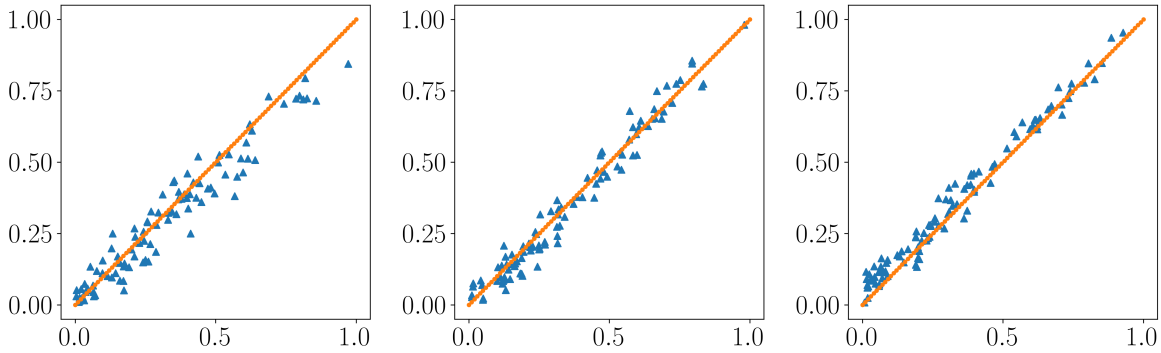

Figure 5.5: Latent component identification in the post-nonlinear mixture model with 30 dB SNR. Each plot shows correlation between the true and estimated latent component in a random sample of training data, with the true components residing on the x-axis and the estimated components on the y-axis. The orange line represents identity.

Table 5.4: MSE of the estimated latent components under various SNR in nonlinear mixture.

Model	**10dB**	**20dB**	**30dB**
VASCA	$7.34 \cdot 10^{-2}$	$6.81 \cdot 10^{-2}$	$4.87 \cdot 10^{-2}$
CNAE/MVES	$6.85 \cdot 10^{-2}$	$2.07 \cdot 10^{-2}$	$5.01 \cdot 10^{-3}$
Proposed	$6.71 \cdot 10^{-2}$	$2.21 \cdot 10^{-2}$	$5.23 \cdot 10^{-3}$

tions in highly noisy conditions, as evidenced by higher R^2 values compared to the benchmark CNAE/MVES pipeline. The benchmark method exhibits a decline in nonlinearity removal effectiveness as noise levels increase, whereas our model maintains robust performance across all SNR levels. We can also see that the efficiency of the latent space identification is similar to

Table 5.5: SD of the estimated latent components under various SNR in nonlinear mixture.

Model	**10dB**	**20dB**	**30dB**
VASCA	0.562	0.611	0.607
CNAE/MVES	0.518	0.226	0.123
Proposed	0.493	0.272	0.131

the linear case. This is expected when the nonlinear distortions are correctly identified, as the latent space is then estimated from the linear mixtures.

Chapter 6: Conclusion

6.1 Implications of Results

In this work, we revisited the simplex component analysis (SCA) from a model identification perspective and introduced NISCA, the first identifiable probabilistic post-nonlinear simplex component analysis model. NISCA provides a rigorous framework for unsupervised learning in noisy post-nonlinear mixtures, particularly when the underlying factors exhibit compositional structure. The theoretical guarantees established by Theorem 7 and Corollary 1 support the model's ability to recover the true generative process under mild regularity conditions. By leveraging the geometric properties of the probability simplex, we demonstrated that latent component identifiability is achievable even in the presence of nonlinear distortions and noise, thus enhancing the interpretability and practical utility of SCA in real-world applications.

Our method has direct implications for a wide range of practical tasks, including audio and speech processing, hyperspectral imaging, topic modeling, and image classification. Given that in such applications, both noise and nonlinear distortions are common, our model can significantly enhance the quality of the learned representations and the accuracy of the downstream tasks. This has been demonstrated in synthetic data, where it was shown that our model outperforms existing methods, particularly in highly noisy environments.

As a generative model, NISCA also offers a principled approach to data generation, enabling the synthesis of new data points by sampling from the learned latent space. This generative capability can be useful in applications where controlled data augmentation or simulation is required, such as in engineering, financial and scientific tasks.

