Identifiability-Guaranteed Simplex-Structured Post-Nonlinear Mixture Learning via Autoencoder

Qi Lyu and Xiao Fu
School of Electrical Engineering and Computer Science
Oregon State University
Email: (lyuqi, xiao.fu)@oregonstate.edu

Abstract

This work focuses on the problem of unraveling nonlinearly mixed latent components in an unsupervised manner. The latent components are assumed to reside in the probability simplex, and are transformed by an unknown post-nonlinear mixing system. This problem finds various applications in signal and data analytics, e.g., nonlinear hyperspectral unmixing, image embedding, and nonlinear clustering. Linear mixture learning problems are already ill-posed, as identifiability of the target latent components is hard to establish in general. With unknown nonlinearity involved, the problem is even more challenging. Prior work offered a function equation-based formulation for provable latent component identification. However, the identifiability conditions are somewhat stringent and unrealistic. In addition, the identifiability analysis is based on the infinite sample (i.e., population) case, while the understanding for practical finite sample cases has been elusive. Moreover, the algorithm in the prior work trades model expressiveness with computational convenience, which often hinders the learning performance. Our contribution is threefold. First, new identifiability conditions are derived under largely relaxed assumptions. Second, comprehensive sample complexity results are presented—which are the first of the kind. Third, a constrained autoencoder-based algorithmic framework is proposed for implementation, which effectively circumvents the challenges in the existing algorithm. Synthetic and real experiments corroborate our theoretical analyses.

1 Introduction

Unsupervised mixture learning (UML) aims at unraveling the aggregated and entangled underlying latent components from ambient data, without using any training samples. This task is also known as *blind source separation* (BSS) and *factor analysis* in the literature [1]. UML has a long history in the signal processing and machine learning communities; see, e.g., the early seminal work of *independent component analysis* (ICA) [1]. Many important applications can be considered as a UML problem, e.g., audio/speech separation [2], EEG signal denoising [3], image representation learning [4], hyperspectral unmixing [5], and topic mining [6], just to name a few.

One of the arguably most important aspects in UML/BSS is the so-called *identifiability* problem—is it possible to identify the mixed latent components from the mixtures in an unsupervised manner? The UML problem is often ill-posed, since an arbitrary number of solutions exist in general; see, e.g., discussions in [1, 7]. To establish identifiability, one may exploit prior knowledge of the mixing process and/or the latent components. Various frameworks were proposed for unraveling *linearly* mixed latent components by exploiting their properties, e.g., statistical independence, nonnegativity, boundedness, sparsity, and simplex structure—which leads to many well-known unsupervised learning models, i.e., ICA [1], *nonnegative matrix factorization* (NMF) [7], *bounded component analysis* (BCA) [8], *sparse component analysis* (SCA) [9], and *simplex-structured matrix factorization* (SSMF) [2, 6, 10]. These structures often stem from physical meaning of their respective engineering problems. For example, the simplex structure that is of interest in this

work is well-motivated in applications such as topic mining, hyperspectral unmixing, community detection, crowdsourced data labeling, and image data representation learning [6, 10–13].

Identifiability research of linear mixtures is relatively well established. However, unknown nonlinear distortions happen ubiquitously in practice; see examples in hyperspectral imaging, audio processing, wireless communications, and brain imaging [14–16]. Naturally, establishing latent component identifiability in the presence of unknown nonlinear transformations is much more challenging relative to classic linear UML cases. Early works tackled the identifiability problem from a nonlinear ICA (nICA) viewpoint. In [17], Hyvärinen *et al.* showed that in general, even strong assumptions like statistical independence of the latent components are not sufficient to establish identifiability of the nonlinear mixture model. In [18, 19], the structure of the nonlinear distortions were used to resolve the problem, leading to the so-called *post-nonlinear mixture* (PNM) ICA framework. In recent years, nICA has attracted renewed attention due to its connection to unsupervised deep learning [20, 21].

Beyond ICA, other classic UML models (e.g., NMF and SSMF) have rarely been extended to the nonlinear regime. Considering latent component properties without assuming statistical independence is of great interest, since this condition among the latent components is not a mild assumption. Without resorting to statistical independence, some works assumed the nonlinear distortions fall into certain *known categories*, e.g., bi-linear, linear-quadratic, and polynomial distortion functions in [22–24], respectively. Nonetheless, identifiability study of UML under *unknown* nonlinear distortions beyond nICA has been largely elusive. A couple of exceptions include the nonlinear multiview analysis model in [25] and a *simplex-constrained post-nonlinear mixture* (SC-PNM) model in [26]. In particular, Yang *et al.* offered a model identification criterion and showed that the unknown nonlinear distortions can be provably removed [26]—under the assumption that the latent components are generated from the probability simplex. Yang *et al.*'s work offered an approachable angle for learning the underlying components in the SC-PNM model. However, there are a number of challenges in theory and practice. First, the model identifiability condition in [26] is stringent—and some key conditions are neither enforceable nor checkable. Second, the identifiability analysis in [26] (and in all nonlinear ICA works such as those in [18–21]) are based on the assumption that infinite data samples are available (i.e., the population case). Performance analysis under the finite sample case has been elusive, yet is of great interest. Third, the implementation in [26] uses a positive neural network to model the target nonlinear transformations. Using such a special neural network is a compromise between computational convenience and model expressiveness, which loses generality and often fails to produce sensible results in practice.

Contributions. This work advances the understanding to the SC-PNM model learning problem in both theory and implementation. Our detailed contributions are as follows:

• **Deepened Identifiability Analysis.** In terms of identifiability theory, we offer a set of new sufficient conditions under which the unknown nonlinear transformations in the SC-PNM model can be provably removed. Unlike the conditions in [26], our conditions do not involve unrealistic non-enforceable conditions. A number of other stringent conditions in [26], e.g., nonnegativity of the mixing system, are also relaxed.

• **Finite Sample Analysis.** Beyond identifiability analysis under infinite data, we also consider performance characterization under the finite sample case. Leveraging a link between our learning criterion and classic generalization theories, we show that the unknown nonlinear transformations can be removed up to some 'residue' (measured by a certain metric) in the order of $O(N^{-\frac{1}{4}})$, where N is the number of samples. To our best knowledge, this is the first finite sample performance analysis for nonlinear UML problems.

• **Neural Autoencoder-Based Algorithm.** We propose a carefully constructed constrained neural autoencoder for implementing the learning criterion. Unlike the implementation in [26], our design retains the expressiveness of the neural networks when enforcing constraints in the learning criterion. We show that finding a set of feasible points of our formulation provably removes the unknown nonlinear distortions in SC-PNM models. We offer a pragmatic and effective Lagrangian multiplier-based algorithmic framework for tackling the autoencoder learning problem. Our framework can easily incorporate popular neural network optimizers (e.g., `Adam` [27]), and thus exhibits substantially improved efficiency relative to the Gauss-Newton method in [26]. We validate the theoretical claims and algorithm on various simulated and real datasets.

Part of the work was accepted by EUSIPCO 2020 [28]. The journal version additionally includes the detailed identifiability analysis, the newly derived finite sample analyses, the constrained autoencoder based implementation, the characterization of its feasible solutions, the Lagrangian multiplier algorithm, and a number of new real data experiments.

Notation. We will follow the established conventions in signal processing. To be specific, $x, \boldsymbol{x}, \boldsymbol{X}$ represent a scalar, vector, and matrix, respectively; $\text{Diag}(\boldsymbol{x})$ denotes a diagonal matrix with $\boldsymbol{x}$ as its diagonal elements; $\|\boldsymbol{x}\|_1$, $\|\boldsymbol{x}\|_2$ and $\|\boldsymbol{x}\|_\infty$ denote the ℓ_1 norm, the Euclidean norm and the infinity norm, respectively; $^\top$, † and $^\perp$ denote the transpose, Moore-Penrose pseudo-inverse operations and orthogonal complement, respectively; $\boldsymbol{P}_{\boldsymbol{X}}$ denotes the orthogonal projector onto the range space of $\boldsymbol{X}$; $\circledast$ denotes the Hadamard product; the shorthand notation 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 operation; $\mathbf{1}$ denotes an all-one vector with a proper length; $[K]$ denotes the integer set $\{1, 2, \ldots, K\}$; $\sigma_{\min}(\boldsymbol{X})$ denotes the smallest singular value of matrix $\boldsymbol{X}$; $\mathbb{E}[\cdot]$ and $\mathbb{V}[\cdot]$ denote expectation and variance of its argument, respectively; $\text{int}\mathcal{X}$ means the interior of the set $\mathcal{X}$.

2 Background

In this section, we briefly introduce the pertinent background of this work.

2.1 Simplex-Constrained Linear Mixture Model

The *simplex-constrained linear mixture model* (SC-LMM) often arises in signal and data analytics. Consider a set of acquired signal/data samples $\boldsymbol{x}_\ell \in \mathbb{R}^M$ for $\ell = 1, \ldots, N$. Under the ideal noiseless LMM, we have

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

where $\boldsymbol{A} \in \mathbb{R}^{M \times K}$ is referred to as the mixing system, $\boldsymbol{s}_\ell = [s_{1,\ell}, \ldots, s_{K,\ell}]^\top \in \mathbb{R}^K$ is a vector that holds the latent components $s_{1,\ell}, \ldots, s_{K,\ell}$, and ℓ is the sample index. In SC-LMM, it is assumed that

$$\boldsymbol{s}_\ell \in \boldsymbol{\Delta}_K,\ \boldsymbol{\Delta}_K = \{\boldsymbol{s} \in \mathbb{R}^K | \mathbf{1}^\top \boldsymbol{s} = 1, \boldsymbol{s} \geq \mathbf{0}\}; \tag{2}$$

i.e., the latent component vector $\boldsymbol{s}_\ell$ resides in the probability simplex. In general, if $\boldsymbol{x}_\ell$ can be associated with different clusters (whose centroids are represented by $\boldsymbol{a}_k$'s) with probability or weight $s_{k,\ell}$, the SC-LMM is considered reasonable. For example, in hyperspectral unmixing (HU) [5], $\boldsymbol{x}_\ell$ is a hyperspectral pixel, $\boldsymbol{A} = [\boldsymbol{a}_1, \ldots, \boldsymbol{a}_K]$ collects K spectral signatures of materials contained in the pixel, and $s_{1,\ell}, \ldots, s_{K,\ell}$ are the abundances of the materials. Many other applications can be approximated by SC-LMM, e.g., image representation learning [12], community detection [11], topic modeling [6], soft data clustering [6], just to name a few; see an illustration in Fig. 1.

LMM learning algorithms aim at recovering both $\boldsymbol{A}$ and $\boldsymbol{s}_\ell$. Note that this is usually an ill-posed problem: Given $\boldsymbol{x}_\ell$, the LMM representation is non-unique. Specifically, consider an arbitrary nonsingular $\boldsymbol{Q} \in \mathbb{R}^{K \times K}$, one can always represent $\boldsymbol{x}_\ell$ as $\boldsymbol{x}_\ell = \boldsymbol{A}\boldsymbol{Q}\boldsymbol{Q}^{-1}\boldsymbol{s}_\ell = \widetilde{\boldsymbol{A}}\widetilde{\boldsymbol{s}}_\ell$, where $\widetilde{\boldsymbol{A}} = \boldsymbol{A}\boldsymbol{Q}$ and $\widetilde{\boldsymbol{s}}_\ell = \boldsymbol{Q}^{-1}\boldsymbol{s}_\ell$—making the latent components not uniquely identifiable. Establishing identifiability of $\boldsymbol{A}$ and $\boldsymbol{s}_\ell$ is often an art—different signal structures or prior knowledge has to be exploited to serve the purpose. As mentioned, ICA [1], NMF [7], and SCA [9] exploit statistical independence, nonnegativity, and sparsity of the latent components in $\boldsymbol{s}_\ell$, respectively, to come up with identifiability-guaranteed LMM learning approaches. Identifiability of SC-LMM has also been extensively studied; see [5,7].

2.2 Post-Nonlinear Mixture Model Learning

Despite of the popularity, LMMs are considered over-simplified in terms of faithfully capturing the complex nature of real-world signals and data. In particular, *unknown* nonlinear effects are widely observed in

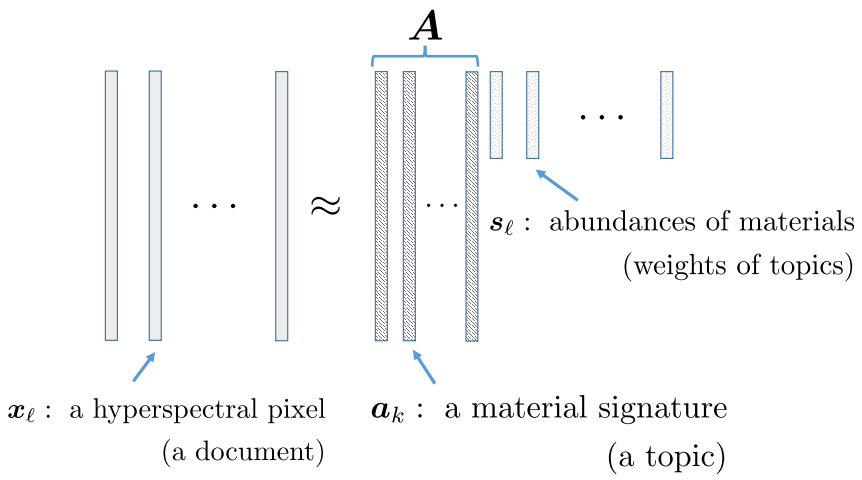

Figure 1: SC-LMM and its applications in hyperspectral imaging and topic mining; adapted from [6].

