Provable Subspace Identification Under Post-Nonlinear Mixtures

Anonymous Author(s)
Affiliation
Address
email

Abstract

Unsupervised mixture learning (UML) aims at identifying linearly or nonlinearly mixed latent components in a blind manner. UML is known to be challenging: Even learning linear mixtures requires highly nontrivial analytical tools, e.g., independent component analysis or nonnegative matrix factorization. In this work, the post-nonlinear (PNL) mixture model is revisited, where *unknown* element-wise nonlinear functions are imposed after a linear mixture, making the identification problem even more ill-posed. The PNL model is widely employed in different fields ranging from brain signal classification, speech separation, remote sensing, to causal discovery. Existing works often assume different properties on the latent components (e.g., statistical independence or probability-simplex structures) to identify and remove the unknown nonlinear functions. This work shows that the existence of a nontrivial *null space* associated with the underlying mixing system suffices to guarantee identification/removal of the unknown nonlinearity. Compared to the existing works, this finding largely relaxes the model identification conditions of PNL models. Consequently, a simple learning criterion is proposed that could benefit applications where no strong structural information on the latent components is known. A sample complexity analysis is offered to characterize the performance of the proposed approach under realistic settings. For implementation, we design an optimization strategy that features a block coordinate descent algorithm. A series of numerical experiments corroborate our theoretical claims.

1 Introduction

Unsupervised mixture learning (UML) has been broadly used in machine learning tasks, for recovering mixed latent components or extracting representative information. Central to the UML problem is the model identifiability problem, which answers the question if the latent components can be identified with guarantees and inconsequential ambiguities—without using any label or training data. This is a challenging problem due to the unsupervised nature. To address the identifiability challenge, linear mixture model identification problems were tackled via exploiting properties of the latent components, e.g., statistical independence [1], nonegativity [2, 3], quasi-stationarity [4], boundedness [5]. This leads to a cohort of *blind source separation* (BSS) methods.

However, linear mixture models oftentimes fail to to perform well on real-world data, due to the unknown nonlinear distortions arising in data generation and acquisition. In the past two decades, nonlinear mixture models have attracted unprecedentedly more attentions [6–12]. However, it is shown in [13] that the nonlinear mixture models are in general not identifiable, even with strong assumptions, e.g., statistical independence among the latent components. To establish identifability, a line of work exploited observable auxiliary variables or specific priors (i.e., extra information about the data or the latent components on top of statistical independence); see, e.g., [11,12,14,15]. Another line of methods take an alternative route via utilizing the structures of the nonlinear distortions. In

Submitted to 36th Conference on Neural Information Processing Systems (NeurIPS 2022). Do not distribute.

particular, the post-nonlinear (PNL) model [6, 7, 16] is a reasonable extension of the classical linear model by assuming *unknown* nonlinear transformations individually imposed on the outputs of a underlying linear mixture model. This structured nonlinear mixture model is used in a wide range of applications, e.g. hyperspectral imaging [17], brain data classification [18–20], and causal discovery [10, 21, 22]. In order to identify the generative model/latent components under the PNL model, a key step is to remove the nonlinear distortions in an unsupervised manner. To this end, the existing methods often explicitly rely on some properties of the latent components, e.g., statistical independence [6, 7, 16], availability of multiview data [23] and the sum-to-one and nonnegativity (or probability-simplex) structure of the latent components [24, 25]. These structures stem from specific applications, and thus are meaningful—yet not always available. A natural question is whether the nonlinear distortions are still removable if the widely used structural assumptions (e.g., those in [6, 7, 16, 23–25]) do not hold?

In this work, we offer a positive answer to the above inquiry—and build a new learning algorithm based on our analyses. Our detailed contributions are as follows.

Subspace-Based PNL Model Identification. We propose a new nonlinearity identification and removal criterion under the PNL model. We show that, as long as the underlying linear mixing system of the PNL model admits a nontrivial orthogonal complement (or, its transpose has a nontrivial null space), the unknown nonlinear transformations are identifiable and removable, under mild conditions. Notably, our approach does not use any specific structural assumptions of the latent components. Our finding presents a fairly different result compared to existing nonlinear mixture learning techniques. The latter often heavily rely on the properties of the latent components. Hence, our result may offer new insights and a useful alternative to this long-standing challenge. An immediate consequence is that our method can work with various types of latent components, including those are not covered by existing PNL model identification approaches in [6, 7, 16, 23–25].

Performance Analysis Under Realistic Settings. The model identification analysis is conducted under the population case where unlimited data is available, with the assumption that a universal function learner can be used to cancel any nonlinear distortion. To advance the understanding under more realistic settings, we also consider the realistic scenario with finite sample and non-universal function learners. The analysis is a combination or statistical generalization theory and numerical differentiation. The analytical strategy is reminiscent of the recently developed framework in [23, 26], with nontrivial derivations to accommodate our learning criterion.

Optimization Algorithm Design. Based on the theoretical analysis, we formulate the identification problem as a constrained optimization. The problem is challenging, as it involves learning unknown nonlinear functions represented by neural networks (NNs) under nonconvex constraints. We propose a block coordinate descent (BCD) algorithm that alternates between two subproblems, namely, null space basis learning and neural network based nonlinear function learning. This way, the latter is disentangled with constraints, and thus off-the-shelf NN training techniques (e.g., backpropagation and Adam [27]) can be readily employed to handle the subproblem. A series of experiments are conducted for validating our theoretical claims.

Notations. The notations used are summarized as follows. $x, \boldsymbol{x}, \boldsymbol{X}$ represent a scalar, vector, and matrix, respectively; $\|\boldsymbol{x}\|_0$, $\|\boldsymbol{x}\|_1$ and $\|\boldsymbol{x}\|_2$ denote the ℓ_0 norm, ℓ_1 norm and the Euclidean norm, respectively; $\odot$ and $\circledast$ denotes the Khatri-Rao product and the Hadamard product, respectively; f is used to denote a function $f(\cdot) : \mathbb{R} \to \mathbb{R}$; f', f'' and $f^{(n)}$ denote the first-order, second-order and nth order derivatives of the function f, respectively; $f \circ g$ denotes the function composition; $\mathcal{N}(\boldsymbol{X})$ means the null space of $\boldsymbol{X}$.