6.2 Comparative Analysis

NISCA's probabilistic formulation offers a distinct advantage over existing deterministic post-nonlinear SCA models, which often require additional data or post-processing for identifiability. Furthermore, employing gradient-based optimization framed as variational inference, our approach avoids constrained optimization needed in deterministic methods, which is often unstable and computationally expensive.

The empirical evaluation of our proposed model, NISCA, was conducted on a synthetic dataset to systematically assess the model's identifiability and performance under controlled conditions. Benchmarked against VASCA [36] (a probabilistic linear baseline) and CNAE/MVES [40] (a deterministic nonlinear benchmark), the model's performance was measured using R-squared (R^2) and subspace distance (SD) for nonlinearity removal, as well as mean squared error (MSE) for latent space recovery, showcasing its robust nonlinearity handling and latent space estimation capabilities.

We found that our model is more robust to noise and provides a more reliable nonlinearity removal algorithm. Specifically, at low SNR values, NISCA outperforms the main benchmark, the constrained nonlinear autoencoder (CNAE), in terms of the R^2 metric, which quantifies how well the composition of the learned and true inverse nonlinearities can be approximated by a straight line, as shown in Table 5.3. The most notable improvement is observed at an SNR of 10 dB, where NISCA achieves an R^2 score of 0.90 compared to 0.77 for CNAE. Our model

demonstrates similar performance to the benchmark in terms of latent component recovery when compared to identifying the latent components using linear SCA methods, particularly MVES, following nonlinearity removal with CNAE, as shown by the MSE and SD metrics in Tables 5.4 and 5.5.

6.3 Limitations and Challenges

While the findings highlight the utility of probabilistic models in estimation tasks, this approach has notable limitations. First, our model incurs significant computational cost when the data feature dimension N is large, as each dimension requires an individual neural network. This poses a challenge for scaling the model to high-dimensional datasets without further optimization. Additionally, latent space sampling, a key aspect of the VAE framework, further contributes to the computational load. In our simulations, we didn't exploit this aspect of the model, mainly due to computational constraints. As a result, our model underperformed in the low noise regime, where the latent space sampling could have been beneficial.

Second, our approach is tailored to the specific generative model underlying the data. In cases where nonlinear distortions are more complex than component-wise nonlinearities, this framework does not guarantee effective nonlinearity removal. Furthermore, the identifiability results are based on the assumption of Gaussian noise and Dirichlet distributed prior, which may not hold true in all practical scenarios.

Third, the learning criterion for NISCA is a nonconvex optimization problem. The proposed algorithm does not guarantee solving the formulated learning criterion, creating a gap between the theoretically identifiable results (assuming optimal optimization) and the results attainable in practice. Consequently, the optimization process can become trapped in local minima, resulting in partial recovery of the true latent components, with some components remaining mixed and corresponding nonlinearities diverging from the true generative process. Achieving a solution close to the true generative process often requires repeated random initializations, which is computationally intensive and necessitates manual intervention to select the best solutions.

6.4 Future Directions

Future research could extend the application of simplex-structured models by incorporating more general geometric constraints on the prior, such as null space constraints, as suggested in [41] for the deterministic post-nonlinear SCA model. Another practical direction would be to investigate the impact of latent space dimensionality on model performance. Since the dimensionality is often unknown in real-world applications, it is important to assess how the model behaves under varying latent space dimensions and whether dimensionality can be inferred from the data using techniques like Bayesian model selection and cross-validation.

Moreover, systematically exploring network architectures and optimization techniques, such as normalization layers, convolutional networks, and adaptive learning rate scheduling, among other, would be valuable for extending the method to practical scenarios where nonlinearities

are by far more complex than those in synthetic data. Preliminary experiments suggest that our model may perform better with limited or corrupted data, highlighting a potential advantage of density estimation over point-wise estimation. Therefore, a finite-sample analysis, especially in the low SNR regime, could offer a structured assessment of the model's performance in real-world scenarios.

Finally, the evaluation of the model on real data was beyond the scope of this study due to computational resource limitations. Addressing this, specifically in the context of the nonlinear hyperspectral unmixing, along with the aforementioned research directions, will be key objectives for future journal publications following this work.