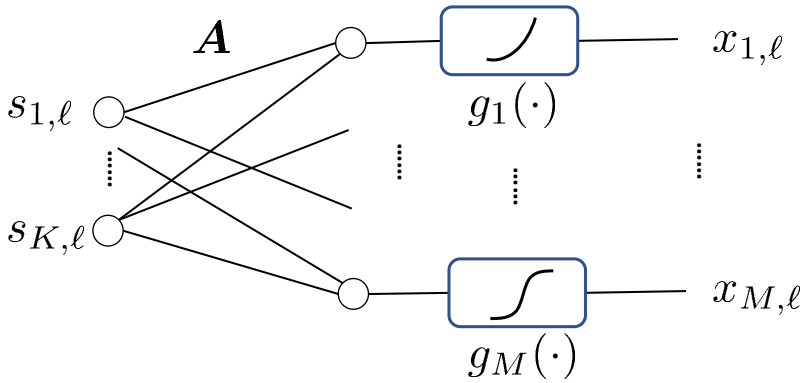

Figure 2: Illustration of PNM, where $\boldsymbol{A}$, $\boldsymbol{s}_\ell$ and $g_m(\cdot)$ are all unknown.

different domains such as hyperspectral imaging [14] and brain signals processing [25,29]. If not accounted for, nonlinear effects could severely degrade the performance of latent component identification.

To take the nonlinear distortions into consideration, one way is to employ the so-called *post-nonlinear mixture* (PNM) model [18,19,25,30,31]. Under PNM, the data model is expressed as follows:

$$x_{m,\ell} = g_m\left(\sum_{k=1}^{K} \boldsymbol{a}_k s_{k,\ell}\right), \quad \ell = 1, 2, \ldots, N, \tag{3}$$

where $g_m(\cdot) : \mathbb{R} \to \mathbb{R}$ is a scalar-to-scalar unknown nonlinear continuous invertible function; see Fig. 2. The model can also be expressed as $\boldsymbol{x}_\ell = \boldsymbol{g}(\boldsymbol{A}\boldsymbol{s}_\ell)$, where $\boldsymbol{g}(\cdot) = [g_1(\cdot), \ldots, g_M(\cdot)]^\top$.

The PNM model is a natural extension of LMM, and has a wide range of applications, e.g., low power/microwave wireless communications [32], chemical sensor design [33] and integrated Hall sensor arrays [34]. In principle, if the data acquisition process is believed to have unknown nonlinear distortions on the sensor end (represented by $g_m(\cdot)$), such a modeling is considered appropriate. In addition, for data analytics problems that have relied on the LMM based representation learning (e.g., image embedding [4]), using PNM may improve the generality of the model, thereby offering benefits in terms of downstream tasks' performance [25].

Identifiability of the latent components under PNM has been studied for a number of special cases.

It was shown in [18, 19] that if $\boldsymbol{s}_{k,\ell}$ for $k = 1, \ldots, K$ are statistically independent, an nICA framework can provably extract $\boldsymbol{s}_\ell$ up to certain ambiguities. In fact, the vast majority of PNM learning works were developed under the nICA framework; see, e.g., [18, 19, 30, 31]. We should mention that there are other nonlinear mixture models in the literature beyond PNM; see [21,35], which are also developed as extensions of ICA.

The statistical independence assumption is considered stringent in many applications (e.g., in hyperspectral unmixing [36] or any application where the simplex constraint in (2) holds). New PNM learning frameworks have been proposed to circumvent this assumption. The work in [25] uses multiple 'views' of the data entities (e.g., audio and image representation of the same entity 'cat') to remove unknown nonlinear distortions without using statistical independence. When only one view of the data is available, the work in [26] studies the model identifiability of the PNM under the simplex constraint, i.e.,

$$\boldsymbol{x}_\ell = \boldsymbol{g}(\boldsymbol{A}\boldsymbol{s}_\ell),\ \boldsymbol{s}_\ell \in \boldsymbol{\Delta}_K,\ \forall \ell \in [N], \tag{4}$$

which will be referred to as the SC-PNM model. The SC-PNM model is well-motivated, since unknown nonlinear distortions are often observed in applications such as hyperspectral imaging and image/text analysis [14], where the constraint (2) on the latent components is believed to be appropriate. Learning the SC-PNM model is our interest in this work.

2.3 Prior Work in [26]

To learn the latent components from SC-PNM, Yang *et al.* proposed to learn a nonlinear function $\boldsymbol{f} : \mathbb{R}^M \to \mathbb{R}^M$ such that it can 'cancel' or inversely identify the distortion function $\boldsymbol{g}(\cdot) = [g_1(\cdot), \ldots, g_M(\cdot)]^\top$ using the following criterion [26]:

$$\begin{aligned} & & \text{find } \boldsymbol{f} = [f_1, \ldots, f_M]^\top, & & \text{(5a)} \\ \text{(P)} & & \text{subject to } \mathbf{1}^\top \boldsymbol{f}(\boldsymbol{x}) = 1,\ \forall \boldsymbol{x} \in \mathcal{X} & & \text{(5b)} \\ & & f_m : \mathbb{R} \to \mathbb{R}:\ \text{invertible} & & \text{(5c)} \end{aligned}$$

where f_m is a scalar-to-scalar nonlinear function and

$$\mathcal{X} = \{\boldsymbol{x} \in \mathbb{R}^M | \boldsymbol{x} = \boldsymbol{g}(\boldsymbol{A}\boldsymbol{s}),\ \forall \boldsymbol{s} \in \mathbb{R}^K,\ \boldsymbol{s} \in \boldsymbol{\Delta}_K\}$$

is the domain where all data samples are generated from. In the above, the subscript 'ℓ' that is used as the sample index is eliminated since the criterion assumes the continuous domain $\mathcal{X}$ is available—and the constraint (5b) is enforced over the entire domain. In [26, Theorem 2], it was shown that when $M > K$, there always exists at least one feasible solution of (5). Here, we first show that feasible solutions also exist for the $M = K$ case under some mild conditions:

Lemma 1 (Feasibility) *Assume that $M \geq K$ and that $\boldsymbol{A}$ is drawn from any joint absolutely continuous distribution. Then, almost surely, there exists at least a feasible solution of* (5) *under the generative model in* (4)*.*

Proof: Let us first consider the $M = K$ case. By construction, let $\boldsymbol{f} = \boldsymbol{D}\boldsymbol{g}^{-1}$ where $\boldsymbol{D} = \mathrm{Diag}(\boldsymbol{\tau})$ such that $\boldsymbol{A}^\top \boldsymbol{\tau} = \mathbf{1}$. Such an $\boldsymbol{f}$ satisfies (5b). Furthermore, if $\boldsymbol{\tau}$ does not have zero elements, then $\boldsymbol{f}$ is invertible. Our proof boils down to showing that $\boldsymbol{A}^\top \boldsymbol{\tau} = \mathbf{1}$ admits a dense solution. Note that for square random matrix $\boldsymbol{A}$, we have $\boldsymbol{\tau} = \boldsymbol{A}^{-\top}\mathbf{1}$.

Since $\boldsymbol{A}$ is continuously distributed, and matrix inversion is an invertible continuous operator, $\boldsymbol{A}^{-\top}$ also follows a joint absolutely continuous distribution. Let us denote $\boldsymbol{p}_k^\top$ be the kth row of $\boldsymbol{A}^{-\top}$. Then, $\mathsf{Prob}\{\boldsymbol{p}_k^\top \mathbf{1} = 0\} = 0$ since $\boldsymbol{p}_k^\top$ follows a joint absolutely continuous distribution. Consequently, $\boldsymbol{\tau} = \boldsymbol{A}^{-\top}\mathbf{1}$ is a dense vector with probability one.

For the $M > K$ case, we show that a random $\boldsymbol{A}$ satisfies the 'incoherent' condition given in [26, Proposition 1]. To be specific, one can see that $\boldsymbol{e}_m \in \mathsf{range}(\boldsymbol{A})$ with probability zero, if $\boldsymbol{A}$ follows any joint absolutely continuous distribution. Hence, $\boldsymbol{A}^\top \boldsymbol{\tau} = \mathbf{1}$ holds with a dense $\boldsymbol{\tau}$ almost surely by invoking [26, Proposition 1]. □

Lemma 1 asserts that when $M \geq K$ (other than $M > K$ in [26]), at least one feasible solution of (5) exists. The reason why we could derive a (slightly) better lower bound of M is because the feasibility analysis in [26] is based on a worst-case argument, while we consider the generic case.

In [26], the following theorem was shown:

Theorem 1 *[26] Consider the SC-PNM model in* (4)*. Assume 1) that $\boldsymbol{A} \geq \mathbf{0}$ is tall (i.e., $M > K$), full rank, and incoherent (see definition in [26]), 2) and that the solution of* (5) *(denoted by $\widehat{f}_m$ for $m \in [M]$) makes the composition $\widehat{h}_m = \widehat{f}_m \circ g_m$ convex or concave for $m \in [M]$. Then, $\widehat{h}_m = \widehat{f}_m \circ g_m$ has to be an affine function; i.e., any f_m that is a solution of* (5) *satisfies $\widehat{\boldsymbol{h}}_m(y) = \widehat{f}_m \circ g_m(y) = c_m y + d_m,\ m \in [M]$, where $c_m \neq 0$ and d_m are constants. In addition, if $\sum_{m=1}^M d_m \neq 1$, we have*

$$\widehat{\boldsymbol{h}}(\boldsymbol{A}\boldsymbol{s}) = \widehat{\boldsymbol{f}} \circ \boldsymbol{g}(\boldsymbol{A}\boldsymbol{s}) = \widehat{\boldsymbol{A}}\boldsymbol{s},$$

where $\widehat{\boldsymbol{A}} = \boldsymbol{D}\boldsymbol{A}$ and $\boldsymbol{D}$ is a full-rank diagonal matrix; i.e., $\widehat{\boldsymbol{h}}(\boldsymbol{A}\boldsymbol{s})$ is a linear function of $\boldsymbol{A}\boldsymbol{s}$ for $\boldsymbol{s} \in \mathrm{int}\boldsymbol{\Delta}_K$.

Theorem 1 means that if (5b) can be solved with Conditions 1)-2) satisfied, the unknown nonlinear transformation $\boldsymbol{g}$ can be removed from $\boldsymbol{x} = \boldsymbol{g}(\boldsymbol{A}\boldsymbol{s})$ and $\boldsymbol{f}(\boldsymbol{x}) = \widehat{\boldsymbol{A}}\boldsymbol{s}$, where $\widehat{\boldsymbol{A}}$ is a row-scaled version of $\boldsymbol{A}$. Hence, $\boldsymbol{y} = \boldsymbol{f}(\boldsymbol{x})$ becomes SC-LMM—and identifying the latent components from SC-LMM has been well-studied (see, e.g., [5–7, 37]), as mentioned in previous sections.

2.4 Remaining Challenges

Theorem 1 has offered certain theoretical justifications for the nonlinear model learning criterion in (5). However, an array of challenges remain.

2.4.1 Stringent Conditions

Theorem 1 was derived under rather restrictive conditions. Condition 1) means that $\boldsymbol{A}$ has to be nonnegative, which may not be satisfied in many applications, e.g., blind channel equalization in wireless communications. Condition 2) is perhaps even more challenging to satisfy or check. Since $\widehat{f}_m$ is a function to be learned and g_m is unknown, the convexity/concavity of the composition $\widehat{h}_m = \widehat{f}_m \circ g_m$ is neither in control of the designers/learners nor enforceable by designing regularization/constraints.

2.4.2 Lack of Understanding For Finite Sample Cases

The proof in Theorem 1 is based on the assumption that the constraints in (5) hold over the entire $\mathcal{X}$. Note that $\mathcal{X}$ is a continuous domain, which means that an uncountably infinite number of $\boldsymbol{x}$'s exist in the domain. In practice, only a finite number of samples $\boldsymbol{x}_\ell \in \mathcal{X}$ are available. It is unclear how the learning performance scales with the number of samples. In general, finite sample analysis for nonlinear mixture learning is of great interest, but has not been addressed.

2.4.3 Implementation Issues

The implementation of (5) in [26] was based on parameterizing f_m's using positive neural networks (NNs), i.e., NNs that only admit positive network weights. This design is driven by the fact that NNs are universal function approximators. Hence, it is natural to use NNs for representing the target unknown nonlinear functions. The positivity is added to enforce the invertibility of the learned f_m's [cf. (5c)]. However, adding positivity constraints to the network weights may hinder the universal approximation capacity—which

often makes the performance unsatisfactory, as one will see in the experiments. Optimization involving positivity-constrained NN is also not trivial. The work in [26] employed a Gauss-Newton based method with dog-leg trust regions, which is not scalable.

3 Nonlinearity Removal: A Deeper Look

In this work, we start with offering a new model identification theorem under the learning criterion in (5), without using the stringent conditions in Theorem 1. Then, we will characterize the criterion under finite sample cases. We will also propose a primal-dual algorithmic framework that effectively circumvents the challenges in the positive NN-based implementation in [26].