2 Background

2.1 UML and Model Identification

In UML, the classical linear mixture model is defined as follows [28]:

$$\boldsymbol{x}_\ell = \boldsymbol{A}\boldsymbol{s}_\ell,\ \ell = 1, 2, \cdots, N, \tag{1}$$

where $\boldsymbol{x}_\ell \in \mathbb{R}^M$ is the ℓth observation sample, $\boldsymbol{s}_\ell \in \mathbb{R}^K$ contains the K latent components, and $\boldsymbol{A} \in \mathbb{R}^{M\times K}$ is the mixing system. The latent components of linear mixture model in (1) are generally not identifiable, if $\boldsymbol{A}$ is not known [28]. As a remedy, information about $\boldsymbol{A}$ and/or $\boldsymbol{s}_\ell$ is often used to

establish identifiability. For instance, statistical independence among $[\boldsymbol{s}_\ell]_i$'s is used in independent component analysis (ICA) [1], and the nonnegativity of $\boldsymbol{A} \in \mathbb{R}_+^{M\times K}$ and $\boldsymbol{S} = [\boldsymbol{s}_1, \cdots, \boldsymbol{s}_N] \in \mathbb{R}_+^{K\times N}$ are used in nonnegative matrix factorization (NMF) [2, 3]; also see [4, 5, 29–31] for more conditions that can assist linear mixture identification.

The linear mixture model in (1) is often over-simplified for real applications. Nonlinear effects are widely observed in data generation and acquisition. In many cases, the following nonlinear mixture model may be of more interest [32]:

$$\boldsymbol{x}_\ell = \boldsymbol{g}(\boldsymbol{s}_\ell),\ \ell = 1, 2, \cdots, N, \tag{2}$$

where $\boldsymbol{g} : \mathbb{R}^K \to \mathbb{R}^M$ is a structure-free invertible nonlinear function. However, it is shown in [32] that the model in (2) is not identifiable, even if the latent components are assumed to be statistically independent. A workaround is to use more information of or additional conditions on the latent components, e.g., the availability of auxiliary variables [12] exponential family priors [14], and the slow changing property [11]. Another approach to establish model identifiability of nonlinear mixtures is by exploiting structural assumptions of the nonlinear mixing process. This leads to the suite of PNL mixture model learning works (see, e.g., [6, 7, 16, 33]), which will be our focus.

2.2 Post-Nonlinear Mixture Model

The PNL mixture model is a natural extension to the linear mixture model [6, 7, 16, 23, 24, 33]:

$$\boldsymbol{x}_\ell = \boldsymbol{g}(\boldsymbol{A}\boldsymbol{s}_\ell),\ \ell = 1, 2, \cdots, N, \tag{3}$$

where $\boldsymbol{g} = [g_1(\cdot), \cdots, g_M(\cdot)]^\top$ with each $g_m(\cdot) : \mathbb{R} \to \mathbb{R}$ being an invertible nonlinear function. We assume $\boldsymbol{s}_\ell \sim p(\boldsymbol{s})$ and the support of $p(\boldsymbol{s})$ is a continuous and open domain $\mathcal{S}$.

Although the PNL model is not the most general one, it has a myriad of of applications across different fields, e.g., causal discovery [10, 21, 22], hyperspectral imaging [17] and brain data classification [18–20]. It is considered as an appropriate model when the unknown nonlinear distortions happen at the sensor end in data acquisition. It also offers a valuable generalization to the linear mixture generative models in data analytics, e.g., those used in NMF and data clustering [3].

An important remark is that *the challenge of UML under the PNL model boils down to identifying and removing* $\boldsymbol{g}(\cdot)$, as this will lead to a well understood linear mixture model in (1). Then, any linear mixture model identification approach in [5, 28, 29, 31, 34] can be applied to identify the latent components $\boldsymbol{s}_\ell$ for $\ell = 1, \ldots, N$.

Existing works tackle the $\boldsymbol{g}$-identification and removal problem from different angles. In [6, 16], the statistical independent among the latent components (i.e., $[\boldsymbol{s}_\ell]_k$ for $k = 1, \ldots, K$) is used. Recently, [24, 25] proposed to identify the PNL model with latent components satisfying the probability simplex constraint, i.e., $\mathbf{1}^\top \boldsymbol{s}_\ell = 1,\ \boldsymbol{s}_\ell \geq \mathbf{0}$, that often arises in soft clustering problems. Beyond that, [23] shows $\boldsymbol{g}(\cdot)$ can be provably removed with the assistance of multiple views of data, where identical copies of $\boldsymbol{s}_\ell$ appears in the latent space of more than one data domain. These methods offered insightful and useful approaches to remove $\boldsymbol{g}(\cdot)$, but the use of specific structural assumptions on the latent components make their applicability relatively narrow.

Unlike prior works, we propose to pin down nonlinear distortions by simply utilizing a subspace structure, instead of resorting to any specific structure of $\boldsymbol{s}_\ell$. Our method starts with a rudimentary observation: Note that if the mixing system $\boldsymbol{A}$ is a tall matrix, $\mathcal{N}(\boldsymbol{A}^\top)$ is a nontrivial subspace. This implies that if only the linear mixture part is considered (i.e., if $\boldsymbol{g}(\cdot)$ can be removed completely), there exists vector $\boldsymbol{v}$ such that

$$\boldsymbol{v}^\top \boldsymbol{A}\boldsymbol{s}_{\boldsymbol{\ell}} = 0,\ \forall \ell, \tag{4}$$

where the nonzero $\boldsymbol{v}$ is a vector from $\mathcal{N}(\boldsymbol{A}^\top)$. In the next section, we will show how this simple equation (4) allows for constructing a provable $\boldsymbol{g}$-identification and removal criterion.

3 Problem Formulation

In this section, we propose a learning criterion to identify/remove the nonlinear transformations $\boldsymbol{g}(\cdot)$. Here, we consider the population case where infinite data points are available as $\boldsymbol{x}_\ell \sim p(\boldsymbol{x})$ over the continuous support $\mathcal{X}$ of $p(\boldsymbol{x})$. Under this assumption, the (idealistic) formulation is as follows:

$$\text{find } \boldsymbol{Q},\ \boldsymbol{f}(\cdot) = [f_1(\cdot), \cdots, f_M(\cdot)]^\top \tag{5a}$$
$$\text{s.t. } \boldsymbol{Q}^\top \boldsymbol{f}(\boldsymbol{x}_\ell) = \boldsymbol{0},\ \forall \boldsymbol{x}_\ell \in \mathcal{X} \tag{5b}$$
$$f_m(\cdot) : \mathbb{R} \to \mathbb{R} \text{ is invertible } \forall m, \tag{5c}$$
$$\|\boldsymbol{Q}\|_0 = MD,\ \boldsymbol{Q}^\top \boldsymbol{Q} = \boldsymbol{I}, \tag{5d}$$

where $\boldsymbol{Q} \in \mathbb{R}^{M \times D}$, D is the dimension of the null space (i.e., $\mathcal{N}(\boldsymbol{A}^\top)$) that we hope to find with $D \geq 1$ and $D < M$. The ℓ_0-norm constraint in (5d) requires $\boldsymbol{Q}$ to be a dense matrix, which is important to establishing identifiability. This will become clearer in Sec. 4. In the formulation, $f_m(\cdot)$ is a nonlinear invertible function that we look for. Ideally, we hope to learn $f_m(\cdot) = g_m^{-1}(\cdot)$ which cancels the nonlinear transformation. Then, $\boldsymbol{Q}$ forms a basis of $\mathcal{N}(\boldsymbol{A}^\top)$. Note that under the (5d) constraint, a possible trivial solution with $\boldsymbol{Q} = \boldsymbol{0}$ is avoided. Besides, the invertibility constraint in 5c is used to prevent other degenerate solutions, e.g., $\boldsymbol{f}(\boldsymbol{x}_\ell) = \boldsymbol{c}$ for all ℓ with a constant $\boldsymbol{c}$.