APPENDICES

Appendix A: Dirichlet Distribution

Definition 5. *The $\boldsymbol{\alpha}$-Dirichlet distributed random variable $\boldsymbol{z} \in \Delta_N$ the density of $\overline{\boldsymbol{z}} = (z_1, \ldots, z_{N-1})$ is:*

$$\mathcal{D}(\overline{\boldsymbol{z}}; \boldsymbol{\alpha}) = \frac{1}{B(\boldsymbol{\alpha})} \left(\prod_{i=1}^{N-1} z_i^{\alpha_i - 1} \right) \left(1 - \sum_{i=1}^{N-1} z_i \right)^{\alpha_N - 1} \mathbb{1}_{\tilde{\Delta}_{N-1}}(\overline{\boldsymbol{z}}), \tag{A.1}$$

where $\boldsymbol{\alpha} \in \mathbb{R}^N_{++}$ is the concentration parameter, $\tilde{\Delta}$ is the distribution support,

$$\tilde{\Delta}_{N-1} = \{\overline{\boldsymbol{z}} \in \mathbb{R}^{N-1}_{++} | 1 - \mathbf{1}^\top \overline{\boldsymbol{z}} > 0\}, \tag{A.2}$$

the multivariate beta function $B(\boldsymbol{\alpha})$ is given by

$$B(\boldsymbol{\alpha}) = \frac{\prod_{i=1}^N \Gamma(\alpha_i)}{\Gamma(\sum_{i=1}^N \alpha_i)}$$

and $\Gamma(x) = \int_0^\infty t^{x-1} e^{-t}\, dt$ is the Gamma function.

It is commonly defined with respect to the Lebesgue measure on the Euclidean space $\mathbb{R}^{N-1}$,

$$\mathcal{D}(\boldsymbol{z}; \boldsymbol{\alpha}) = \frac{1}{B(\boldsymbol{\alpha})} \left(\prod_{i=1}^{N} z_i^{\alpha_i - 1} \right) \mathbb{1}_{\overline{\Delta}}(\boldsymbol{z})$$

where $\boldsymbol{z}$ belongs to the standard $(N-1)$-simplex (**??**), and write $\boldsymbol{z} \sim \mathcal{D}(\cdot; \boldsymbol{\alpha})$ to specify a Dirichlet random variable. When $\boldsymbol{\alpha} = \mathbf{1}$, we obtain the uniform unit-simplex distribution:

$$\mathcal{D}(\boldsymbol{z}; \mathbf{1}) = (N-1)! \cdot \mathbb{1}_{\overline{\Delta}}(\boldsymbol{z}). \tag{A.3}$$

Fact 4 (Dirichlet moments, see [43])**.** *Let $\boldsymbol{z} \sim \mathcal{D}(\cdot; \boldsymbol{\alpha})$.*

(a) The expectation of $\boldsymbol{z}$ is given by

$$\mathbb{E}[\boldsymbol{z}] = \tilde{\boldsymbol{\alpha}}, \tag{A.4}$$

where $\tilde{\boldsymbol{\alpha}} = \boldsymbol{\alpha}/\alpha_0$ and $\alpha_0 = \sum_{i=1}^N \alpha_i$.

(b) The covariance of $\boldsymbol{z}$ is

$$Cov(\boldsymbol{z}) = \frac{1}{1 + \alpha_0} \left(diag(\tilde{\boldsymbol{\alpha}}) - \tilde{\boldsymbol{\alpha}} \tilde{\boldsymbol{\alpha}}^\top \right) \tag{A.5}$$

(c) The entropy of $\boldsymbol{z}$ is

$$H(\boldsymbol{z}) \coloneqq \mathbb{E}[-\log \mathcal{D}(\boldsymbol{z}; \boldsymbol{\alpha})] = \log B(\boldsymbol{\alpha}) - \sum_{i=1}^N (\alpha_i - 1)(\psi(\alpha_i) - \psi(\alpha_0)) \tag{A.6}$$

where $\psi(x) = \log \Gamma(x)'$ is the digamma function.