We begin with showing that the stringent conditions in 1) and 2) of Theorem 1 can be relaxed. To proceed, we first show the following lemma:

Lemma 2 *Assume $K \geq 3$. Consider $\boldsymbol{s} = [s_1, \ldots, s_K]^\top \in \mathrm{int}\boldsymbol{\Delta}_K$. Then, $\frac{\partial s_i}{\partial s_j} = 0$ for $i \neq j$ where $i, j = 1, \ldots, K-1$.*

Proof: First, for $\boldsymbol{s}_\ell \in \mathrm{int}\boldsymbol{\Delta}_K$, we only have $K-1$ free variables, i.e., without loss of generality, s_i for $i = 1, \ldots, K-1$. Assume that $i, j \in [K-1]$ and $i \neq j$. For any fixed $\bar{s}_i$, s_j can take any possible values within a nonempty continuous domain (e.g., if $s_i = 0.5$ then the domain of s_j is $(0, 0.5)$ regardless of other components). Hence, if one treats s_i as a function of s_j, $\frac{\partial s_i}{\partial s_j} = 0$ always holds within $\boldsymbol{s} \in \mathrm{int}\boldsymbol{\Delta}_K$. □

Equipped with Lemma 1 and Lemma 2, we show our first main theorem:

Theorem 2 (Nonlinearity Removal) *Under the model in* (4)*, 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, and that the learned $\widehat{h}_m = \widehat{f}_m \circ g_m$ is twice differentiable for all $m \in [M]$. Suppose that*

$$3 \leq K \leq M \leq \frac{K(K-1)}{2}. \tag{6}$$

Then, almost surely, any $\widehat{f}_m$ that is a solution of (5) *satisfies $\widehat{h}_m(y) = \widehat{f}_m \circ g_m(y) = c_m y + d_m$, $\forall m \in [M]$, where $c_m \neq 0$ and d_m are constants. Furthermore, if $\sum_{m=1}^M d_m \neq 1$, then, almost surely, we have*

$$\widehat{\boldsymbol{h}}(\boldsymbol{A}\boldsymbol{s}) = \widehat{\boldsymbol{f}} \circ \boldsymbol{g}(\boldsymbol{A}\boldsymbol{s}) = \widehat{\boldsymbol{A}}\boldsymbol{s},$$

where $\widehat{\boldsymbol{A}} = \boldsymbol{D}\boldsymbol{A}$ and $\boldsymbol{D}$ is a full-rank diagonal matrix; i.e., $\widehat{\boldsymbol{h}}(\boldsymbol{A}\boldsymbol{s})$ is a linear function of $\boldsymbol{A}\boldsymbol{s}$ for $\boldsymbol{s} \in \mathrm{int}\boldsymbol{\Delta}_K$.

Proof: Consider $\boldsymbol{s} \in \mathrm{int}\boldsymbol{\Delta}_K$. With Lemma 2, we have $\frac{\partial s_i}{\partial s_j} = 0$ where $i, j = 1, \ldots, K-1$ and $i \neq j$. Hence, by solving problem (5), we have the following equation:

$$\sum_{i=1}^{M} \widehat{h}_i \left(a_{i,1} s_1 + a_{i,2} s_2 + \ldots + a_{i,K} \left(1 - \sum_{j=1}^{K-1} s_j \right) \right) = 1,$$

where we have used $s_K = 1 - s_1 - \ldots - s_{K-1}$. By taking second-order derivatives of both sides w.r.t s_i and s_j for $i, j \in [K-1]$, we have

$$\underbrace{\begin{bmatrix} (\boldsymbol{b}_1 \circledast \boldsymbol{b}_1)^\top \\ \cdots \\ (\boldsymbol{b}_{K-1} \circledast \boldsymbol{b}_{K-1})^\top \\ (\boldsymbol{b}_1 \circledast \boldsymbol{b}_2)^\top \\ \cdots \\ (\boldsymbol{b}_{K-2} \circledast \boldsymbol{b}_{K-1})^\top \end{bmatrix}}_{\boldsymbol{G}} \begin{bmatrix} \widehat{h}_1'' \\ \vdots \\ \widehat{h}_M'' \end{bmatrix} = \boldsymbol{0}, \tag{7}$$

where $\boldsymbol{b}_i = [a_{1,i} - a_{1,K},\ a_{2,i} - a_{2,K},\ \ldots,\ a_{M,i} - a_{M,K}]^\top$ where $i = 1, \ldots, K-1$ and $\boldsymbol{B} = [\boldsymbol{b}_1, \ldots, \boldsymbol{b}_{K-1}]$. Note that $\boldsymbol{G}$ has a size of $K(K-1)/2 \times M$.

We hope to show that $\mathrm{rank}(\boldsymbol{G}) = M$. To achieve this goal, we show that there are M rows of $\boldsymbol{G}$ that are linearly independent. This can be shown by showing that there exists a particular case such that an $M \times M$ submatrix of $\boldsymbol{G}$ has full column rank. The reason is that the determinant of any $M \times M$ submatrix of $\boldsymbol{G}$ is a polynomial of $\boldsymbol{A}$, and a polynomial is nonzero almost everywhere if it is nonzero somewhere [38].

To this end, consider a special case where $\boldsymbol{B}$ is a Vandermonde matrix, i.e., $\boldsymbol{b}_i = [1, z_i, z_i^2, \ldots, z_i^{M-1}]^\top$, and $z_i \neq z_j$. Such a $\boldsymbol{B}$ can always be constructed by letting $\boldsymbol{A}^\top$'s first $K-1$ rows to be a Vandermonde matrix and the last row to be all zeros. By picking $\widetilde{K}$ columns from the $\boldsymbol{B}$ matrix, where we require $\widetilde{K} \leq K-1$ to satisfy $M \leq \frac{\widetilde{K}(\widetilde{K}+1)}{2}$ (for simplicity, we take $M = \frac{\widetilde{K}(\widetilde{K}+1)}{2}$ for the rest of proof). Hence, the corresponding rows in the matrix of interest have the following form:

$$\begin{bmatrix} 1 & z_1^2 & \ldots & z_1^{2(M-1)} \\ \ldots & \ldots & & \ldots \\ 1 & z_{\widetilde{K}}^2 & \ldots & z_{\widetilde{K}}^{2(M-1)} \\ 1 & z_1 z_2 & \ldots & (z_1 z_2)^{M-1} \\ \ldots & \ldots & & \ldots \\ 1 & z_{\widetilde{K}-1} z_{\widetilde{K}} & \ldots & (z_{\widetilde{K}-1} z_{\widetilde{K}})^{M-1} \end{bmatrix} \tag{8}$$

Note that one can always construct such a sequence—e.g., $z_1 = 1, z_2 = 1.1, z_3 = 1.11, \ldots$ such that the matrix in (8) is full rank. This means that the linear combination of this second order homogeneous polynomials is not identically zero, which implies that it is non-zero almost everywhere [38]. Hence, the matrix $\boldsymbol{G}$ has full column rank almost surely which further means that $\widehat{h}_m'' = 0$ for all $m \in [M]$.

Note that $\widehat{h}_m'' = 0$ indicates that h_m is affine. Since f_m and g_m for all $m \in [M]$ are both invertible, $\widehat{h}_m(y) = c_m y + d_m$ with $c_m \neq 0$—otherwise, h_m is not invertible. Then, following arguments in [26, Remark 1], one can show that $\boldsymbol{h}(\boldsymbol{As})$ is linear in $\boldsymbol{As}$ if $\sum_{m=1}^M d_m \neq 1$. □

Comparing Theorems 1 and 2, one can see that the conditions in 1) and 2) in Theorem 1 are no longer needed. Relaxing the nonnegativity of $\boldsymbol{A}$ makes the method applicable to many more problems where the mixing system can have negative entries (e.g., speech separation and wireless communications). Removing Condition 2) is even more important, since this condition is not checkable or enforceable. Specifically, instead of asking for $\widehat{h}_m = \widehat{f}_m \circ g_m$ to be all convex or all concave as in Theorem 1, the conditions in Theorem 2 only need $\widehat{h}_m = \widehat{f}_m \circ g_m$ to be twice differentiable. This may be *enforced* through construction and regularization. For example, if one uses a neural network to represent the learning function $\widehat{f}_m$ (as we will do in this work), the differentiability can be promoted via bounding the network weights and using differentiable activation functions.

Notably, Theorem 2 holds if $M \leq \frac{K(K-1)}{2}$ [cf. Eq. (6)]. This, at first glance, seems uncommon. By the 'conventional wisdom' from LMM, having more 'channels' (i.e., larger M) means more degrees of freedom, and normally better performance. However, in PNM learning under the criterion in (5), having more output channels is not necessarily helpful, since more channels means that more unknown nonlinear functions $g_m(\cdot)$'s need to be learned; more precisely, more $h_m = f_m \circ g_m$'s that need to be tied to the solution of a linear system [see Eq. (7)]—and this increases the difficulty, which is a sharp contrast to the LMM case. Formally, we use the following theorem to show that $M \leq \frac{K(K-1)}{2}$ is necessary under the learning criterion (5):

Theorem 3 *Under the same model of Theorem 2, assume that $M > \frac{K(K-1)}{2}$. Then, there exist solutions f_m's that satisfy the constraints in* (5) *but $h_m = f_m \circ g_m$'s are not affine.*

Theorem 3 puts our discussions into context: In SC-PNM learning, one should avoid directly applying the criterion in (5) onto the cases where $M \gg K$. Nonetheless, this issue is more of an artifact of the criterion

used in (5), and can be easily fixed by modifying the criterion. For the $M > K$ cases, one may change the constraints in (5) to be segment by segment:

$$\mathbf{1}^\top \boldsymbol{f}\left([\boldsymbol{x}]_{(p-1)K+1:pK}\right) = 1,\ p \in [M/K], \tag{9}$$

if M/K is an integer. If M/K is not an integer, then overlap between the segments can be used.

4 Sample Complexity Analysis

The last section was developed under the assumption that the function equation $\mathbf{1}^\top \boldsymbol{f}(\boldsymbol{x}) = 1$ holds over the continuous domain $\mathcal{X}$ [cf. Eq. (5)]. In this section, we turn our attention to more practical settings where only finite samples are available. Consider the finite sample version of (5):

$$\text{find } \boldsymbol{f} = [f_1, \ldots, f_M]^\top, \tag{10a}$$

$$(\widehat{\mathsf{P}})\ \text{ subject to } \mathbf{1}^\top \boldsymbol{f}(\boldsymbol{x}_\ell) = 1,\ \forall \ell \in [N] \tag{10b}$$

$$f_m : \mathbb{R} \to \mathbb{R} :\ \text{invertible},\ f_m \in \mathcal{F},$$

where $\mathcal{F}$ is the function class where every f_m is chosen from. For example, $\mathcal{F}$ may represent classes such as polynomial functions, kernels, and neural networks. In practice, the criterion in (5) can only be carried out via tackling criterion (10). Hence, understanding key characterizations of (10), e.g., the trade-off between sample complexity and the effectiveness of nonlinearity removal, is critical.

4.1 Finite Function Class

We first show the case where $\mathcal{F}$ contains a finite number of atoms, i.e., the finite function class. To this end, we begin with the following definition and assumptions:

Definition 1 *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 $u \circ g(y) = y$, $\forall g \in \mathcal{G}$.*

Assumption 1 (Realizability) Assume that $g_m \in \mathcal{G}$, $f_m \in \mathcal{F}$, and $\mathcal{G}^{-1} \subseteq \mathcal{F}$.

Assumption 2 (Finite Class) Assume that $|\mathcal{F}| \leq d_{\mathcal{F}}$ where $d_{\mathcal{F}}$ is an upper bound of the cardinality of $\mathcal{F}$.

Assumption 3 (Boundedness) Assume that the fourth-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 $m \in [M]$. Also assume that $\boldsymbol{A}$ has bounded elements.

Assumption 1 corresponds to the 'realizable case' in statistical learning theory—i.e., a solution does exists in the function class used for learning. Assumption 2 formalizes the finite function class postulate. Assumption 3 restricts our analysis to the function class $\mathcal{F}$ and the unknown nonlinear distortions that admit bounded nth-order derivatives (with n up to 4), as a regularity condition.

Using the above assumptions, we show the following:

Theorem 4 (Finite Class Sample Complexity) *Under the generative model* (4)*, assume that Assumptions 1-3 hold, that $\boldsymbol{x}_\ell$ for $\ell \in [N]$ are i.i.d. samples from $\mathcal{X}$ according to a certain distribution $\mathcal{D}$, and that the criterion in* (10) *is solved. Denote any solution of* (10) *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 $1 - \delta$, if*

$$\gamma = \Omega\big(\big(C^2 \log(2d_{\mathcal{F}}/\delta)/NC_\phi^4\big)^{1/16}\big),$$

we have

$$\mathbb{E}\left[\left\|\widehat{\boldsymbol{h}}''(\boldsymbol{A}\boldsymbol{s})\right\|_2^2\right] = O\left(\frac{\left(\log\left(\frac{2d_{\mathcal{F}}}{\delta}\right)\right)^{1/4}}{N^{1/4}\sigma_{\min}^2(\boldsymbol{G})}\right), \tag{11}$$

for any $\boldsymbol{s} \in \boldsymbol{\Delta}_K$ such that $1-\gamma \geq s_i \geq \gamma$ for all $i \in [K]$, where $\boldsymbol{G}$ is a function of the mixing system $\boldsymbol{A}$ that is defined in Theorem 2 [cf. Eq. (7)] and $\widehat{\boldsymbol{h}}'' = [\widehat{h}_1'', \ldots, \widehat{h}_M'']^\top$, with C and C_ϕ defined in Lemmas 3 and 4, respectively.

The parameter γ confines the applicability to the interior of $\boldsymbol{\Delta}_K$—when N grows, γ approaches zero, making the applicability of the theorem gradually cover the entire $\operatorname{int}\boldsymbol{\Delta}_K$. Note that the performance metric here is the 'size' of $\widehat{\boldsymbol{h}}''$, since $\widehat{\boldsymbol{h}}'' = \boldsymbol{0}$ means that the learned composition $\widehat{h}_m = \widehat{f}_m \circ g_m$ is affine, and thus $\|\widehat{\boldsymbol{h}}''\|_2^2 \approx 0$ indicates that an approximate solution has been attained.

Theorem 4 uses the assumption that $\mathcal{F}$ is finite. Considering finite classes is meaningful, as continuous functions $f_m : \mathbb{R} \to \mathbb{R}$ are always approximated by (high-precision) discrete representations in digital systems. For example, if the function values of f_m are represented by a 64-bit double-precision floating-point format consisting of q real-valued parameters, then the number of $f_m(\cdot)$'s is from a finite class $\mathcal{F}$ consisting of at most 2^{64q} functions, which is not unrealistic considering the $\log$ operation is involved.