4 Provable Nonlinearity Removal

In this section, we will show that by solving (5), it is guaranteed that the nonlinear transformation $\boldsymbol{g}(\cdot)$ is removed.

First, we give the following condition which will be useful in the proof.

Definition 1 (Locally Free Variables) *Consider real-valued random components $\boldsymbol{v} = [v_1, \ldots, v_d]^\top \in \mathbb{R}^d$, where v_i resides in a continuous and open set $\mathcal{V}_i \subseteq \mathbb{R}$. Assume that the following holds for all j*

$$p(v_j | \boldsymbol{v}_{-j}) > 0, \text{ with continuous support } \mathcal{V}_{j|\boldsymbol{v}_{-j}} \text{ given any } \boldsymbol{v}_{-j},$$

where $\boldsymbol{v}_{-j}$ denotes the vector with all entries but v_j. Then, the components in $\boldsymbol{v}$ all locally free variables.

Note that the definition above specifies that each v_j has some degree of freedom no matter what value the $\boldsymbol{v}_{-j}$ part is. As a special case, independent random variable $\boldsymbol{v}$ clearly satisfies the definition of free variable. However, statistical independence is a much stronger condition. Dependent variables could also be free variables. For instance, the vectors that are subject to the probability simplex, i.e.,

$$\boldsymbol{v} \in \boldsymbol{\Delta}_K,\ \boldsymbol{\Delta}_K = \{\boldsymbol{v} \in \mathbb{R}^K | \boldsymbol{1}^\top \boldsymbol{v} = 1, \boldsymbol{v} > \boldsymbol{0}\}, \tag{6}$$

also share this property for any $\boldsymbol{v}_{-k}$, which means any $K-1$ of the K components in $\boldsymbol{v}$ are free variables. In particular, with v_i and v_j being free variables, we could observe any possible value of v_j given a fixed $\bar{v}_i$. In particular, consider the functional equation $\boldsymbol{q}^\top \boldsymbol{h}(\boldsymbol{A}\boldsymbol{v}) = 0$ that holds for all $\boldsymbol{v}$. Then, we can always observe

$$\boldsymbol{q}^\top \boldsymbol{h}\left(\boldsymbol{A}[\bar{v}_1, \cdots, v_j, \cdots, \bar{v}_d]^\top\right) = 0, \tag{7}$$

with fixed $\bar{\boldsymbol{v}}_{-j}$ and v_j is from certain continuous support with $p(v_j | \bar{\boldsymbol{v}}_{-j}) > 0$. This implies we may take derivative w.r.t. v_j on the equation above, which gives us

$$(\boldsymbol{q}^\top \odot \boldsymbol{a}_j^\top) \boldsymbol{h}'\left(\boldsymbol{A}[\bar{v}_1, \cdots, v_j, \cdots, \bar{v}_d]^\top\right) = 0, \tag{8}$$

as all the terms $\frac{\bar{v}_i}{v_j} = 0$ for any $i \neq j$. This property will help us show the identifiability result next.

Given the conditions, we have the following theorem for identifiability of the PNL model.

Theorem 1 (Nonlinearity Identification and Removal) *Under the model in* (3)*, assume that the criterion* (5) *is solved. In addition, assume that $\boldsymbol{A} \in \mathbb{R}^{M \times K}$ is drawn from any joint absolutely continuous distribution, $\widetilde{K}$ of the K components of $\boldsymbol{s}$ are free variables (cf. Def. 1), and that the learned $\widehat{h}_m = \widehat{f}_m \circ g_m$ is twice differentiable for all $m \in [M]$. Suppose that*

$$\frac{\widetilde{K}(\widetilde{K}+1)}{2} \geq M. \tag{9}$$

Then, almost surely, any $\widehat{\boldsymbol{f}}(\cdot)$ that is a solution of (5) *satisfies $\widehat{h}_m(x) = \widehat{f}_m \circ g_m(x) = c_m x + d_m$, $\forall m \in [M]$, where $c_m \neq 0$ and d_m is a constant, in the limit of infinite samples.*

Note that Theorem 1 indicates that if any solution $\widehat{\boldsymbol{f}}(\cdot)$ of (5) is found, then the composition $\widehat{h}_m = \widehat{f}_m \circ g_m$ is an affine function for all m. We do not require all the K latent components to be free variables as long as the condition $\frac{\widetilde{K}(\widetilde{K}+1)}{2} \geq M$ is satisfied. Intuitively, if $\widetilde{K}$ is too small then it means that the remaining $K - \widetilde{K}$ components do not provide useful information, while a larger $\widetilde{K}$ implies that there is more diversity among the latent components to assist nonlinearity removal. Note that the existence of the solution implies that the null space $\mathcal{N}(\boldsymbol{A}^\top)$ must exist ($D \geq 1$). Otherwise, one may fail to learn a nonzero $\boldsymbol{Q}$. We show the sketch of proof as follows.

Proof Sketch: By solving (5), we have the following equation for all $\boldsymbol{s}_\ell \in \mathcal{S}$

$$\boldsymbol{Q}^\top \boldsymbol{h}(\boldsymbol{A}\boldsymbol{s}_\ell) = \boldsymbol{0},$$

where $\boldsymbol{h} = \boldsymbol{f} \circ \boldsymbol{g}$ is the function composition.

First, we show the existence of solution to (5). It is obvious that a solution is $\boldsymbol{f} = \boldsymbol{g}^{-1}$, columns of $\boldsymbol{Q}$ being an orthogonal basis of $\mathcal{N}(\boldsymbol{A}^\top)$ and a dense matrix. Next, we show that such a solution is unique up to only affine ambiguities.