Fact 5 (uniform distribution on a full-dimensional simplex)**.** *Let* $\boldsymbol{x} = \boldsymbol{B}\boldsymbol{z}$*, where* $\boldsymbol{B} \in \mathbb{R}^{(N-1)\times N}$ *is affinely independent and* $\boldsymbol{z} \sim \mathcal{D}(\cdot, \mathbf{1})$*. The PDF of* $\boldsymbol{x}$ *is given by:*

$$p(\boldsymbol{x}) = \frac{1}{\operatorname{vol} \boldsymbol{B}} \mathbb{1}_{\overline{\operatorname{conv}}(\boldsymbol{B})}(\boldsymbol{x}). \tag{A.7}$$

Proof. By $z_N = 1 - \mathbf{1}^\top \bar{\boldsymbol{z}}$, we can write,

$$\boldsymbol{x} = \sum_{i=1}^{N-1} \boldsymbol{b}_i z_i + \boldsymbol{b}_N \left(1 - \sum_{i=1}^{N-1} z_i\right) = \bar{\boldsymbol{B}}\bar{\boldsymbol{z}} + \boldsymbol{b}_N$$

where $\overline{\boldsymbol{B}}$ is invertible due to the affine independence of $\boldsymbol{B}$. Since the mapping from $\bar{\boldsymbol{z}}$ to $\boldsymbol{x}$ is bijective, we can apply transformation of random variables to obtain

$$p(\boldsymbol{x}) = \frac{1}{|\det \bar{\boldsymbol{B}}|} \mathcal{D}(\bar{\boldsymbol{B}}^{-1}(\boldsymbol{x} - \boldsymbol{b}_N); \mathbf{1}),$$

where D is defined in the Dirichlet distribution in (A.1). It can be verified that

$$\bar{\boldsymbol{B}}^{-1}(\boldsymbol{x} - \boldsymbol{b}_N) \in \tilde{\Delta} \Leftrightarrow \boldsymbol{x} \in \overline{\operatorname{conv}}(\boldsymbol{B}),$$

and hence

$$\mathcal{D}(\bar{\boldsymbol{B}}^{-1}(\boldsymbol{x} - \boldsymbol{b}_N); \mathbf{1}) = (N-1)!\, \mathbb{1}_{\overline{\operatorname{conv}}(\boldsymbol{B})}(\boldsymbol{x}).$$

Finally, from (**??**), we have $\operatorname{vol} \boldsymbol{B} = |\det \bar{\boldsymbol{B}}|/(N-1)!$ when $\bar{\boldsymbol{B}}$ is square.

□

Appendix B: Change of Variables

Definition 6 (The Lebesgue integral)**.** *Let $f : \mathbb{R}^N \to \mathbb{R}$ be a measurable function. The Lebesgue integral of f over a set $\mathcal{X} \subseteq \mathbb{R}^N$ is defined as*

$$\int_{\mathcal{X}} f(\boldsymbol{x})\, d\mu(\boldsymbol{x}) = \int_{\mathbb{R}^N} \mathbb{1}_E(\boldsymbol{x}) f(\boldsymbol{x})\, d\mu(\boldsymbol{x}), \tag{B.1}$$

where $\mathbb{1}_{\mathcal{X}}(\boldsymbol{x})$ is the indicator function of $\mathcal{X}$, and μ is the Lebesgue measure on $\mathbb{R}^N$.

The integral over the simplex Δ^N is given by

$$\int_{\Delta^N} f(\boldsymbol{z})\, d\mu(\boldsymbol{z}) = \int_{\mathbb{R}_+^{N-1}} f(\bar{\boldsymbol{z}}, 1 - \mathbf{1}^\top \bar{\boldsymbol{z}})\, d\bar{\boldsymbol{z}},$$

where we use the Lebesgue integral for a compact and symmetric representation.

The change of variables formula allows to transforming an integral over one set $\mathcal{Z} \subset \mathbb{R}^M$ into an integral over another set $\mathcal{X} \subset \mathbb{R}^N$ $(N > M)$ via a sufficiently well-behaved function $\phi : \mathcal{X} \to \mathcal{Z}$:

$$\int_{\mathcal{Z}} f(\boldsymbol{z})\, d\boldsymbol{z} = \int_{\mathcal{X}} (f \circ \phi)(\boldsymbol{x}) \operatorname{vol}(J_\phi(\boldsymbol{x}))\, d\boldsymbol{x}, \tag{B.2}$$

where $\operatorname{vol}(\cdot)$ denotes the matrix volume, J_ϕ is the full column rank Jacobian matrix of ϕ over $\mathcal{X}$,

$$J_\phi = \left(\frac{\partial \phi_i}{\partial x_j} \right),$$

and the integration measures $d\boldsymbol{z}$ and $d\boldsymbol{x}$ are defined with respect to the Lebesgue measure on $\mathcal{Z}$ and $\mathcal{X}$, respectively. The volume of a matrix is defined as the square root of the sum of the squares of the determinants of all possible maximal square submatrices (submatrices of full rank)