4.2 Neural Networks

One may also be interested in analyzing the sample complexity if specific function classes (in particular, neural networks) are used. In this subsection, we make the following assumption on $\mathcal{F}$:

Assumption 4 (Neural Network Structure) Assume that f_m is parameterized by a two-layer neural network with R neurons and z-Lipschitz nonlinear activation function $\zeta(\cdot) : \mathbb{R} \to \mathbb{R}$ with $\zeta(0) = 0$[1], i.e.,

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

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

Given these assumptions, we first show the following theorem:

Theorem 5 (Neural Network Sample Complexity) *Under the generative model (4), assume that Assumptions 1, 3 and 4 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 (10) 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 $1-\delta$, if $\gamma = \Omega((MzB^2C_x/C_\phi)^{1/4}(\sqrt{R/N} + \sqrt{\log(1/\delta)/N})^{1/8})$, it holds that*

$$\mathbb{E}\left[\left\|\widehat{\boldsymbol{h}}''(\boldsymbol{A}\boldsymbol{s})\right\|_2^2\right] = O\left(\frac{M\left(\sqrt{R} + \sqrt{\log(1/\delta)}\right)^{1/2}}{\sigma_{\min}^2(\boldsymbol{G})N^{1/4}}\right). \tag{13}$$

for any $\boldsymbol{s} \in \boldsymbol{\Delta}_K$ such that $1-\gamma \geq s_i \geq \gamma$ for all $i \in [K]$, with C_ϕ and C_x defined in Lemmas 4 and 5, respectively.

Note that by increasing R, the employed neural networks are more expressive, and thus Assumption 4 may have a better chance to be satisfied. On the other hand, increasing R makes the sample complexity increases. This balance is more articulated in the next subsection.

[1]Many popular activation functions used in modern neural networks satisfy $\zeta(0) = 0$, e.g., tanh, rectified linear unit (ReLU), centered sigmoid, Gaussian error linear unit (GELU), and exponential linear unit (ELU) [39].

4.3 In The Presence of Function Mismatches

For both Theorem 4 and Theorem 5, a key assumption is that $\mathcal{G}^{-1} \subset \mathcal{F}$ for all m (i.e., Assumption 1). In practice, since $\mathcal{F}$ is picked by the learner, and thus it is likely $g_m^{-1} \notin \mathcal{F}$. To address this case, we show the following theorem:

Theorem 6 *Assume that for any $f \in \mathcal{F}$, there exists $u \in \mathcal{G}^{-1}$ such that*

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

Then, under the same generative model of $\boldsymbol{x}$ as in Theorems 4-5, with probability of at least $1-\delta$, there exists an $\widehat{f} \in \mathcal{F}$ that violates (10b) *up to $O(M\nu)$ on average. In addition, the following holds for any $\boldsymbol{s} \in \boldsymbol{\Delta}_K$ such that $1-\gamma \geq s_i \geq \gamma$ for all $i \in [K]$:*
1) if $|\mathcal{F}| \leq d_{\mathcal{F}}$ and $\gamma = \Omega((C^2 \log(2d_{\mathcal{F}}/\delta)/NC_\phi^4)^{1/16})$, then

$$\mathbb{E}\left[\left\|\widehat{\boldsymbol{h}}''(\boldsymbol{A}\boldsymbol{s})\right\|_2^2\right] = O\left(\frac{1}{\sigma_{\min}^2(\boldsymbol{G})}\left(\frac{\log\left(\frac{2d_{\mathcal{F}}}{\delta}\right)}{N} + M^2\nu^2\right)^{1/4}\right);$$

2) moreover, if $\mathcal{F}$ is the neural network class from Assumption 4 and $\gamma = \Omega((MzB^2C_x/C_\phi)^{1/4}(\sqrt{R/N}+\sqrt{\log(1/\delta)/N})^{1/8})$, then

$$\mathbb{E}\left[\left\|\widehat{\boldsymbol{h}}''(\boldsymbol{A}\boldsymbol{s})\right\|_2^2\right] = O\left(\frac{M\left(\sqrt{R}+\sqrt{\log(1/\delta)}\right)^{1/2}}{\sigma_{\min}^2(\boldsymbol{G})N^{1/4}} + \frac{M\nu}{\sigma_{\min}^2(\boldsymbol{G})}\right).$$

The proof is relegated to Appendix 11. Theorem 6 reveals a trade-off between the expressiveness/complexity of the picked function class $\mathcal{F}$ and the nonlinearity removal 'residue' induced by the function mismatch error ν. For example, for the neural network case, we can always decrease the error ν by increasing the width or depth of the employed neural network. However, this inevitably increases the Rademacher complexity [40] of $\mathcal{F}$. In practice, these two aspects need to be carefully balanced. In addition, note that function mismatch could be a result of the presence of noise. Hence, Theorem 6 also sheds some light on the noise robustness of the proposed approach.

We hope to remark that the conditions in Theorems 4-6 are often unknown *a priori*. This means that it may not be straightforward to leverage these results for practical purposes, e.g., neural network structure design and hyperparameter selection. Nonetheless, Theorems 4-6 are the first to explain the reason why post-nonlinear mixture model learning is viable even if only finite samples are available. This set of results also helps understand how the learning performance scales with some key problem parameters, e.g., the sample size N and the complexity of $\mathcal{G}$ that contains $\boldsymbol{g}$.

5 Algorithm Design

Theorems 4-6 assert that solving the formulation in (10) removes the nonlinear distortion $\boldsymbol{g}$. However, implementing the criterion is nontrivial. In this section, we develop an algorithm to realize the learning criterion in (10).

5.1 Challenge: Enforcing Invertibility

One particular challenge comes from the constraint that f_m is invertible, which is critical for avoiding trivial solutions, as alluded in the proof of Theorem 2; also see discussions in [26]. In [26], the implementation of

(10) is by recasting the problem into a constrained optimization problem:

$$\underset{f_m \in \mathcal{F}_+, \forall m}{\text{minimize}} \ \frac{1}{N}\sum_{\ell=1}^{N}(1-\mathbf{1}^\top \boldsymbol{f}(\boldsymbol{x}_\ell))^2, \tag{14}$$

where $\mathcal{F}_+ = \{f(x) \mid f(x) = \boldsymbol{w}_2^\top \boldsymbol{\zeta}(\boldsymbol{w}_1 x),\ \boldsymbol{w}_1, \boldsymbol{w}_2 > 0\}$. The idea of adding $\boldsymbol{w}_i > \mathbf{0}$ is to ensure that the learned f_m's are invertible—It is not hard to see that positive neural networks [cf. Eq. (12)] equipped with invertible activation functions are always invertible functions. There are a couple of challenges. First, the reason why NNs are employed for parametrizing f_m is that NNs are universal approximators. However, it is unclear if the universal approximation property can still be retained after adding the positivity constraints. Second, in terms of optimization, the reformulation in (14) is not necessarily easy. Note that many classic first-order optimization approaches, e.g., projected gradient, works over convex closed sets. However, this reformulation admits an open set constraint, which is normally handled by some more complex algorithms, e.g., the interior point methods. The work in [26] ended up employing a Gauss-Newton method with trust-region based updates for handling (14), which faces scalablity challenges.

5.2 Proposed: Neural Autoencoder Based Reformulation

We propose the following reformulation of (10):

$$(\widetilde{\mathsf{P}}) \quad \underset{\boldsymbol{\theta}}{\text{minimize}} \ \frac{1}{N}\sum_{\ell=1}^{N}\|\boldsymbol{q}\left(\boldsymbol{f}(\boldsymbol{x}_\ell)\right)-\boldsymbol{x}_\ell\|_2^2 \tag{15a}$$

$$\text{subject to } \mathbf{1}^\top \boldsymbol{f}(\boldsymbol{x}_\ell) - 1 = 0,\ \forall \ell \in [N], \tag{15b}$$

Our formulation uses a reconstruction NN $\boldsymbol{q}(\cdot) = [q_1(\cdot), \ldots, q_M(\cdot)]^\top$ where $q_m(\cdot) : \mathbb{R} \to \mathbb{R}$ together with $\boldsymbol{f}$ such that

$$f_m, q_m \in \mathcal{F} = \{f(x) : \mathbb{R} \to \mathbb{R} \mid f(x) = \boldsymbol{w}_2^\top \boldsymbol{\zeta}(\boldsymbol{w}_1 x)\},$$

and $\boldsymbol{\theta}$ in (15) collects all the network parameters from f_m, q_m for all m. Unlike the network in [26], the proposed method does not impose any constraints on $\boldsymbol{\theta}$. Instead, the invertibility is promoted by introducing the reconstruction network q_m for f_m. Note that if $q_m \circ f_m(x_m) = x_m$ for all $\boldsymbol{x}$ and all m, it means that $f_m(x)$ is invertible over $\mathcal{X}$. This simple idea allows us to work with conventional NNs (instead of positive NNs)—so that the universal function approximation property of f_m is not hindered.

A remark is that the nonlinear dimensionality reduction plus reconstruction idea is a classic NN architecture for unsupervised learning, i.e., the neural autoencoder [41]. There, $\boldsymbol{f}$ and $\boldsymbol{q}$ are often fully connected neural networks (FCN) or convolutional neural networks (CNN), and constraints are not considered. Our formulation can be understood as a specially constructed constrained autoencoder. In our case, the 'bottleneck layer output' is constrained to be sum-to-one. In addition, $\boldsymbol{f}$ and $\boldsymbol{q}$ consist of individual neural networks for their elements; see Fig. 3.

Solving the problem in (15) is hard. Nonetheless, we find that it is not really necessary to solve the formulated problem in (15) to optimality—finding a feasible solution often suffices to achieve our goal. Note that finding a feasible solution of Problem (15) is also not an easy task, but practical effective algorithms exist for this purpose, as one will see.

To proceed, we show the following proposition that connects a set of feasible points of (15) and the desired solution. To see this, let us denote $v_{N,\boldsymbol{\theta}}$ as the objective value of $(\widetilde{\mathsf{P}})$ under N samples and a solution $\boldsymbol{\theta}$. In our definition, $v_{\infty,\boldsymbol{\theta}}$ corresponds to the population case where the sample average in (15) is replaced by expectation.

Proposition 1 *Assume that the generative model in Theorem 2 holds, that $M = K$ and that a feasible point of* (15) *is attained at $\boldsymbol{\theta}$. In addition, the corresponding objective value $v_{\infty,\boldsymbol{\theta}}$ satisfies $v_{\infty,\boldsymbol{\theta}} < \sum_{k=1}^{K}\sigma_k^2$, where $\sigma_k^2 = \mathbb{V}[x_k]$ is the variance of x_k. Then, the learned $\widehat{f}_m$ for $m = 1, \ldots, M$ are invertible and $\widehat{f}_m \circ g_m$ is affine with probability one.*

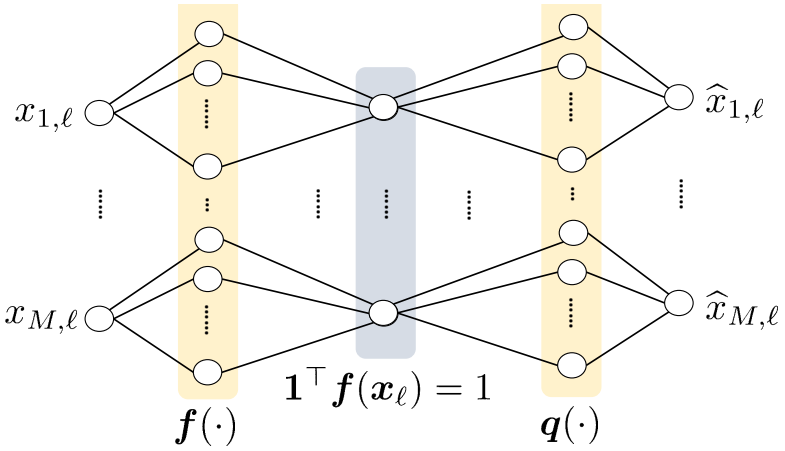

Figure 3: The 'constrained autoencoder' neural network structure of the proposed formulation.

Proof: In the population case, consider the first-order derivatives of $\phi(\boldsymbol{s}) = \mathbf{1}^\top \boldsymbol{h}(\boldsymbol{A}\boldsymbol{s}) = 1$. Putting $\frac{\partial \phi(\boldsymbol{s})}{\partial s_i}$ for $i \in [K-1]$ together, we have a system of linear equations:

$$\boldsymbol{B}^\top \boldsymbol{h}' = \mathbf{0},$$

where $\boldsymbol{h}' = [h_1', \ldots, h_K']^\top$, and $\boldsymbol{B} = [\boldsymbol{b}_1, \ldots, \boldsymbol{b}_{K-1}] \in \mathbb{R}^{K \times (K-1)}$ is defined in Theorem 2 with $\boldsymbol{b}_i = [a_{1,i} - a_{1,K},\ a_{2,i} - a_{2,K},\ \ldots,\ a_{K,i} - a_{K,K}]^\top$ where $i = 1, \ldots, K-1$.

Note that when $h_i' = 0$ for a particular i, all the other h_j' with $j \neq i$ have to be zeros. This is because any $(K-1) \times (K-1)$ submatrix of $\boldsymbol{B}$ is nonsingular with probability one, due to the assumption that the $\boldsymbol{A}$ matrix follows any joint absolutely continuous distribution; see the proof technique in Theorem 2 for showing $\boldsymbol{G}$ to be full rank. Therefore, when any $h_i' = 0$, which means h_i is not invertible, all the other h_j's are also not invertible. Thus, the h_i's are either all non-invertible or they are all invertible.