By the assumptions, there are $\widetilde{K}$ of the K components of $\boldsymbol{s}$ are free variables. Thus, by taking the second-order (cross) derivatives of the equation above w.r.t. each of the $\widetilde{K}$ variables, as shown in (8), we have the following linear system

$$\boldsymbol{Q}^\top \odot \underbrace{\begin{bmatrix} (\boldsymbol{a}_1 \circledast \boldsymbol{a}_1)^\top \\ \vdots \\ \left(\boldsymbol{a}_{\widetilde{K}} \circledast \boldsymbol{a}_{\widetilde{K}}\right)^\top \\ (\boldsymbol{a}_1 \circledast \boldsymbol{a}_2)^\top \\ \vdots \\ \left(\boldsymbol{a}_{\widetilde{K}-1} \circledast \boldsymbol{a}_{\widetilde{K}}\right)^\top \end{bmatrix}}_{\widehat{\boldsymbol{A}}} \begin{bmatrix} \widehat{h}''_1([\boldsymbol{A}\boldsymbol{s}]_1) \\ \vdots \\ \widehat{h}''_M([\boldsymbol{A}\boldsymbol{s}]_M) \end{bmatrix} = \boldsymbol{0}, \ \forall \boldsymbol{s} \in \mathcal{S}, \tag{10}$$

where $\boldsymbol{a}_k$ is the kth column of the $\boldsymbol{A}$ matrix which corresponds to one of the free variables (without loss of generality, we assume that the first $\widetilde{K}$ variables of $\boldsymbol{s}$ are locally free variables).

It is straightforward that if one can show $\widehat{h}''_m([\boldsymbol{A}\boldsymbol{s}]_m) = 0$ for any m over any $\boldsymbol{s}$, then it implies that any $\widehat{h}_m(\cdot)$ is an affine function. Following this thought, we actually need to show that $\boldsymbol{Q}^\top \odot \widehat{\boldsymbol{A}}$ is full column rank with $\boldsymbol{Q} \in \mathbb{R}^{M \times D}$ and $\widehat{\boldsymbol{A}} \in \mathbb{R}^{\frac{\widetilde{K}(\widetilde{K}+1)}{2} \times M}$.

By Lemma 1 of [35], for matrices $\boldsymbol{U} \in \mathbb{R}^{a \times m}$ and $\boldsymbol{V} \in \mathbb{R}^{b \times m}$, $\boldsymbol{U} \odot \boldsymbol{V}$ has full column rank if $\mathrm{krank}(\boldsymbol{U}) + \mathrm{krank}(\boldsymbol{V}) \geq m + 1$ with $\mathrm{krank}(\boldsymbol{U})$ and $\mathrm{krank}(\boldsymbol{B})$ being the Kruskal rank of $\boldsymbol{U}$ and $\boldsymbol{V}$, respectively. The Kruskal rank of matrix $\boldsymbol{U}$ is defined as the largest number τ such that every set of τ columns of $\boldsymbol{U}$ is linearly independent. For $\boldsymbol{Q}^\top \odot \widehat{\boldsymbol{A}}$, we do not have control over $\boldsymbol{Q}$ since it is an optimization variable. But the Kruskal rank $k_{Q^\top}$ is at least 1 since we have the constraint $\|\boldsymbol{Q}\|_0 = MD$. In terms of $\widehat{\boldsymbol{A}}$, we can show that the matrix $\widehat{\boldsymbol{A}}$ has full Kruskal rank $\min\left(\frac{\widetilde{K}(\widetilde{K}+1)}{2}, M\right)$. The detailed proof can be found in Appendix B.

To derive the conditions required for model identification (i.e., $\mathrm{rank}(\boldsymbol{Q}^\top \odot \widehat{\boldsymbol{A}}) = M$), we consider the following two cases

a) For the case $\mathrm{krank}(\widehat{\boldsymbol{A}}) = \frac{\widetilde{K}(\widetilde{K}+1)}{2}$, it requires

$$M \geq \frac{\widetilde{K}(\widetilde{K}+1)}{2} \quad \text{and} \quad 1 + \frac{\widetilde{K}(\widetilde{K}+1)}{2} \geq M + 1,$$

which is $M = \frac{\widetilde{K}(\widetilde{K}+1)}{2}$.

b) For the case $\mathrm{krank}(\widehat{\boldsymbol{A}}) = M$, it is required that

$$M \leq \frac{\widetilde{K}(\widetilde{K}+1)}{2} \quad \text{and} \quad 1 + M \geq M + 1,$$

which is $\frac{\widetilde{K}(\widetilde{K}+1)}{2} \geq M$.

The above implies that if $\frac{\widetilde{K}(\widetilde{K}+1)}{2} \geq M$ is satisfied, then the nonlinear functions $g_m(\cdot)$'s can be removed almost surely. ■

5 Sample Complexity Analysis

The identifiability analysis in Sec. 4 is based on the assumption that (uncountably) infinite samples are available. In practice, there are only finite number of samples. In addition, the proof of Theorem 1 assumes that the $f_m(\cdot)$'s are universal function approximators, which is also overly ideal. In practice, the function learners, e.g., neural networks, may have nontrivial approximation errors for modeling nonlinear functions. In this section, we provide analyses under more realistic conditions.

To begin with, note that when only N i.i.d. samples are available, one could rewrite the finite-sample version of (15) as follows

$$\min_{\boldsymbol{Q},\boldsymbol{f}} \ \frac{1}{N}\sum_{\ell=1}^{N} \left\|\boldsymbol{Q}^\top \boldsymbol{f}(\boldsymbol{x}_\ell)\right\|_2^2 \tag{11a}$$

$$\text{s.t.} \quad \|\boldsymbol{Q}\|_0 = MD,\ \boldsymbol{Q}^\top\boldsymbol{Q} = \boldsymbol{I},\ f_m(\cdot) \in \mathcal{F}, \tag{11b}$$

where $\mathcal{F} : \mathbb{R} \to \mathbb{R}$ is an invertible function class used for modeling the inverse of function $g_m(\cdot)$. Note that in the population case (i.e., $N = \infty$), (11) and (5) have the same solutions. To characterize the finite sample performance of (11) under $\boldsymbol{f}(\cdot)$ that has limited expressive power, we will use the following definition and assumptions.