$$\operatorname{vol}(A) = \sqrt{\sum_{(I,J) \in \mathcal{N}(A)} (\det A_{IJ})^2}, \tag{B.3}$$

where the sum is taken over the index set $\mathcal{N}(A)$ of all rr nonsingular submatrices A_{IJ}. The matrix volume is the generalization of the absolute value of determinant from nonsingular to arbitrary matrices. Alternatively, it can be viewed as the product of the singular values of the matrix. For a full column rank matrix, this volume simplifies to:

$$\operatorname{vol}(A) = \sqrt{\det(A^T A)}, \tag{B.4}$$

and for a square matrix, $\operatorname{vol}(A) = |\det(A)|$.

According to the inverse function theorem, the inverse of the Jacobian of a function is the

Jacobian of the inverse function,

$$\boldsymbol{J}_{f^{-1}}(\boldsymbol{x}) = (\boldsymbol{J}_f(f^{-1}(\boldsymbol{x})))^{-1}. \tag{B.5}$$

Appendix C: Fourier Transformation

For $f \in L^1(E^n)$, the Fourier Transform (FT) $\mathrm{FT}[f]$: $E^n \to E$ is defined as:

$$\mathrm{FT}[f](\boldsymbol{\xi}) = \int_{E^n} f(\boldsymbol{x}) e^{-2\pi i \boldsymbol{\xi}^\top \boldsymbol{x}}\, d\boldsymbol{x}, \tag{C.1}$$

where $\xi \in E^n$. The inverse FT is given by:

$$f(\boldsymbol{x}) = \frac{1}{2\pi} \int_{E^n} \mathrm{FT}[f](\boldsymbol{\xi}) e^{i2\pi \boldsymbol{\xi}^\top \boldsymbol{x}}\, d\boldsymbol{\xi}. \tag{C.2}$$

A convolution of two functions $f, g \in L^1(E^n) : E^n \to E$ is defined as:

$$(f * g)(\boldsymbol{x}) = \int_{E^n} f(\boldsymbol{x} - \boldsymbol{y}) g(\boldsymbol{y})\, d\boldsymbol{y}. \tag{C.3}$$

The FT of a convolution is the product of the FTs of the functions (Theorem 1.4 in [49]):

$$\mathrm{FT}[f * g](\boldsymbol{\xi}) = \mathrm{FT}[f](\boldsymbol{\xi}) \mathrm{FT}[g](\boldsymbol{\xi}). \tag{C.4}$$

We say that $f \in L^1(E^n)$ is Gauss summable to l if

$$\lim_{\varepsilon \to 0} G_\varepsilon(f) = \lim_{\varepsilon \to 0} \int_{E^n} f(\boldsymbol{x}) e^{-\varepsilon \|\boldsymbol{x}\|^2}\, d\boldsymbol{x} = l. \tag{C.5}$$

exists and equals l.

It can be put the in the form

$$M_{\varepsilon,\Phi}(f) = M_\varepsilon(f) = \int_{E^n} \Phi(\varepsilon \boldsymbol{x}) f(\boldsymbol{x})\, d\boldsymbol{x} \tag{C.6}$$

where $\Phi \in C_0$ and $\Phi(0) = 1$. Then $\int_{E_n} f$ is summable to l if

$$\lim_{\varepsilon \to 0} M_\varepsilon(f) = l. \tag{C.7}$$

We shall call $M_\varepsilon(f)$ the Φ means of this integral.

The Gauss-Weierstrass (GW) kernel $\phi_\sigma(\boldsymbol{z})$ is defined as:

$$\phi_\sigma(\boldsymbol{z}) = (2\pi\sigma^2)^{-n/2} e^{-\frac{\|\boldsymbol{z}\|^2}{2\sigma^2}}, \tag{C.8}$$

$$\phi_\sigma(\alpha) = (4\pi\alpha)^{-n/2} e^{-\frac{\|\boldsymbol{z}\|^2}{4\alpha}}, \tag{C.9}$$

where $\sigma > 0$ is the standard deviation. The FT of the GW kernel is given by (Theorem 1.13 in [49]):

$$\mathrm{FT}[\phi_\sigma](\boldsymbol{\xi}) = \int e^{-i2\pi\boldsymbol{\xi}^\top \boldsymbol{n}} \phi_\sigma(\boldsymbol{n}) d\mu(\boldsymbol{n}) = e^{-\frac{1}{2}\sigma^2\|\boldsymbol{\xi}\|^2} \tag{C.10}$$