Note that $v_{\infty,\boldsymbol{\theta}}$ corresponds to the population case. Hence, following the proof of Theorem 2, one can show that $h_m'' = 0$ always holds. Therefore, if h_m is not invertible, it has to be a constant. As a result, the only possible non-invertible case is that all h_m's are constant functions. Therefore, $q_k \circ f_k \circ g_k(\cdot) = q_k \circ h_k(\cdot)$ has to be a constant, no matter what is q_k—since the input to q_k is a constant. Let us denote $c_k = q_k \circ f_k(x_k)$ as the constant. Then, we have

$$\begin{aligned}
\mathbb{E}_{\boldsymbol{x} \sim \mathcal{D}} \left[\|\boldsymbol{q}(\boldsymbol{f}(\boldsymbol{x})) - \boldsymbol{x}\|_2^2 \right] &= \mathbb{E}\left[\sum_{k=1}^K (x_k - \mu_k + \mu_k - c_k)^2 \right] \\
&= \sum_{k=1}^K \mathbb{E}\left[(x_k - \mu_k)^2 + 2(x_k - \mu_k)(\mu_k - c_k) + (\mu_k - c_k)^2 \right] \\
&= \sum_{k=1}^K \mathbb{E}\left[(x_k - \mu_k)^2 \right] + \mathbb{E}\left[(\mu_k - c_k)^2 \right] \geq \sum_{k=1}^K \sigma_k^2,
\end{aligned}$$

where $\mu_k = \mathbb{E}[x_k]$ and $\sigma_k^2 = \mathbb{V}[x_k]$ are the mean and variance of x_k, respectively. Note that the above is a necessary condition of all h_k's being zeros. Therefore, if $v_{\infty,\boldsymbol{\theta}} < \sum_{k=1}^K \sigma_k^2$ is satisfied by a feasible point $\boldsymbol{\theta}$, then all the learned functions parametrized by $\boldsymbol{\theta}$ are invertible and the composite functions $f_m \circ g_m$'s are all affine. □

Proposition 1 indicates that if one attains a feasible solution of the expectation version of Problem (10) with a sufficiently small objective value in the *population* case, then the learned $\boldsymbol{f}$ is invertible and $\boldsymbol{f} \circ \boldsymbol{g}$ is

affine. This also has practical implications: σ_k^2 can be estimated by sample variance, and the conditions in Proposition 1 can be checked when a solution $\boldsymbol{\theta}$ is obtained. Again, in practice, one can only work with finite samples (i.e., $N < \infty$). Nonetheless, the next proposition shows that when N is large enough, any feasible solution of (10) is also an approximately feasible solution of the population case. In addition, $v_{N,\boldsymbol{\theta}}$ is close to $v_{\infty,\boldsymbol{\theta}}$.

Proposition 2 *Assume that Assumptions 1 and 4 hold. Then, when*

$$N = \max\left\{ \Omega\left(\frac{M^4 z^4 B^8 C_x^4 \left(\sqrt{R} + \sqrt{\log(1/\delta)}\right)^2}{\varepsilon^2} \right),\ \ \Omega\left(\frac{M^2 z^8 B^{16} C_x^4 \left(\sqrt{R} + \sqrt{\log(1/\delta)}\right)^2}{\varepsilon^2} \right) \right\},$$

the following holds with probability of at least $1-\delta$*:*

$$\left|\mathbb{E}[\mathcal{C}_{\boldsymbol{\theta}^*}(\boldsymbol{x})]\right|^2 \leq \varepsilon,\ v_{\infty,\boldsymbol{\theta}^*} \leq v_{N,\boldsymbol{\theta}^*} + \varepsilon, \tag{16}$$

where $\mathcal{C}_{\boldsymbol{\theta}}(\boldsymbol{x}) = \mathbf{1}^\top \boldsymbol{f}(\boldsymbol{x}) - 1$ *with* $\boldsymbol{x} \in \mathcal{X}$.

The proof of Proposition 2 can be found in Appendix 16. Note that the first inequality in (16) is the basic ingredient for showing that $\mathbb{E}[\|\widehat{\boldsymbol{h}}''\|_2^2] \approx 0$ in the proofs of Theorems 4-6. Hence, (16) also implies that with a sufficient number of samples, the nonlinear distortions can be approximately removed.

5.3 Augmented Lagrangian Multiplier Method

Finding a feasible point of Problem (10) is nontrivial. Note that any Karush–Kuhn–Tucker (KKT) point is feasible. Hence, one can leverage effective KKT-point searching algorithms from nonlinear programming. To proceed, we consider the standard augmented Lagrangian $\mathcal{L}(\boldsymbol{\theta}, \boldsymbol{\lambda})$ that can be expressed as follows:

$$\frac{1}{N}\sum_{\ell=1}^{N} \mathcal{J}_{\boldsymbol{\theta}}(\boldsymbol{x}_\ell) + \frac{1}{N}\sum_{\ell=1}^{N} \lambda_\ell \mathcal{C}_{\boldsymbol{\theta}}(\boldsymbol{x}_\ell) + \frac{\rho}{2N}\sum_{\ell=1}^{N} \left|\mathcal{C}_{\boldsymbol{\theta}}(\boldsymbol{x}_\ell)\right|^2,$$

where $\mathcal{J}_{\boldsymbol{\theta}}(\boldsymbol{x}_\ell) = \|\boldsymbol{q}\left(\boldsymbol{f}(\boldsymbol{x}_\ell)\right) - \boldsymbol{x}_\ell\|_2^2$, $\mathcal{C}_{\boldsymbol{\theta}}(\boldsymbol{x}_\ell) = \mathbf{1}^\top \boldsymbol{f}(\boldsymbol{x}_\ell) - 1$ and $\boldsymbol{\lambda} = [\lambda_1, \cdots, \lambda_N]$ collects the dual variables associated with all the constraints. Here $\rho > 0$ reflects the 'importance' of the equality constraint. The algorithm updates $\boldsymbol{\theta}$ and $\boldsymbol{\lambda}$ using the following rule:

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

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

where $\kappa > 1$ is a pre-specified parameter. Note that in (17a), 'inexact minimization' means that one needs not solve the subproblem exactly. The proposed algorithm here falls into the category of augmented Lagrangian multiplier methods [42]. Applying classic convergence results for this class of algorithms, one can show that

Fact 1 *Assume that* $\mathcal{L}(\boldsymbol{\theta}, \boldsymbol{\lambda})$ *is differentiable with respect to* $\boldsymbol{\theta}$*, that each update of* $\boldsymbol{\theta}$ *in* (17a) *satisfies* $\|\nabla \mathcal{L}(\boldsymbol{\theta}^{t+1}, \boldsymbol{\lambda}^t)\|_2^2 \leq \epsilon^t$*, and that* $\epsilon^t \to 0$*. Also assume that* $\boldsymbol{\lambda}^t < \infty$ *for all* t*. Then, every limit point of the proposed algorithm is a KKT point.*

Proof: The proof is by simply invoking [42, Proposition 5.5.2]. □

Per Fact 1, the primal update for $\boldsymbol{\theta}$ should reach an ϵ-stationary point of the primal subproblem. There are many options for this purpose. In our case, the primal update involves NN-parametrized $\boldsymbol{f}$ and $\boldsymbol{q}$, and thus using gradient-based methods are natural—since back-propagation based gradient computation of

functions involving NNs is sophisticated and efficient. To further improve efficiency of the primal update, one can use *stochastic gradient descent* (SGD)-based methods that have proven effective in neural network optimization, e.g., `Adam` and `Adagrad`. In practice, increasing ρ^t to infinity may cause numerical issues, as mentioned in [42, 43]. Pragmatic remedies include increasing ρ^t in a slow rate (i.e., using $\kappa \approx 1$ so that the algorithm may converge before ρ^t becomes too large) or fixing ρ^t—which both work in practice, according to our simulations. Another remark is that the sample complexity proofs in Theorem 5 and Proposition 2 both assume using neural networks whose weights are bounded. In our implementation, we do not explicitly impose such constraints since it may complicates the computation. Nonetheless, unbounded solutions are rarely observed in our extensive experiments.

6 Numerical Results

In this section we showcase the effectiveness of the proposed approach using both synthetic and real data experiments. The proposed method is implemented in Python (with Pytorch) on a Linux workstation with 8 cores at 3.7GHz, 32GB RAM and GeForce GTX 1080 Ti GPU. The code and data will be made publicly available upon publication.

6.1 Synthetic Experiment

In the synthetic data experiments, we use the *nonlinear matrix factor recovery* (`NMFR`) method [26] as our main baseline for nonlinearity removal, since the method was developed under the same generative model in this work. We name the proposed method as *constrained neural autoencoder* (`CNAE`). We also use generic dimensionality reduction tools (e.g., `PCA`, `NMF` and the general purpose autoencoder (`AE`) [41]) as benchmarks in different experiments. Note that both `NMFR` and `AE` are neural network-based methods. The `NMFR` method uses positive NNs to parameterize f_m's. In our experiments, we use the setting following the default one in [26] (i.e., using 40-neuron one-hidden-layer and `tanh` activation functions to represent f_m) unless specified. Since `NMFR` uses a Gauss-Newton algorithm that is computationally expensive, increasing the number of neurons beyond the suggested setting in [26] is often computationally challenging. The `AE`'s encoder and decoder are represented by fully connected one-hidden-layer 256-neuron neural networks, where the neurons are ReLU functions.

For the proposed `CNAE` method, we use an individual similar network structure as in `AE`. The difference is that `CNAE` uses a single-hidden-layer fully connected network with R neurons to represent *each* of $f_m : \mathbb{R} \to \mathbb{R}$ (and also q_m) for $m \in [M]$, as opposed to the whole $\boldsymbol{f} : \mathbb{R}^M \to \mathbb{R}^K$. Since the simulations serve for proof-of-concept and theorem validation, our focus is not testing multiple neural network structures. Nonetheless, we find that using a single-hidden-layer network with a moderate R (e.g., 64 and 128) is generally effective under our simulation settings. For more challenging real experiments, the network architecture design may be assisted by procedures such as cross validation. We employ the `Adam` algorithm [27] for updating the primal variable $\boldsymbol{\theta}$. The initial learning rate of `Adam` is set to be 10^{-3}. The batch size is $1,000$, i.e., every iteration of `Adam` randomly samples 1,000 $\boldsymbol{x}_\ell$ to compute the gradient estimation for updating $\boldsymbol{\theta}$. We run up to 100 epochs of `Adam` for solving the primal subproblem in (17a) (an epoch means a full sweep of all the data samples, if gradient estimations are constructed by sampling without replacement). In addition, if $(1/N)\sum_{\ell=1}^N |\mathcal{C}_{\boldsymbol{\theta}}(\boldsymbol{x}_\ell)|^2$ goes below 10^{-5}, we stop the algorithm—as a feasible solution of (15) is approximately reached. In our Lagrangian multiplier method, $\rho^t = \rho = 10^2$ is used.

To verify the nonlinearity removal theorem, the theorems indicate that $\boldsymbol{F} = [\widehat{\boldsymbol{f}}(\boldsymbol{x}_1), \ldots, \widehat{\boldsymbol{f}}(\boldsymbol{x}_N)] \approx \boldsymbol{DAS}$, where $\boldsymbol{S} = [\boldsymbol{s}_1, \ldots, \boldsymbol{s}_N]$, is expected under the proposed learning criterion, where $\boldsymbol{D}$ is a full rank diagonal matrix. Hence, we adopt the *subspace distance* $\mathsf{dist}(\mathcal{S}, \widehat{\mathcal{S}}) = \|\boldsymbol{P}_{\boldsymbol{S}^\top}^{\perp}\boldsymbol{Q}_{\boldsymbol{F}^\top}^{\top}\|_2$ with $\mathcal{S} = \mathrm{range}(\boldsymbol{S}^\top)$ and $\widehat{\mathcal{S}} = \mathrm{range}(\boldsymbol{F}^\top)$ as the performance metric, where $\boldsymbol{Q}_{\boldsymbol{F}^\top}$ is the orthogonal basis of $\boldsymbol{F}^\top$ and $\boldsymbol{P}_{\boldsymbol{S}^\top}^{\perp}$ is the orthogonal projector of the orthogonal complement of $\mathrm{range}(\boldsymbol{S}^\top)$. This metric ranges within $[0, 1]$ with 0 being the best. When it comes to evaluating the performance of recovering $\boldsymbol{s}_\ell$, we also use another baseline, namely,

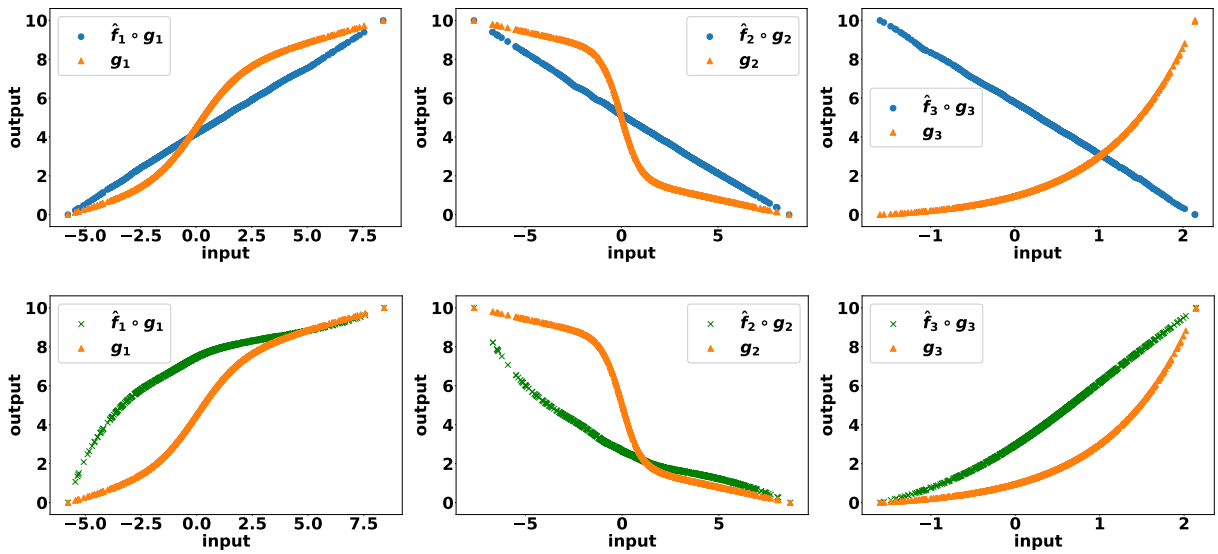

Figure 4: Learned composition functions by the proposed CNAE (top) and NMFR in [26] (bottom); both methods use 64-neuron single-hidden-layer neural networks to model the nonlinear functions.

the *minimum-volume enclosing simplex* (MVES) algorithm that provably identifies the latent components from SC-LMM under some conditions [2, 6, 37].