Definition 2 (Function Class and Inverse) *Assume $g \in \mathcal{G}$ is an invertible continuous nonlinear function. We define its inverse class $\mathcal{G}^{-1}$ as a function class that contains all the u's satisfying $u \circ g(y) = \alpha y + \beta$ with $\alpha \neq 0$, $\forall g \in \mathcal{G}$.*

Assumption 1 (Realization Gap) *Assume that for any $u \in \mathcal{G}^{-1}$, there exists $f \in \mathcal{F}$ such that*

$$\sup_{\boldsymbol{x}\in\mathcal{X}} |f(x_m) - u(x_m)| \leq \nu,\ \forall m \in [M].$$

Assumption 2 (Boundedness) *Assume that the 4th-order derivatives of $f_m \in \mathcal{F}$ and $g_m \in \mathcal{G}$ exist. In addition, $|f_m^{(n)}(\cdot)|$ and $|g_m^{(n)}(\cdot)|$ are bounded for all $n \in [4]$ and $m \in [M]$. Any 4th-order derivative of $\boldsymbol{q}_k^\top \boldsymbol{h}(\boldsymbol{A}\boldsymbol{s}_\ell)$ with $k = [D]$ is bounded by C_d. Assume that $\boldsymbol{A}$ has bounded elements.*

Assumption 3 (Neural Network Structure) *Assume that $f_m(\cdot)$ is parameterized by a one-hidden-layer neural network with R neurons and 1-Lipschitz nonlinear activation function $\zeta(\cdot) : \mathbb{R} \to \mathbb{R}$ with $\zeta(0) = 0$.*

$$\mathcal{F} = \{f_m | f_m(x) = \boldsymbol{w}_2^\top \boldsymbol{\zeta}(\boldsymbol{w}_1 x), \|\boldsymbol{w}_i\|_2 \leq B,\ i = 1, 2\}, \tag{12}$$

where $\boldsymbol{\zeta}(\boldsymbol{y}) = [\zeta(y_1), \ldots, \zeta(y_R)]^\top$, $\boldsymbol{w}_i \in \mathbb{R}^R$ for $i = 1, 2$.

Next, we show the sample complexity results for the case where the above neural network class is used. To this end, we use the following lemma:

Lemma 1 *Assume that $\mathcal{F}$ is the function class defined in Assumption 3 and the input $\boldsymbol{x}$ is bounded by $|x_m| \leq C_x$ for all m. Then, the loss function $\left\|\boldsymbol{Q}^\top \boldsymbol{f}(\boldsymbol{x}_\ell)\right\|_2^2$ has the following Rademacher complexity*

$$\mathfrak{R} \leq 2DMB^4C_x^2\sqrt{\frac{R}{N}}. \tag{13}$$

The proof can be found in Appendix C. Note that Lemma 1 indicates that the complexity of the function class is proportional to the width of the neural network R and inversely proportional to the sample size N. With Lemma 1, we give the following theorem on sample complexity.

Theorem 2 (Sample Complexity) *Under the generative model* (3)*, assume that Assumptions 1, 2 and 3 hold. Suppose that $\boldsymbol{x}_\ell$ for $\ell \in [N]$ are i.i.d. samples from $\mathcal{X}$ according to a certain distribution $\mathcal{D}$. Denote any solution of* (11) *as $\widehat{\boldsymbol{f}} = [\widehat{f}_1, \ldots, \widehat{f}_M]^\top$ and $\widehat{h}_m = \widehat{f}_m \circ g_m$ for $m \in [M]$. Then, with probability of at least $1-\delta$, it holds that*

$$\mathbb{E}\left[\left\|\widehat{\boldsymbol{h}}''(\boldsymbol{A}\boldsymbol{s})\right\|_2^2\right] = O\left(\frac{C_d\sqrt{M}\nu}{\sigma_{\min}^2(\boldsymbol{Q}^\top \odot \widehat{\boldsymbol{A}})} + \frac{C_d B^2 C_x \left(\sqrt{R} + \sqrt{2\log(4/\delta)}\right)^{1/2}}{\sigma_{\min}^2(\boldsymbol{Q}^\top \odot \widehat{\boldsymbol{A}}) N^{1/4}}\right), \tag{14}$$

where $\sigma_{\min}^2(\boldsymbol{Q}^\top \odot \widehat{\boldsymbol{A}})$ denotes the smallest singular value of the matrix $\boldsymbol{Q}^\top \odot \widehat{\boldsymbol{A}}$, which is defined in the linear system (10)*.*

The detailed proof is relegated to Appendix D. The proof consists of three steps: 1) Treat the loss function as a supervised regression problem and estimate the "generalization error" using the empirical error and Rademacher complexity; 2) Approximate the linear system in (10) using the generalization error and numerical differentiation; 3) Bound and characterize the second-order derivatives based on the approximated linear system.

Note that Theorem 2 implies that the second-order derivatives of $\widehat{h}_m(\cdot)$'s approach zero with sufficiently large sample size N. However, note that the ν term is not determined by N and it only decreases to zero when $\mathcal{F}$ is a universal function approximator (which could be attained if R is large enough). Therefore, there is always a trade-off between the complexity (encoded by key parameters like depth and width) of the neural network and the effectiveness of model identification given fixed sample size N. According to Theorem 2, one hopes to employ an expressive enough neural network for canceling $g_m(\cdot)$, but excessively increasing the expressiveness would hurt the performance under a given N—this is consistent with our experience in many neural network learning problems.

6 Optimization Design

In this section, we will design an optimization problem to implement the criterion in (11) Specifically, we propose to the following formulation:

$$\min_{\boldsymbol{Q},\boldsymbol{f}} \ \frac{1}{N}\sum_{\ell=1}^{N} \left\|\boldsymbol{Q}^\top \boldsymbol{f}(\boldsymbol{x}_\ell)\right\|_2^2 \tag{15a}$$

$$\text{s.t.} \quad \boldsymbol{Q}^\top \boldsymbol{Q} = \boldsymbol{I}, \ f_m(\cdot) \in \mathcal{F}, \ f_m(\cdot) \text{ is invertible.} \tag{15b}$$

Note that the formulation in (15) does not ensure a dense $\boldsymbol{Q}$. Nonetheless, if $\boldsymbol{Q}$ is initialized randomly using any continuous distribution, iterative optimizers would never return a $\boldsymbol{Q}$ that contains zeros, if the iterative algorithm can be approximated by a continuous transformation. We use the neural network structure defined in Assumption 3 to approximate $f_m(\cdot)$. Note that with sufficiently large R, the function $f_m(\cdot)$ is a universal approximator [36, 37]. In addition, we introduce another neural network $\boldsymbol{r}(\cdot)$, with the same structure as that of $\boldsymbol{f}(\cdot)$, to make sure that $\boldsymbol{f}(\boldsymbol{x}_\ell)$ can be converted back to $\boldsymbol{x}_\ell$. This idea is from autoencoder [38] and it is a sufficient condition to enforce $\boldsymbol{f}(\cdot)$ to be invertible. The problem is then reformulated as follows

$$\min_{\boldsymbol{Q},\boldsymbol{u}_\ell,\boldsymbol{f},\boldsymbol{r}} \ \underbrace{\frac{1}{N}\sum_{\ell=1}^{N} \left\|\boldsymbol{Q}^\top \boldsymbol{f}(\boldsymbol{x}_\ell)\right\|_2^2}_{\mathcal{L}_1} + \lambda \underbrace{\frac{1}{N}\sum_{\ell=1}^{N} \left\|\boldsymbol{x}_\ell - \boldsymbol{r}(\boldsymbol{f}(\boldsymbol{x}_\ell))\right\|_2^2}_{\mathcal{L}_2} \tag{16a}$$

$$\text{s.t.} \ \boldsymbol{Q}^\top \boldsymbol{Q} = \boldsymbol{I}, \tag{16b}$$

where the hyperparameter λ is used for balancing the energy between the loss and the regularizer term. In addition, we define $\mathcal{L} = \mathcal{L}_1 + \lambda \mathcal{L}_2$.