Fact 6 ([49], Theorem 1.16)**.** *If $f : \mathbb{R}^n \to \mathbb{R}$ belongs to $L^1(\mathbb{R}^n)$, the space of all measurable functions defined on $\mathbb{R}^n$ and with $\int_{\mathbb{R}^n} \|f(\boldsymbol{x})\|_1 \, d\boldsymbol{x} < \infty$ ($\|\cdot\|_1$ denotes the 1-norm), then*

$$\int_{\mathbb{R}^n} FT[\phi_\varepsilon](\boldsymbol{\xi}) f(\boldsymbol{\xi}) e^{i2\pi\boldsymbol{\xi}^\top \boldsymbol{z}} \, d\boldsymbol{\xi} = \int_{\mathbb{R}^n} \phi_\varepsilon(\boldsymbol{z} - \boldsymbol{x}) f(\boldsymbol{x}) \, d\boldsymbol{x} \tag{C.11}$$

for all $\varepsilon > 0$.

Bibliography

[1] Gaussian Mixture Model - an overview | ScienceDirect Topics. https://www.sciencedirect.com/topics/mathematics/gaussian-mixture-model.

[2] Learning the parts of objects by non-negative matrix factorization | Nature. https://www.nature.com/articles/44565.

[3] Nonlinear Unmixing of Hyperspectral Images: Models and Algorithms | IEEE Journals & Magazine | IEEE Xplore. https://ieeexplore.ieee.org/document/6678284.

[4] Numerical Optimization | SpringerLink. https://link.springer.com/book/10.1007/978-0-387-40065-5.

[5] A Signal Processing Perspective on Hyperspectral Unmixing: Insights from Remote Sensing | IEEE Journals & Magazine | IEEE Xplore. https://ieeexplore.ieee.org/document/6678258.

[6] S. Achard and C. Jutten. Identifiability of post-nonlinear mixtures. *IEEE Signal Processing Letters*, 12(5):423–426, May 2005.

[7] Yoann Altmann, Nicolas Dobigeon, and Jean-Yves Tourneret. Unsupervised Post-Nonlinear Unmixing of Hyperspectral Images Using a Hamiltonian Monte Carlo Algorithm. *IEEE Transactions on Image Processing*, 23(6):2663–2675, June 2014.

[8] Shunichi Amari. A Theory of Adaptive Pattern Classifiers. *IEEE Transactions on Electronic Computers*, EC-16(3):299–307, June 1967.

[9] Christopher M. Bishop. *Pattern Recognition and Machine Learning*. Information Science and Statistics. Springer, New York, 2006.

[10] V. I. Bogachev. *Measure Theory*. Springer, Berlin ; New York, 2007.

[11] Richard Caron. The Zero Set of a Polynomial, 2005.

[12] Pierre Comon. Independent component analysis, A new concept? *Signal Processing*, 36(3):287–314, April 1994.

[13] Sergio Cruces. Bounded Component Analysis of Linear Mixtures: A Criterion of Minimum Convex Perimeter. *IEEE Transactions on Signal Processing*, 58(4):2141–2154, April 2010.

[14] Yannick Deville. From separability/identifiability properties of bilinear and linear-quadratic mixture matrix factorization to factorization algorithms. *Digital Signal Processing*, 87:21–33, April 2019.

[15] Nicolas Dobigeon, SaÏd Moussaoui, Martial Coulon, Jean-Yves Tourneret, and Alfred O. Hero. Joint Bayesian Endmember Extraction and Linear Unmixing for Hyperspectral Imagery. *IEEE Transactions on Signal Processing*, 57(11):4355–4368, November 2009.

[16] Denis G. Fantinato, Leonardo T. Duarte, Yannick Deville, Romis Attux, Christian Jutten, and Aline Neves. A second-order statistics method for blind source separation in post-nonlinear mixtures. *Signal Processing*, 155:63–72, February 2019.

[17] Richard P. Feynman, Albert R. Hibbs, and Daniel F. Styer. *Quantum Mechanics and Path Integrals*. Courier Corporation, July 2010.