For the first simulation, we let $M = K = 3$. The three nonlinear distortion functions are: $g_1(x) = 5\text{sigmoid}(x) + 0.3x$, $g_2(x) = -3\tanh(x) - 0.2x$ and $g_3 = 0.4\exp(x)$, and $N = 5,000$. The matrix $\boldsymbol{A}$ is drawn from the standard Gaussian distribution.

Fig. 4 shows that the $\widehat{f}_m \circ g_m$'s learned by the proposed method are visually affine functions. In contrast, NMFR fails to remove the nonlinear distortions. This may be a bit surprising at first glance, since both methods start with the same formulation in (5). However, the implementation of NMFR is based on *positive* neural networks, which does not necessarily have the universal approximation ability—which is critical for learning complex nonlinear functions. Note that in this case, we deliberately used the same number of neurons (i.e., 64) for the forward networks (i.e., f_m's) in both methods in order to achieve better expressiveness for NMFR, despite using a number of neurons beyond the default 40 substantially slows down the algorithm of NMFR. Nonetheless, this comparison indicates that the lack of effectiveness of NMFR may be more likely the result of the positive neural network used, instead of the small number of neurons that it can afford in computation.

To quantitatively assess the effectiveness of nonlinearity removal, we show the subspace distance between the range spaces recovered latent components (i.e., $\boldsymbol{F}^\top$) and the ground truth $\boldsymbol{S}^\top$. The simulations are run under various numbers of samples including $5,000$, $10,000$ and $20,000$, respectively.

The results are shown in Table 1, which are averaged from 10 random trials. One can see that the proposed CNAE method attains the lowest subspace distances in all the cases. MVES apparently fails since it does not take nonlinear distortions into consideration. NMFR does not work well as in the first simulation, which may be again because of its positive NN-based implementation strategy. The generic NN-based unsupervised learning tool, namely, AE (which uses one-hidden-layer fully connected 256-neuron neural networks as its encoder and decoder), also does not output desired solutions. This shows the importance of exploiting structural prior information—i.e., the simplex-constrained post-nonlinear model in our case—to attain high-quality solutions when combating nonlinear distortions. In addition, one can see that the proposed method exhibits improved performance as the sample size increases, which is consistent with our analysis on sample complexity. In terms of runtime, our method improves upon NMFR by large margins in all the cases. In particular, when $N = 20,000$, the proposed method uses $1/6$ of the time that NMFR uses.

Table 1: Subspace distance between the learned ranges of $\boldsymbol{F}$ and $\boldsymbol{S}^\top$ under various N's; $M = K = 3$. Entries: dist (running time).

N	5,000	10,000	20,000
MVES	0.99 (0.35s)	0.99 (0.15s)	0.99 (0.21s)
AE	0.73 (125.9s)	0.74 (258.7s)	0.73 (505.4s)
NMFR [26]	0.69 (559.1s)	0.78 (2714.2s)	0.77 (4903.7s)
CNAE (Proposed)	0.01 (177.6s)	0.008 (305.2s)	0.005 (784.7s)

Table 2: MSE between $\boldsymbol{S}$ and estimated $\widehat{\boldsymbol{S}}$. Entries: mean (std).

N	5,000	10,000	20,000
MVES	$8.7e{-2} \pm 4.6e{-3}$	$9.2e{-2} \pm 6.8e{-3}$	$8.9e{-2} \pm 2.0e{-3}$
AE+MVES	$9.6e{-2} \pm 5.5e{-3}$	$9.7e{-2} \pm 4.4e{-3}$	$7.9e{-2} \pm 6.9e{-3}$
NMFR [26]+MVES	$1.4e{-1} \pm 3.4e{-2}$	$1.1e{-1} \pm 3.3e{-2}$	$9.2e{-2} \pm 1.2e{-2}$
CNAE (Proposed)+MVES	$4.1e{-4} \pm 3.2e{-4}$	$4.1e{-5} \pm 1.5e{-5}$	$1.4e{-5} \pm 9.9e{-6}$

This is because our framework can easily incorporate efficient NN optimizers such as Adam.

Table 2 shows the *mean squared error* (MSE) of the estimated $\widehat{\boldsymbol{S}}$, which is defined as follows:

$$\min_{\boldsymbol{\pi}\in\boldsymbol{\Pi}} \frac{1}{K}\sum_{k=1}^{K}\left\| \frac{\boldsymbol{S}_{k,:}}{\|\boldsymbol{S}_{k,:}\|_2} - \frac{\widehat{\boldsymbol{S}}_{\pi_k,:}}{\|\widehat{\boldsymbol{S}}_{\pi_k,:}\|_2}\right\|_2^2,$$

where $\boldsymbol{\Pi}$ is the set of all permutations of $\{1,\cdots,K\}$, $\boldsymbol{S}_{k,:}$ and $\widehat{\boldsymbol{S}}_{k,:}$ are the ground truth of the kth row of $\boldsymbol{S}$ and the corresponding estimate, respectively. Note that the kth row in $\boldsymbol{S}$ represents the kth latent component. The permutation matrix $\boldsymbol{\Pi}$ is used since mixture learning has an intrinsic row permutation ambiguity in the estimated $\widehat{\boldsymbol{S}}$ that cannot be removed without additional prior knowledge. The latent components are estimated by applying the SC-LMM learning algorithm MVES onto $\boldsymbol{F}$ after the nonlinearity removal methods are used to cancel $\boldsymbol{g}$. Note that the generated $\boldsymbol{s}_\ell$'s are *sufficiently scattered* (see definition in [7]) in $\boldsymbol{\Delta}_K$. Hence, $\boldsymbol{S}$ is provably identifiable (up to permutation ambiguities) from $\boldsymbol{F}$ by MVES if $\boldsymbol{F}$ is indeed a linear mixture of $\boldsymbol{S}$. The results are also averaged from 10 random trials. From the table, one can see that CNAE largely outperforms the baselines, i.e., by three orders of magnitude in terms of MSE. This also asserts that the nonlinearity removal stage performed by CNAE is successful.

Fig. 5 shows the *non-identifiability* of the proposed method when $M \leq K(K-1)/2$ is violated (cf. Theorem 3). Specifically, we set $K = 3$ and $M = 4$. In this case, $M = 4$ is larger than $\frac{K(K-1)}{2} = 3$. The first row shows the result of applying CNAE on all of the four out channels, i.e., the entire column vector $\boldsymbol{x}_\ell$ for all ℓ. Apparently, the nonlinearity of every channel is not removed, which supports our claim in Theorem 3 that $M \leq K(K-1)/2$ is not only sufficient, but also necessary if the criterion in (5) is used. However, if one only selects three out of four channels and applies CNAE onto the selected channels [cf. Eq. 9], the nonlinear distortions can be successfully removed—see the second and the third rows in Fig. 5.

To evaluate the impact of noise, we consider two different noisy models, i.e.,

$$\boldsymbol{x}_\ell = \boldsymbol{g}(\boldsymbol{A}\boldsymbol{s}_\ell) + \boldsymbol{v}_\ell,\ \boldsymbol{x}_\ell = \boldsymbol{g}(\boldsymbol{A}\boldsymbol{s}_\ell + \boldsymbol{v}_\ell),$$

where $\boldsymbol{v}_\ell$ denotes the zero-mean white Gaussian noise. We define the signal-to-noise ratio (SNR) for the two models as

$$\text{SNR}_1 = 10\log_{10}(\textstyle\sum_{\ell=1}^N \|\boldsymbol{g}(\boldsymbol{A}\boldsymbol{s}_\ell)\|_2^2 / \sum_{\ell=1}^N \|\boldsymbol{v}_\ell\|_2^2)\ \text{dB},$$
$$\text{SNR}_2 = 10\log_{10}(\textstyle\sum_{\ell=1}^N \|\boldsymbol{A}\boldsymbol{s}_\ell\|_2^2 / \sum_{\ell=1}^N \|\boldsymbol{v}_\ell\|_2^2)\ \text{dB},$$

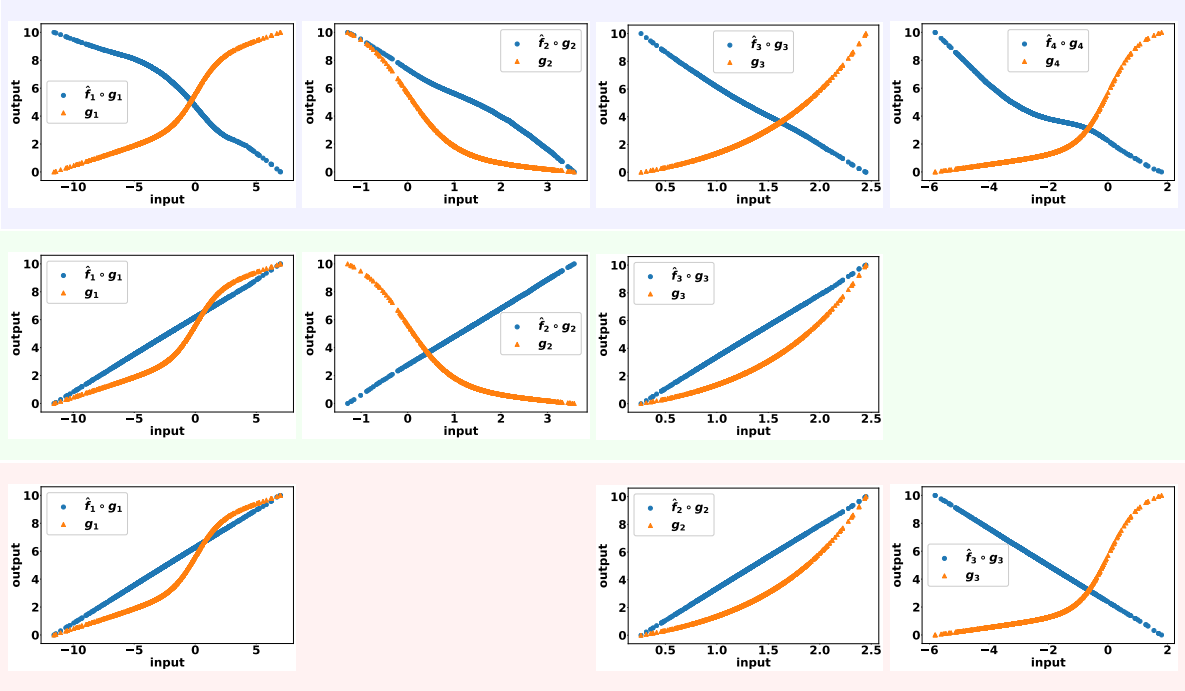

Figure 5: The results of the learned $\widehat{h}_m = \widehat{f}_m \circ g_m$'s under $(M, K) = (4, 3)$. Top: using all 4 channels; Middle: using channel x_1, x_2, and x_3; Bottom: using channel x_1, x_3, and x_4.

Table 3: Subspace distance between $\boldsymbol{F}$ and $\boldsymbol{S}^\top$ and MSE between $\boldsymbol{S}$ and $\widehat{\boldsymbol{S}}$ under various SNR_1. Entries: dist/MSE.

SNR_1	10 dB	20 dB	30 dB	40 dB
MVES	0.997/0.10	0.998/0.10	0.998/$8.9e{-2}$	0.998/0.10
AE	0.591/$6.5e{-2}$	0.510/$6.2e{-2}$	0.470/$6.3e{-2}$	0.461/$6.6e{-2}$
NMFR [26]	0.701/$8.3e{-2}$	0.682/$8.1e{-2}$	0.395/$6.1e{-2}$	0.324/$2.2e{-2}$
CNAE (Proposed)	0.510/$6.7e{-2}$	0.218/$1.8e{-2}$	0.077/$3.9e{-3}$	0.055/$4.3e{-3}$

respectively. We test the noisy cases using $N = 5,000$ samples under $K = 4$ and $M = 5$. In addition to the three nonlinear invertible functions used in the first simulation for generating data, $g_4(x) = -4\text{sigmoid}(x) - 0.2x$ and $g_5(x) = 5\tanh(x) - 0.3x$ are included. Other settings are the same as before.

Table 3 and Table 4 show the results under the two noisy models, respectively. One can see that the proposed method is robust to both noise models to a certain extent. In particular, when the SNR_1 and SNR_2 are larger than or equal to 20dB, the subspace distance between the estimated $\text{range}(\boldsymbol{F}^\top)$ and the ground truth $\text{range}(\boldsymbol{S}^\top)$ is around 0.2. This is analogous to around only 10 degrees of misalignment between two vectors. The best baseline's subspace distance is at least twice higher under the same noise level. The MSEs of the estimated $\boldsymbol{S}$ also reflect a similar performance gap.

6.2 Real-Data Experiment: Nonlinear Hyperspectral Unmixing

We consider the hyperspectral unmixing problem in the presence of nonlinearly mixed pixels. In this experiment, CNAE and NMFR use the same neural network settings as before. For AE, the encoder and decoder use identical (but mirrored) one-hidden-layer structure, where the number of neurons is 512. In addition to the baselines used in the simulations, we also consider another baseline that is specialized for hyper-

Table 4: Subspace distance between $\boldsymbol{F}$ and $\boldsymbol{S}^\top$ and MSE between $\boldsymbol{S}$ and $\widehat{\boldsymbol{S}}$ under various SNR_2. Entries: dist/MSE.