The problem can be handled by a block coordinate descent (BCD) algorithm as shown in Algorithm 1. Note that any stochastic optimizer designed for neural network training (e.g., [27]) can be used here for the $\boldsymbol{f}(\cdot)$ and $\boldsymbol{r}(\cdot)$ updates. We use $\boldsymbol{\theta}_f$ and $\boldsymbol{\theta}_r$ to denote the parameters of the corresponding neural networks. Besides, we denote $\widehat{\mathcal{L}}$ and $\widehat{\mathcal{L}}_i$ for $i = 1, 2$ as an estimation of $\mathcal{L}$ and $\mathcal{L}_i$ over a random

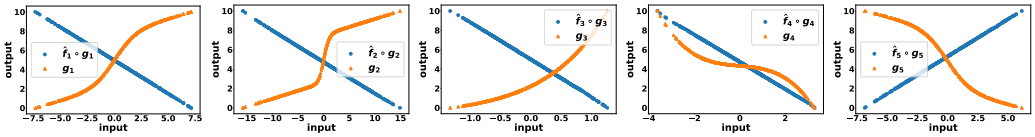

Figure 1: Learned composite functions with independent components.

batch $\mathcal{B}$, respectively. For example, $\widehat{\mathcal{L}}_1 = \frac{1}{|\mathcal{B}|}\sum_{\ell\in\mathcal{B}} \|\boldsymbol{Q}^\top \boldsymbol{f}(\boldsymbol{x}_\ell)\|_2^2$, and $\widehat{\mathcal{L}}_2$'s is defined in an identical way. The term $\nabla_{\boldsymbol{\theta}_r}\widehat{\mathcal{L}}_2$ is defined in the same way, with the derivative taken w.r.t. the parameters of the $\boldsymbol{r}$ networks.

We refer to the algorithm as *post-nonlinear subspace identification*. This is because Theorem 1 indicates that the method identifies $\mathrm{range}(\boldsymbol{S}^\top)$ where $\boldsymbol{S} = [\boldsymbol{s}_1, \ldots, \boldsymbol{s}_N]$ for $N \geq K$ after removing the nonlinear distortions (technically, this needs $d_m = 0$ for all m (cf. Theorem 1)—which automatically holds with our normal minimization objective).

Algorithm 1: Post-Nonlinear Subspace Identification.

Data: $\boldsymbol{x}_\ell$ for $\ell = 1, \cdots, N$
Result: $\boldsymbol{f}$

```
while stopping criterion is not reached do
    Q ← U[:, M − D : M] where UDV^⊤ = SVD(F) with F = [f(x_1), ··· , f(x_N)];
    while stopping criterion is not reached do
        Draw a random batch B;
        ∇_{θ_f} L̂ ← ∇_{θ_f} L̂_1 + λ ∇_{θ_f} L̂_2;
        ∇_{θ_r} L̂ ← ∇_{θ_r} L̂_2;
        Update f and r using ∇_{θ_f} L̂ and ∇_{θ_r} L̂ with any stochastic optimizer, respectively;
    end
end
```

7 Numerical Experiments

In this section, we showcase the effectiveness of the proposed method using numerical validations. We generate data using the model in (3). For the neural networks to represent $f_m(\cdot)$ and $r_m(\cdot)$ for all m, we use $R = 256$ with ReLU activations. Besides, we use Adam [27] optimizer with the initial learning rate being $2e^{-4}$ for the network optimization part. For the hyperparameters, we set $\lambda = 1e^{-4}$. The nonlinear functions $g_m(\cdot)$'s are selected to be variants of e^x, $\mathrm{sigmoid}(x)$ and $\tanh(\mathrm{x})$ to guarantee invertibility.

Independent Latent Components In this simulation, we make $K = 3$ and $M = 5$. Each of the latent components $\boldsymbol{s}_\ell$ is drawn from uniform distribution $U[-1, 1]$ with $N = 10000$, and the mixing matrix $\boldsymbol{A}$ is drawn from the normal distribution. Under this setting, it is obvious that $\boldsymbol{Q} \in \mathbb{R}^{5\times 2}$. We plot the composition $\widehat{h}_m(\cdot) = \widehat{f}_m \circ g_m(\cdot)$ in Fig. 7. The generative functions are in orange while the composite functions are plotted in blue. One can see that all the 5 composite functions are almost affine, which means we successfully identify and remove $\boldsymbol{g}(\cdot)$. Prior works on post-nonlinear ICA oftentimes involve computation and sample-size demanding implementations. For example, in [6], a nonparameteric density estimation step of the learned components needs to be involved *in every iteration* in order to measure and promote statistical independence. In comparison, the proposed approach is much more straightforward and easier to implement, as it does not rely on statistical independence.

Dependent Latent Components In this case, we consider the latent components drawn from the probability simplex as in (6). The mixing matrix $\boldsymbol{A}$ is still drawn from the normal distribution. Note that the latent components are dependent on each other in this case. Consequently the dimension of the null space is $M - K + 1$ because there are only $\widetilde{K} = K - 1$ free variables in $\boldsymbol{s}_\ell$. For this

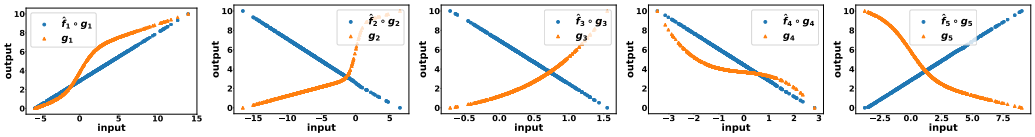

Figure 2: Learned composite functions with dependent components.

Table 1: Subspace distance with independent components.