[18] Xiao Fu, Kejun Huang, Nicholas D. Sidiropoulos, and Wing-Kin Ma. Nonnegative Matrix Factorization for Signal and Data Analytics: Identifiability, Algorithms, and Applications. *IEEE Signal Processing Magazine*, 36(2):59–80, March 2019.

[19] Xiao Fu, Kejun Huang, Bo Yang, Wing-Kin Ma, and Nicholas D. Sidiropoulos. Robust Volume Minimization-Based Matrix Factorization for Remote Sensing and Document Clustering, August 2016.

[20] Xiao Fu, Wing-Kin Ma, Kejun Huang, and Nicholas D. Sidiropoulos. Blind Separation of Quasi-Stationary Sources: Exploiting Convex Geometry in Covariance Domain. *IEEE Transactions on Signal Processing*, 63(9):2306–2320, May 2015.

[21] Ian J. Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative Adversarial Networks, June 2014.

[22] Michael U Gutmann, Michael Gutmann, Aapo Hyvarinen, and Aapo Hyvarinen. Noise-Contrastive Estimation of Unnormalized Statistical Models, with Applications to Natural Image Statistics.

[23] Kejun Huang and Xiao Fu. Detecting Overlapping and Correlated Communities without Pure Nodes: Identifiability and Algorithm. In *Proceedings of the 36th International Conference on Machine Learning*, pages 2859–2868. PMLR, May 2019.

[24] Aapo Hyvärinen, Juha Karhunen, and Erkki Oja. *Independent Component Analysis*. J. Wiley, New York, 2001.

[25] Aapo Hyvärinen, Ilyes Khemakhem, and Ricardo Monti. Identifiability of latent-variable and structural-equation models: From linear to nonlinear. *Annals of the Institute of Statistical Mathematics*, 76(1):1–33, February 2024.

[26] Aapo Hyvärinen and Hiroshi Morioka. Unsupervised Feature Extraction by Time-Contrastive Learning and Nonlinear ICA. In *Advances in Neural Information Processing Systems*, volume 29. Curran Associates, Inc., 2016.

[27] Aapo Hyvärinen and Petteri Pajunen. Nonlinear independent component analysis: Existence and uniqueness results. *Neural Networks*, 12(3):429–439, April 1999.

[28] Aapo Hyvärinen, Hiroaki Sasaki, and Richard Turner. Nonlinear ICA Using Auxiliary Variables and Generalized Contrastive Learning. In *Proceedings of the Twenty-Second International Conference on Artificial Intelligence and Statistics*, pages 859–868. PMLR, April 2019.

[29] Michael I. Jordan, Zoubin Ghahramani, Tommi S. Jaakkola, and Lawrence K. Saul. An Introduction to Variational Methods for Graphical Models. In Michael I. Jordan, editor, *Learning in Graphical Models*, pages 105–161. Springer Netherlands, Dordrecht, 1998.

[30] Jagat Narain Kapur. *Maximum-Entropy Models in Science and Engineering*. John Wiley & Sons, 1989.

[31] Ilyes Khemakhem, Diederik Kingma, Ricardo Monti, and Aapo Hyvarinen. Variational Autoencoders and Nonlinear ICA: A Unifying Framework. In *Proceedings of the Twenty Third International Conference on Artificial Intelligence and Statistics*, pages 2207–2217. PMLR, June 2020.

[32] Ilyes Khemakhem, Diederik Kingma, Ricardo Monti, and Aapo Hyvärinen. Variational Autoencoders and Nonlinear ICA: A Unifying Framework. In *Proceedings of the Twenty Third International Conference on Artificial Intelligence and Statistics*, pages 2207–2217. PMLR, June 2020.

[33] Diederik P. Kingma and Jimmy Ba. Adam: A Method for Stochastic Optimization, January 2017.

[34] Diederik P. Kingma and Max Welling. Auto-Encoding Variational Bayes, December 2022.

[35] Yi-Ou Li, Tülay Adali, Wei Wang, and Vince D. Calhoun. Joint Blind Source Separation by Multiset Canonical Correlation Analysis. *IEEE Transactions on Signal Processing*, 57(10):3918–3929, October 2009.

[36] Yuening Li, Xiao Fu, and Wing-Kin Ma. Probabilistc Simplex Component Analysis Via Variational Auto-Encoding. 2024.