SNR_2	10 dB	20 dB	30 dB	40 dB
MVES	$0.998/0.10$	$0.997/0.11$	$0.997/0.10$	$0.998/0.10$
AE	$0.713/8.6e{-2}$	$0.501/6.2e{-2}$	$0.471/5.9e{-2}$	$0.469/6.7e{-2}$
NMFR [26]	$0.718/8.9e{-2}$	$0.438/5.3e{-2}$	$0.412/1.6e{-2}$	$0.267/6.5e{-3}$
CNAE (Proposed)	$0.482/5.4e{-2}$	$0.181/9.1e{-3}$	$0.067/1.5e{-3}$	$0.025/7.4e{-4}$

spectral unmixing, namely, *nonlinear hyperspectral unmixing based on deep autoencoder* (NHUDAE) [44]. Our implementation follows the deep network architecture recommended in [44]. Note that the autoencoder used in NHUDAE is closer to a generic one, instead of the post-nonlinear model-driven structure used in CNAE.

Our experiment uses the Moffett field data captured in California, which is known for the existence of nonlinearly distorted pixels [6, 14, 26]. The considered image has three major materials, namely, water, soil and vegetation. The image consists of 50×50 pixels. Each pixel is represented by $M = 198$ different spectral measurements. In other words, we have $\boldsymbol{x}_\ell \in \mathbb{R}^{198}$ and the sample size is $N = 2,500$. Although no ground-truth is available, there are many pixels in the image that can be visually identified as purely water. The water identity of these pixels is further verified by comparing with the previously recorded spectral signature of water in the literature [45, 46]—and this information is used for evaluation; see Fig. 6, where a purely water region is highlighted using a red rectangle.

Our evaluation strategy follows that in [26]. To be specific, we compute the $\widehat{\boldsymbol{s}}_\ell$'s in this region using the methods under test, e.g., CNAE+MVES, and observe if the $\widehat{\boldsymbol{s}}_\ell$'s are unit vectors indicating that the region only contains one material (i.e., water). Following this idea, we compute the average of $\widehat{s}_{k,\ell}$ over the pixels in the rectangle region. The ideal value should be 1 for the water and 0 for soil. The results are shown in Table 5, where the k's associated with 'water' and 'soil' are visually identified by plotting $\widehat{\boldsymbol{S}}_{k,:}$ and comparing to Fig. 6. Obviously, the SC-LMM based method MVES barely works on this data due to the existence of nonlinear distortions. All of the nonlinearity removal methods except for NHUDAE improve upon the result of MVES. In particular, both NMFR and CNAE perform better than the generic nonlinear learning tool AE, which shows that using latent component prior information in unsupervised nonlinear learning in an identifiability-guaranteed way is indeed helpful. Nevertheless, the proposed method outputs results that are the closest to the ideal values.

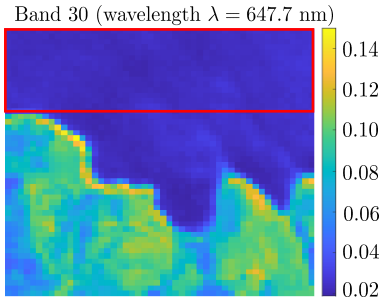

Figure 6: The Moffett field data at band 30. The rectangle region corresponds to a lake. The water abundance is 1.0 in this region, while the abundances of other materials are zeros.

Table 5: Averaged estimated weights of the highlighted area in Fig. 6 for the water and soil abundance map, respectively.

material	water abundance (true = 1.0)	soil abundance (true = 0.0)
MVES	0.657 ± 0.003	0.326 ± 0.004
AE+MVES	0.741 ± 0.145	0.202 ± 0.160
NHUDAE [44]	0.603 ± 0.032	0.165 ± 0.040
NMFR [26] +MVES	0.866 ± 0.057	0.119 ± 0.059
CNAE (Proposed)+MVES	0.943 ± 0.030	0.025 ± 0.029

Figure 7: Sample images from COIL dataset that are from 10 different classes.

6.3 Real-Data Experiment: Image Representation Learning

We also use the proposed CNAE method to serve as an unsupervised representation learner for image data. Note that the SC-LMM has been successful in image representation learning. For example, the works in [4, 12, 47] take the $\boldsymbol{x}_\ell = \boldsymbol{A}\boldsymbol{s}_\ell$ model for an image (e.g., a human face), where $\boldsymbol{a}_k$ for $k = 1, \ldots, K$ are the constituents (e.g., nose, lips, and eyes) of a collection of such images, and $s_{k,\ell}$ is the weight of the kth constituent in image ℓ; also see [7] for illustration. Our hypothesis is that using SC-PNM is at least not harmful, and may exhibits some benefits since SC-PNM is a more general model relative to SC-LMM.

To verify our hypothesis, we test the methods on the COIL dataset [48]. The COIL dataset has various classes of images, and each class admits 72 images. The images are represented by gray-level 32×32 pixels. Some sample images are shown in Fig. 7.

In our experiment, we test the algorithms using 5 and 10 classes of images, respectively. Each class is split into training/validation/test sets following a $42/15/15$ ratio. In order to show the effectiveness of different nonlinear dimensionality reduction (low-dimensional representation) methods (i.e., CNAE, NMFR and AE), we run k-means clustering algorithms followed the nonlinearity removal stage. We also use popular linear dimensionality reduction methods, i.e., NMF and PCA, to serve as our baselines. For the k-means algorithm, we run clustering on the training set and then use the learned cluster centers to label the test set. For the proposed CNAE, the number of neurons for each f_m is selected from $[32, 64, 128, 256]$ and the latent dimension is chosen from $K \in [5, 10, 20, 40, 100, 200]$. These hyperparameters are selected using the validation set.

In Table 6, the performance measures include clustering accuracy (ACC), normalized mutual information (NMI) and adjusted Rand index (ARI) [49]. ARI ranges from -1 to $+1$, with one being the best and minus one the worst. ACC and NMI range from 0 to 1 with 1 being the best. The results are averaged over 5 random selections of different classes. One can see that CNAE and AE in general outperform the baselines in terms of clustering performance. CNAE works even better compared to AE, perhaps because it enforces the sum-to-one condition in the nonlinearity reduction step that is consistent with the subsequent k-means model.

Table 6: Kmeans clustering performance on COIL data.

	5 classes			10 classes		
Methods	ACC	NMI	ARI	ACC	NMI	ARI
raw data	0.829	0.851	0.746	0.698	0.799	0.611
PCA	0.837	0.864	0.763	0.725	0.804	0.630
AE	0.840	0.855	0.759	0.724	0.803	0.622
NMFR [26]	†[1]	†	†	†	†	†
MVES	0.829	0.829	0.734	0.716	0.782	0.589
NMF	0.728	0.767	0.603	0.590	0.737	0.480
CNAE (Proposed)	0.896	0.888	0.824	0.753	0.823	0.662

[1] Out of memory or too slow.

7 Conclusions

In this work, we addressed a number of challenges of the unsupervised SC-PNM learning problem. Specifically, we first advanced the understanding to the model identifiability of SC-PNM by offering largely relaxed identifiability conditions. In addition, we extended the population case based model identification analysis to the practical finite sample scenarios, and offered a number of sample complexity analyses. To our best knowledge, this is the first set of finite sample analyses for nonlinear mixture learning problems. Furthermore, we proposed a constrained neural autoencoder based formulation to realize the learning criterion, and showed that a set of feasible solutions of this formulation provably removes the unknown nonlinearity in SC-PNM. We also offered an efficient Lagrangian multiplier based algorithmic framework to handle the optimization problem. We used synthetic and real-data experiments to show the effectiveness of our approach. Our approach is a combination of model-based mixture identification and data-driven nonlinear function learning. Our analyses may lead to better understanding for the effectiveness of neural autoencoders and more principled neural network design.

There are also a number of limitations and potential future directions. For example, under the current framework, every data dimension needs a neural autoencoder for nonlinearity removal, which may not be scalable (see Appendix 18 in the supplementary material). More judicious neural architecture design may be used to circumvent this challenge. In addition, beyond the probability simplex-structured latent signals, it is also much meaningful to consider more general latent structures. Considering nonlinear distortions beyond the post-nonlinear case is another appealing direction—which may serve a wider range of applications.

Acknowledgement

Xiao Fu wishes to dedicate this work to Prof. José Bioucas-Dias, whose work inspired him greatly in the last decade.

Appendix

8 Proof of Theorem 4

The proof is split into several steps.

A Useful Lemmas

Note that there exists a constant $C \geq 0$ such that $(1-\mathbf{1}^\top \boldsymbol{f}(\boldsymbol{x}))^2 \leq C$ for all $f_m \in \mathcal{F}$ and $\boldsymbol{x} \in \mathcal{X}$ if Assumption 3 holds. We show the following lemma:

Lemma 3 *Assume that N samples $\{\boldsymbol{x}_\ell\}_{\ell=1}^N$ are available, where $\boldsymbol{x}_\ell$'s are i.i.d. samples drawn from the domain $\mathcal{X}$ following distribution $\mathcal{D}$. Assume that Assumption 1 holds, and that we have $(1 - \mathbf{1}^\top \boldsymbol{f}(\boldsymbol{x}))^2 \leq C$ for all $\boldsymbol{x} \in \mathcal{X}$. Then, if Problem* (10) *is solved with a solution $\widehat{\boldsymbol{f}}$ and*

$$N \geq \frac{C^2 \log(2d_{\mathcal{F}}/\delta)}{2\varepsilon^2},$$

the following holds with a probability at least $1-\delta$:

$$\mathbb{E}\left[\left(1 - \mathbf{1}^\top \widehat{\boldsymbol{f}}(\boldsymbol{x})\right)^2\right] \leq \varepsilon, \quad \forall \boldsymbol{x} \in \mathcal{X}, \tag{18}$$

The proof can be found in Appendix 12. We also show the following lemma:

Lemma 4 *Assume that Assumption 3 holds. Specifically, assume that the fourth-order derivatives of learned $\widehat{f}_i$ and the nonlinear distortion g_i exist. In addition, $|f_i^{(n)}(\cdot)| \leq C_f$ and $|g_i^{(n)}(\cdot)| \leq C_g$, i.e., the nth-order derivatives are bounded for all $n \in \{1, \ldots, 4\}$. Also assume that $\boldsymbol{A}$ has bounded elements, i.e., $|\boldsymbol{A}(i,j)| \leq C_a$. Denote*

$$C_\phi = 16MC_a^4\left(C_f C_g^4 + 6C_f C_g^3 + 3C_f C_g^2 + 4C_f C_g^2 + C_f C_g\right).$$

Then, the fourth-order derivative of $\phi(\boldsymbol{s}) = \mathbf{1}^\top \boldsymbol{h}(\boldsymbol{A}\boldsymbol{s})$ w.r.t. s_i is bounded by

$$\left|\frac{\partial^4 \phi(\boldsymbol{s})}{\partial s_i^4}\right| \leq C_\phi. \tag{19}$$

In addition, the other cross-derivatives are also bounded by

$$\left|\frac{\partial^4 \phi(\boldsymbol{s})}{\partial s_i^3 \partial s_j}\right| \leq C_\phi, \quad \left|\frac{\partial^4 \phi(\boldsymbol{s})}{\partial s_i^2 \partial s_j^2}\right| \leq C_\phi. \tag{20}$$

The proof can be found in Appendix 13.

B Fitting Error Estimation

With the lemmas above, we are ready to show the theorem. Our proof is constructive. Note that the fitting error estimation at any sample $\boldsymbol{x}_\ell \sim \mathcal{D}$ can be expressed as $(1 - \mathbf{1}^\top \widehat{\boldsymbol{f}}(\boldsymbol{x}_\ell))^2 = \varepsilon_\ell$. If $\boldsymbol{x}_\ell$ is observed (i.e., $\boldsymbol{x}_\ell \in \{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_N\}$) in the 'training samples', then we have $\varepsilon_\ell = 0$, otherwise we have $\mathbb{E}[\varepsilon_\ell] \leq \varepsilon$ with high probability, as shown in Lemma 3. This is because ε_ℓ is a derived random variable of $\boldsymbol{x}_\ell$, and its expectation is, by the *fundamental theorem of expectation*, $\mathbb{E}[\varepsilon_\ell] = \mathbb{E}[(1 - \mathbf{1}^\top \widehat{\boldsymbol{f}}(\boldsymbol{x}_\ell))^2]$.

Using this notion, we will numerically 'estimate' the second-order (cross-)derivatives of $\widehat{\boldsymbol{h}} = \widehat{\boldsymbol{f}} \circ \boldsymbol{g}$.

C Estimating $\frac{\partial^2 \phi(\boldsymbol{s})}{\partial s_i^2}$

Recall that we have

$$\mathbf{1}^\top \widehat{\boldsymbol{h}}(\boldsymbol{A}\boldsymbol{s}_\ell) = 1 \pm \sqrt{\varepsilon_\ell}. \tag{21}$$

Define $\Delta \boldsymbol{s}_i = [0, \ldots, \Delta s_i, \ldots, 0, -\Delta s_i]^\top$ with

$$\Delta s_i \in \mathcal{S}_i = [0, \min\{s_{i,\ell}, 1 - s_{i,\ell}\}), \tag{22}$$

for $i = 1, \ldots, K-1$, and $\boldsymbol{s}_{\widehat{\ell}} = \boldsymbol{s}_\ell + \Delta \boldsymbol{s}_i$ and $\boldsymbol{s}_{\widetilde{\ell}} = \boldsymbol{s}_\ell - \Delta \boldsymbol{s}_i$. Note that $\boldsymbol{s}_\ell \pm \Delta \boldsymbol{s}_i \in \operatorname{int} \boldsymbol{\Delta}_K$ still holds. Therefore, we have

$$\begin{aligned} \mathbf{1}^\top \widehat{\boldsymbol{h}}(\boldsymbol{A}(\boldsymbol{s}_\ell + \Delta \boldsymbol{s}_i)) &= 1 \pm \sqrt{\varepsilon_{\widehat{\ell}}}, \\ \mathbf{1}^\top \widehat{\boldsymbol{h}}(\boldsymbol{A}(\boldsymbol{s}_\ell - \Delta \boldsymbol{s}_i)) &= 1 \pm \sqrt{\varepsilon_{\widetilde{\ell}}}, \end{aligned} \tag{23}$$

where $\mathbb{E}[\varepsilon_{\widehat{\ell}}] \leq \varepsilon$ and $\mathbb{E}[\varepsilon_{\widetilde{\ell}}] \leq \varepsilon$ hold with high probability when N is large.