	$R=8$	$R=16$	$R=32$	$R=64$	$R=128$	$R=256$	$R=1024$
$N=10000$	0.47±0.08	0.22±0.05	0.08±0.02	0.05±0.02	0.04±0.01	0.03±0.01	0.07±0.01
$N=20000$	0.44±0.06	0.20±0.08	0.07±0.02	0.05±0.01	0.03±0.01	0.02±0.01	0.06±0.01

simulation, we set sample size to be $N = 10000$, with $K = 3$ and $M = 5$. Accordingly, the dimension of the null space is $M - K + 1 = 3$ which makes the dimension of $\boldsymbol{Q}$ matrix 5×3. The results are plotted in Fig. 7 and it is obvious that the nonlinearity is successfully removed. Readers might have noticed that in this case the condition in (9) is actually violated since $\frac{\widetilde{K}(\widetilde{K}+1)}{2} = 3 < 5$. Interestingly, the method still works. This is likely because in this numerical example $\boldsymbol{Q}^\top$ tends to have full Kruskal rank while in our proof we only used that $\mathrm{krank}(\boldsymbol{Q}^\top) = 1$. It turns out that the identifiability condition could actually be relaxed if $\boldsymbol{Q}^\top$ has full Kruskal rank. We refer interested readers to further analyses in Appendix A.

Validating Theorem 2 In this part, we will demonstrate the impact of R (i.e., the width of the employed neural network) on the performance of nonlinearity removal, as stated in Theorem 2. To be specific, we measure the subspace distance (see the definition in [23]) between the estimated $\mathrm{range}(\widehat{\boldsymbol{S}}^\top)$ and the ground truth $\mathrm{range}(\boldsymbol{S}^\top)$, where $\boldsymbol{S} = [\boldsymbol{s}_1, \cdots, \boldsymbol{s}_N]$. This value is within [0,1] with 0 being the best. The setting is the same as that of the independent case. The results are averaged over 5 random trials and shown in Table 1. It is obvious that when the function $f_m(\cdot)$ is not expressive enough, i.e., when $R = 8$, the performance is far from satisfactory. This is because the term ν dominates in the bound (14). As the number of neurons increases, the subspace distance keeps decreasing, which indicates that the latent subspace is well identified. However, if the function is overly complex ($R = 1024$), the performance starts to deteriorate since in this case the R becomes dominant on the right hand side of (14). In addition, one can see that larger N leads to a better performance in general, which also validates our claim in (14).

8 Conclusion

In this work, we revisited the PNL mixture model from a model identification perspective. In particular, we proposed a new framework to identify and remove the unknown nonlinear distortions in the data generating/acquisition process. Compared to existing works, the proposed approach does not require the latent components to have strong properties or prior knowledge, e.g., statistical independence or probability simplex structures. Consequently, our new result could offer a PNL model identification tool to a much wider spectrum of applications, where the latent components do not exhibit structural constraints used in prior works. In addition, we provide analysis of the cases where only finite samples and non-universal function learners are available—as in practice. Beyond that, we design a constrained optimization scheme to implement the proposed learning criterion with a block coordinate descent algorithm. Our theoretical claims are supported by the experiment results.

There are a number of limitations noticed during the process of developing our subspace-based learning method. First, the PNL model would incur high computational cost when the data feature dimension M is large, since each dimension is modeled with an individual neural network. Second, there are meaningful applications where the $\boldsymbol{A}$ matrix may not have a nontrivial orthogonal complement (e.g., in PNL mixture based causal direction estimation [10]), yet this current framework does not guarantee nonlinearity removal in these cases. Third, the PNL problem's learning criterion is a nonconvex optimization problem. The proposed algorithm does not guarantee to solve the formulated learning criterion, which creates a gap between the identifiable results (by assuming optimization optimality) and the practically attainable results.

References

[1] P. Comon, "Independent component analysis, a new concept?" *Signal Processing*, vol. 36, no. 3, pp. 287 – 314, 1994.

[2] P. Paatero and U. Tapper, "Positive matrix factorization: A non-negative factor model with optimal utilization of error estimates of data values," *Environmetrics*, vol. 5, no. 2, pp. 111–126, 1994.

[3] D. Lee and H. Seung, "Learning the parts of objects by non-negative matrix factorization," *Nature*, vol. 401, no. 6755, pp. 788–791, 1999.

[4] L. D. Lathauwer and J. Castaing, "Blind identification of underdetermined mixtures by simultaneous matrix diagonalization," *IEEE Trans. Signal Process.*, vol. 56, no. 3, pp. 1096 –1105, Mar. 2008.

[5] S. Cruces, "Bounded component analysis of linear mixtures: A criterion of minimum convex perimeter," *IEEE Trans. Signal Process.*, vol. 58, no. 4, pp. 2141–2154, April 2010.

[6] A. Taleb and C. Jutten, "Source separation in post-nonlinear mixtures," *IEEE Trans. Signal Process.*, vol. 47, no. 10, pp. 2807–2820, 1999.

[7] A. Ziehe, M. Kawanabe, S. Harmeling, and K.-R. Müller, "Blind separation of post-nonlinear mixtures using linearizing transformations and temporal decorrelation," *Journal of Machine Learning Research*, vol. 4, no. Dec, pp. 1319–1338, 2003.

[8] A. Hyvarinen and H. Morioka, "Nonlinear ICA of temporally dependent stationary sources," in *International Conference on Artificial Intelligence and Statistics*, vol. 54, 20–22 Apr 2017, pp. 460–469.

[9] R. S. Zimmermann, Y. Sharma, S. Schneider, M. Bethge, and W. Brendel, "Contrastive learning inverts the data generating process," in *Proceedings of the 38th International Conference on Machine Learning*, ser. Proceedings of Machine Learning Research, vol. 139. PMLR, 18–24 Jul 2021, pp. 12 979–12 990.

[10] K. Zhang and A. Hyvarinen, "On the identifiability of the post-nonlinear causal model," *arXiv preprint arXiv:1205.2599*, 2012.

[11] A. Hyvarinen and H. Morioka, "Unsupervised feature extraction by time-contrastive learning and nonlinear ICA," in *Advances in Neural Information Processing Systems*, vol. 29, 2016.

[12] A. Hyvarinen, H. Sasaki, and R. Turner, "Nonlinear ICA using auxiliary variables and generalized contrastive learning," in *International Conference on Artificial Intelligence and Statistics*, 2019, pp. 859–868.

[13] A. Hyvarinen, "Fast and robust fixed-point algorithms for independent component analysis," *IEEE Trans. Neural Netw.,*, vol. 10, no. 3, pp. 626–634, 1999.

[14] I. Khemakhem, D. Kingma, R. Monti, and A. Hyvarinen, "Variational autoencoders and nonlinear ICA: A unifying framework," in *International Conference on Artificial Intelligence and Statistics*. PMLR, 2020, pp. 2207–2217.

[15] L. Gresele, P. K. Rubenstein, A. Mehrjou, F. Locatello, and B. Schölkopf, "The incomplete Rosetta stone problem: Identifiability results for multi-view nonlinear ICA," in *Proceedings of UAI 2020*, 2020, pp. 217–227.