[37] Seppo Linnainmaa. Taylor expansion of the accumulated rounding error. *BIT Numerical Mathematics*, 16(2):146–160, June 1976.

[38] Francesco Locatello, Stefan Bauer, Mario Lucic, Gunnar Rätsch, Sylvain Gelly, Bernhard Schölkopf, and Olivier Bachem. Challenging Common Assumptions in the Unsupervised Learning of Disentangled Representations, June 2019.

[39] Qi Lyu and Xiao Fu. Nonlinear Multiview Analysis: Identifiability and Neural Network-Assisted Implementation. *IEEE Transactions on Signal Processing*, 68:2697–2712, 2020.

[40] Qi Lyu and Xiao Fu. Identifiability-Guaranteed Simplex-Structured Post-Nonlinear Mixture Learning via Autoencoder. *IEEE Transactions on Signal Processing*, 69:4921–4936, 2021.

[41] Qi Lyu and Xiao Fu. Provable Subspace Identification Under Post-Nonlinear Mixtures. 2022.

[42] Nestor Maslej, Loredana Fattorini, Raymond Perrault, Vanessa Parli, Anka Reuel, Erik Brynjolfsson, John Etchemendy, Katrina Ligett, Terah Lyons, James Manyika, Juan Carlos Niebles, Yoav Shoham, Russell Wald, and Jack Clark. Artificial Intelligence Index Report 2024, May 2024.

[43] Kai Wang Ng, Guo-Liang Tian, and Man-Lai Tang. *Dirichlet and Related Distributions: Theory, Methods and Applications*. Wiley Series in Probability and Statistics. Wiley, 1 edition, April 2011.

[44] John T. Oden and Junuthula N. Reddy. *Variational Methods in Theoretical Mechanics*. Universitext. Springer, Berlin, Heidelberg, 1983.

[45] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-Resolution Image Synthesis with Latent Diffusion Models, April 2022.

[46] David E. Rumelhart, Geoffrey E. Hinton, and Ronald J. Williams. Learning representations by back-propagating errors. *Nature*, 323(6088):533–536, October 1986.

[47] Claude Sammut. Markov Chain Monte Carlo. In Claude Sammut and Geoffrey I. Webb, editors, *Encyclopedia of Machine Learning*, pages 639–642. Springer US, Boston, MA, 2010.

[48] N.D. Sidiropoulos, R. Bro, and G.B. Giannakis. Parallel factor analysis in sensor array processing. *IEEE Transactions on Signal Processing*, 48(8):2377–2388, Aug./2000.

[49] Elias M. Stein and Guido Weiss. *Introduction to Fourier Analysis on Euclidean Spaces*. Number 32 in Princeton Mathematical Series. Princeton University Press, Princeton, N.J, 1975.

[50] Anisse Taleb and Christian Jutten. Source separation in post-nonlinear mixtures. *Signal Processing, IEEE Transactions on*, 47:2807–2820, November 1999.

[51] Ruiyuan Wu, Wing-Kin Ma, Yuening Li, Anthony Man-Cho So, and Nicholas D. Sidiropoulos. Probabilistic Simplex Component Analysis. *IEEE Transactions on Signal Processing*, 70:582–599, 2022.

[52] Bo Yang, Xiao Fu, Nicholas D. Sidiropoulos, and Kejun Huang. Learning Nonlinear Mixtures: Identifiability and Algorithm. *IEEE Transactions on Signal Processing*, 68:2857–2869, 2020.

[53] Guoxu Zhou, Shengli Xie, Zuyuan Yang, Jun-Mei Yang, and Zhaoshui He. Minimum-Volume-Constrained Nonnegative Matrix Factorization: Enhanced Ability of Learning Parts. *IEEE Transactions on Neural Networks*, 22(10):1626–1637, October 2011.

[54] Michael Zibulevsky and Barak A. Pearlmutter. Blind Source Separation by Sparse Decomposition in a Signal Dictionary. *Neural Computation*, 13(4):863–882, April 2001.

[55] A. Ziehe, K. R. Müller, G. Nolte, B. M. Mackert, and G. Curio. Artifact reduction in magnetoneurography based on time-delayed second-order correlations. *IEEE transactions on bio-medical engineering*, 47(1):75–87, January 2000.