[16] S. Achard and C. Jutten, "Identifiability of post-nonlinear mixtures," *IEEE Signal Process. Lett.*, vol. 12, no. 5, pp. 423–426, 2005.

[17] Y. Altmann, A. Halimi, N. Dobigeon, and J.-Y. Tourneret, "A post nonlinear mixing model for hyperspectral images unmixing," in *2011 IEEE International Geoscience and Remote Sensing Symposium*, 2011, pp. 1882–1885.

[18] F. Oveisi, "EEG signal classification using nonlinear Independent Component Analysis," in *Proc. IEEE ICASSP 2009*, April 2009, pp. 361–364.

[19] A. Ziehe, K. . Muller, G. Nolte, B. . Mackert, and G. Curio, "Artifact reduction in magnetoneurography based on time-delayed second-order correlations," *IEEE Trans. Biomedical Eng.*, vol. 47, no. 1, pp. 75–87, Jan 2000.

[20] F. Oveisi, S. Oveisi, A. Efranian, and I. Patras, " Nonlinear Independent Component Analysis for EEG-Based Brain-Computer Interface Systems," *Independent Component Analysis for Audio and Biosignal Applications, Edited by Ganesh R. Naik*, p. 165, 2012.

[21] K. Zhang and A. Hyvärinen, "Distinguishing causes from effects using nonlinear acyclic causal models," in *Causality: Objectives and Assessment*. PMLR, 2010, pp. 157–164.

[22] K. Uemura, T. Takagi, K. Takayuki, H. Yoshida, and S. Shimizu, "A multivariate causal discovery based on post-nonlinear model," in *First Conference on Causal Learning and Reasoning*, 2022. [Online]. Available: https://openreview.net/forum?id=avZrtg416yr

[23] Q. Lyu and X. Fu, "Nonlinear multiview analysis: Identifiability and neural network-assisted implementation," *IEEE Trans. Signal Process.*, vol. 68, pp. 2697–2712, 2020.

[24] B. Yang, X. Fu, N. D. Sidiropoulos, and K. Huang, "Learning nonlinear mixtures: Identifiability and algorithm," *IEEE Trans. Signal Process.*, vol. 68, pp. 2857–2869, 2020.

[25] Q. Lyu and X. Fu, "Identifiability-guaranteed simplex-structured post-nonlinear mixture learning via autoencoder," *IEEE Trans. Signal Process.*, vol. 69, pp. 4921–4936, 2021.

[26] Q. Lyu, X. Fu, W. Wang, and S. Lu, "Understanding latent correlation-based multiview learning and self-supervision: An identifiability perspective," in *International Conference on Learning Representations*, 2022. [Online]. Available: https://openreview.net/forum?id=5FUq05QRc5b

[27] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *International Conference on Learning Representations*, 2015.

[28] P. Comon and C. Jutten, *Handbook of Blind Source Separation*. Elsevier, 2010.

[29] M. Zibulevsky and B. A. Pearlmutter, "Blind source separation by sparse decomposition in a signal dictionary," *Neural computation*, vol. 13, no. 4, pp. 863–882, 2001.

[30] X. Fu, K. Huang, B. Yang, W.-K. Ma, and N. D. Sidiropoulos, "Robust volume minimization-based matrix factorization for remote sensing and document clustering," *IEEE Trans. Signal Process.*, vol. 64, no. 23, pp. 6254–6268, Dec 2016.

[31] N. Gillis and S. Vavasis, "Fast and robust recursive algorithms for separable nonnegative matrix factorization," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 36, no. 4, pp. 698–714, April 2014.

[32] A. Hyvärinen and P. Pajunen, "Nonlinear Independent Component Analysis: Existence and uniqueness results," *Neural Networks*, vol. 12, no. 3, pp. 429–439, 1999.

[33] E. Oja, "The nonlinear PCA learning rule in Independent Component Analysis," *Neurocomputing*, vol. 17, no. 1, pp. 25–45, 1997.

[34] X. Fu, K. Huang, N. D. Sidiropoulos, and W.-K. Ma, "Nonnegative matrix factorization for signal and data analytics: Identifiability, Algorithms, and Applications," *IEEE Signal Process. Mag.*, vol. 36, no. 2, pp. 59–80, March 2019.

[35] N. D. Sidiropoulos, R. Bro, and G. B. Giannakis, "Parallel factor analysis in sensor array processing," *IEEE transactions on Signal Processing*, vol. 48, no. 8, pp. 2377–2388, 2000.

[36] G. Cybenko, "Approximation by superpositions of a sigmoidal function," *Mathematics of Control, Signals and Systems*, vol. 2, no. 4, pp. 303–314, 1989.

[37] K. Hornik, "Approximation capabilities of multilayer feedforward networks," *Neural networks*, vol. 4, no. 2, pp. 251–257, 1991.

[38] G. E. Hinton and R. S. Zemel, "Autoencoders, minimum description length and Helmholtz free energy," in *Advances in Neural Information Processing Systems*, 1994, pp. 3–10.

Checklist

1. For all authors...
   (a) Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope? [Yes]
   (b) Did you describe the limitations of your work? [Yes]
   (c) Did you discuss any potential negative societal impacts of your work? [N/A]

(d) Have you read the ethics review guidelines and ensured that your paper conforms to them? [Yes]

2. If you are including theoretical results...
   (a) Did you state the full set of assumptions of all theoretical results? [Yes]
   (b) Did you include complete proofs of all theoretical results? [Yes]
3. If you ran experiments...
   (a) Did you include the code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL)? [Yes]
   (b) Did you specify all the training details (e.g., data splits, hyperparameters, how they were chosen)? [Yes]
   (c) Did you report error bars (e.g., with respect to the random seed after running experiments multiple times)? [Yes]
   (d) Did you include the total amount of compute and the type of resources used (e.g., type of GPUs, internal cluster, or cloud provider)? [Yes]
4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets...
   (a) If your work uses existing assets, did you cite the creators? [N/A]
   (b) Did you mention the license of the assets? [N/A]
   (c) Did you include any new assets either in the supplemental material or as a URL? [N/A]
   (d) Did you discuss whether and how consent was obtained from people whose data you're using/curating? [N/A]
   (e) Did you discuss whether the data you are using/curating contains personally identifiable information or offensive content? [N/A]
5. If you used crowdsourcing or conducted research with human subjects...
   (a) Did you include the full text of instructions given to participants and screenshots, if applicable? [N/A]
   (b) Did you describe any potential participant risks, with links to Institutional Review Board (IRB) approvals, if applicable? [N/A]
   (c) Did you include the estimated hourly wage paid to participants and the total amount spent on participant compensation? [N/A]