For any continuous function $\omega(z)$ that admits non-vanishing 4th order derivatives, the second order derivative at z can be estimated as follows [50]:

$$\omega''(z) = \frac{\omega(z + \Delta z) - 2\omega(z) + \omega(z - \Delta z)}{\Delta z^2} - \frac{\Delta z^2}{12} \omega^{(4)}(\xi),$$

where $\xi \in (z - \Delta z, z + \Delta z)$.

Following this definition, one can see that

$$\frac{\partial^2 \phi(\boldsymbol{s})}{\partial s_i^2} = \frac{\pm\sqrt{\varepsilon_{\widehat{\ell}}} \mp 2\sqrt{\varepsilon_\ell} \pm \sqrt{\varepsilon_{\widetilde{\ell}}}}{\Delta s_i^2} - \frac{\Delta s_i^2}{12} \phi^{(4)}(\boldsymbol{\xi}_i),$$

where $\boldsymbol{\xi}_i \in (\boldsymbol{s}_{\widetilde{\ell}}, \boldsymbol{s}_{\widehat{\ell}})$. Consequently, we have the following inequalities:

$$\begin{aligned} \left| \sum_{m=1}^M (a_{m,i} - a_{m,K})^2 \widehat{h}_m''(\boldsymbol{A}\boldsymbol{s}_\ell) \right| &= \left| \frac{\pm\sqrt{\varepsilon_{\widehat{\ell}}} \mp 2\sqrt{\varepsilon_\ell} \pm \sqrt{\varepsilon_{\widetilde{\ell}}}}{\Delta s_i^2} - \frac{\Delta s_i^2}{12} \phi^{(4)}(\boldsymbol{\xi}_i) \right| \\ &\leq \frac{\sqrt{\varepsilon_{\widehat{\ell}}} + 2\sqrt{\varepsilon_\ell} + \sqrt{\varepsilon_{\widetilde{\ell}}}}{\Delta s_i^2} + \frac{\Delta s_i^2}{12} \left| \phi^{(4)}(\boldsymbol{\xi}_i) \right|. \end{aligned}$$

By taking expectation, we have the following holds with probability at least $1 - \delta$:

$$\begin{aligned} \mathbb{E}\left[\left| \sum_{m=1}^M (a_{m,i} - a_{m,K})^2 \widehat{h}_m''(\boldsymbol{A}\boldsymbol{s}_\ell) \right| \right] &\leq \frac{\mathbb{E}[\sqrt{\varepsilon_{\widehat{\ell}}}] + 2\mathbb{E}[\sqrt{\varepsilon_\ell}] + \mathbb{E}[\sqrt{\varepsilon_{\widetilde{\ell}}}]}{\Delta s_i^2} + \frac{\Delta s_i^2}{12} \left| \phi^{(4)}(\boldsymbol{\xi}_i) \right| \\ &\leq \frac{4\sqrt{\varepsilon}}{\Delta s_i^2} + \frac{\left| \phi^{(4)}(\boldsymbol{\xi}_i) \right| \Delta s_i^2}{12}, \end{aligned} \tag{24}$$

where the second inequality is by the Jensen's inequality

$$\mathbb{E}[\sqrt{\varepsilon_\ell}] \leq \sqrt{\mathbb{E}[\varepsilon_\ell]} \leq \sqrt{\varepsilon},$$

which holds by the concavity of $\sqrt{x}$ when $x \geq 0$.

Note that the bound in (24) holds for all Δs_i that satisfy (22). We are interested in finding the best upper bound, i.e.,

$$\inf_{\Delta s_i \in \mathcal{S}_i} \frac{4\sqrt{\varepsilon}}{\Delta s_i^2} + \frac{\left| \phi^{(4)}(\boldsymbol{\xi}) \right|}{12} \Delta s_i^2. \tag{25}$$

Note that the function in (25) is convex and smooth when $\Delta s_i \in \mathcal{S}_i$. Hence, it is straightforward to show that the infimum is attained by either the minimizer of the convex function or the boundary of $\mathcal{S}_i$, i.e.,:

$$\Delta s_i^\star \in \left\{ \left(\frac{48\sqrt{\varepsilon}}{\left| \phi^{(4)}(\boldsymbol{\xi}_i) \right|} \right)^{1/4}, \min\{s_{i,\ell}, 1 - s_{i,\ell}\} \right\},$$

which gives the minimum as follows:

$$\inf_{\Delta s_i} \frac{4\sqrt{\varepsilon}}{\Delta s_i^2} + \frac{\left|\phi^{(4)}(\boldsymbol{\xi}_i)\right|}{12}\Delta s_i^2 \leq \min\left\{\frac{2\sqrt{3\left|\phi^{(4)}(\boldsymbol{\xi}_i)\right|}\varepsilon^{1/4}}{3}, \frac{4\sqrt{\varepsilon}}{\kappa_i^2} + \frac{\left|\phi^{(4)}(\boldsymbol{\xi}_i)\right|}{12}\kappa_i^2\right\}, \tag{26}$$

where $\kappa_i = \min\{s_{i,\ell}, 1 - s_{i,\ell}\}$. Note that we always have $\kappa_i \geq \gamma$. Hence, if $\left(\frac{48\sqrt{\varepsilon}}{\left|\phi^{(4)}(\boldsymbol{\xi}_i)\right|}\right)^{1/4} \leq \gamma$, we can simplify the bound. This means that, if we have $\gamma \geq \left(\frac{48\sqrt{\varepsilon}}{C_\phi}\right)^{1/4}$, then, (26) can be bounded by

$$\left|\mathbb{E}\left[\sum_{m=1}^{M}(a_{m,i} - a_{m,K})^2\widehat{h}_m''(\boldsymbol{A}\boldsymbol{s})\right]\right| \leq \frac{2\sqrt{3\left|\phi^{(4)}(\boldsymbol{\xi}_i)\right|}\varepsilon^{1/4}}{3}.$$

Given that N is fixed, one may pick $\varepsilon = \sqrt{\frac{C^2\log(2d_{\mathcal{F}}/\delta)}{2N}}$, which gives the conclusion that if $\gamma \geq \left(\frac{48}{C_\phi}\right)^{1/4}\left(\frac{C^2\log\left(\frac{2d_{\mathcal{F}}}{\delta}\right)}{2N}\right)^{1/16}$, we have

$$\left|\mathbb{E}\left[\sum_{m=1}^{M}(a_{m,i} - a_{m,K})^2\widehat{h}_m''(\boldsymbol{A}\boldsymbol{s})\right]\right| \leq \frac{2\sqrt{3C_\phi}\left(\frac{C^2\log\left(\frac{2d_{\mathcal{F}}}{\delta}\right)}{2N}\right)^{1/8}}{3}. \tag{27}$$

D Estimating $\frac{\partial^2\phi(\boldsymbol{s})}{\partial s_i s_j}$

In this subsection, we show a similar bound as in (27) for the cross-derivatives $\frac{\partial^2\phi(\boldsymbol{s})}{\partial s_i s_j}$. The detailed proof is relegated to Appendix 15. The bound for cross-derivatives is as follows.

Given a fixed N, pick $\varepsilon = \sqrt{\frac{C^2\log(2d_{\mathcal{F}}/\delta)}{2N}}$. Then, if $\gamma \geq (\frac{3}{C_\phi})^{1/4}\left(C^2\log\left(\frac{2d_{\mathcal{F}}}{\delta}\right)/2N\right)^{1/16}$, the following holds for $i \neq j$:

$$\left|\mathbb{E}\left[\sum_{m=1}^{M}(a_{m,i} - a_{m,K})(a_{m,j} - a_{m,K})\widehat{h}_m''(\boldsymbol{A}\boldsymbol{s}_\ell)\right]\right| \leq \frac{\sqrt{3C_\phi}\left(\frac{C^2\log\left(\frac{2d_{\mathcal{F}}}{\delta}\right)}{2N}\right)^{1/8}}{6}. \tag{28}$$

E Putting Together

For both derivative estimations, by (27) and (28), we have the following bound since $\|\cdot\|_2$ is upper bounded by $\|\cdot\|_1$:

$$\mathbb{E}\left[\left\|\boldsymbol{G}\widehat{\boldsymbol{h}}''(\boldsymbol{A}\boldsymbol{s})\right\|_2^2\right] = O\left(C_\phi\sqrt{C}\left(\frac{\log\left(\frac{2d_{\mathcal{F}}}{\delta}\right)}{N}\right)^{1/4}\right),$$

which can be further simplified as:

$$\mathbb{E}\left[\left\|\widehat{\boldsymbol{h}}''(\boldsymbol{A}\boldsymbol{s})\right\|_2^2\right] = O\left(\frac{C_\phi\sqrt{C}}{\sigma_{\min}^2(\boldsymbol{G})}\left(\frac{\log\left(\frac{2d_{\mathcal{F}}}{\delta}\right)}{N}\right)^{1/4}\right),$$

where the above is because the expectation is taken over $\boldsymbol{x} \sim \mathcal{D}$ and $\boldsymbol{G}$ is not a function of $\boldsymbol{x}$.

9 Proof of Theorem 5

The key of the proof is to establish a similar bound as in (18) for the NN-approximated case. We start with the following lemma:

Lemma 5 *Consider the function class*

$$\mathcal{H} = \left\{ l(\boldsymbol{x}) \middle| l(\boldsymbol{x}) = \left(1 - \sum_{m=1}^{M} f_m(x_m)\right)^2 \right\}, \tag{29}$$

where each $f_m(\cdot) : \mathbb{R} \to \mathbb{R} \in \mathcal{F}$ which is defined in Assumption 4. Assume that $\boldsymbol{x}_\ell$ for $\ell \in [N]$ are i.i.d. samples from $\mathcal{X}$ according to a certain distribution $\mathcal{D}$. Then, the Rademacher complexity of class $\mathcal{H}$ is bounded by $\mathfrak{R}_N(\mathcal{H}) \leq 2M(MzB^2C_x + 1)\mathfrak{R}_N(\mathcal{F})$, where the bound of class $\mathcal{F}$ is $\mathfrak{R}_N(\mathcal{F}) \leq 2zB^2C_x\sqrt{\frac{R}{N}}$, and C_x is the upper bound of $|x_m|$ for all $\boldsymbol{x} \in \mathcal{X}$.

The detailed proof of this lemma is in Appendix 14. The notion of *Rademacher complexity* (see Appendix 17) is commonly used in statistical machine learning to come up with generalization error bounds in terms of sample complexity, typically in supervised learning (see details in [40]). It also proves handy in proving sample complexity of our unsupervised problem.

Following Lemma 5 and [51, Theorem 3.3] we have that with probability of at least $1 - \delta$, the following holds:

$$\begin{aligned}
\mathbb{E}[(1 - \mathbf{1}^\top \widehat{\boldsymbol{f}}(\boldsymbol{x}))^2] &\leq \frac{1}{N}\sum_{\ell=1}^{N}\left(1 - \mathbf{1}^\top \widehat{\boldsymbol{f}}(\boldsymbol{x}_\ell)\right)^2 + 2\mathfrak{R}_N(\mathcal{H}) + (MzB^2C_x + 1)^2\sqrt{\frac{\log(1/\delta)}{2N}} \\
&\leq \frac{1}{N}\sum_{\ell=1}^{N}\left(1 - \mathbf{1}^\top \widehat{\boldsymbol{f}}(\boldsymbol{x}_\ell)\right)^2 + 8MzB^2C_x(MzB^2C_x + 1)\sqrt{\frac{R}{N}} + (MzB^2C_x + 1)^2\sqrt{\frac{\log(1/\delta)}{2N}}.
\end{aligned}$$

Since we have assumed $\mathcal{G}^{-1} \subseteq \mathcal{F}$, the empirical error should be zero by learning $\widehat{\boldsymbol{f}}$ as an inverse of $\boldsymbol{g}$, i.e., $|1 - \mathbf{1}^\top \widehat{\boldsymbol{f}}(\boldsymbol{x}_\ell)| = 0$, which leads to the following:

$$\mathbb{E}[(1 - \mathbf{1}^\top \widehat{\boldsymbol{f}}(\boldsymbol{x}))^2] \leq 8MzB^2C_x(MzB^2C_x + 1)\sqrt{\frac{R}{N}} + (MzB^2C_x + 1)^2\sqrt{\frac{\log(1/\delta)}{2N}}.$$

By following the similar proof of Theorem 4 for the second-order derivative estimation and let

$$\varepsilon = 8MzB^2C_x(MzB^2C_x + 1)\sqrt{\frac{R}{N}} + (MzB^2C_x + 1)^2\sqrt{\frac{\log(1/\delta)}{2N}}, \tag{30}$$

Thus, when $\gamma = \Omega\left(\left(\frac{MzB^2C_x}{C_\phi}\right)^{1/4}\left(\sqrt{\frac{R}{N}} + \sqrt{\frac{\log(1/\delta)}{N}}\right)^{1/8}\right)$, we have

$$\mathbb{E}\left[\left\|\widehat{\boldsymbol{h}}''(\boldsymbol{A}\boldsymbol{s})\right\|_2^2\right] = O\left(\frac{C_\phi MzB^2C_x\left(\sqrt{R} + \sqrt{\log(1/\delta)}\right)^{1/2}}{\sigma_{\min}^2(\boldsymbol{G})N^{1/4}}\right),$$

for any $\boldsymbol{s}$ such that $1 - \gamma \geq s_i \geq \gamma > 0$ for all $i \in [K]$. This completes the proof for the neural network case.

References

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

[2] X. Fu, W.-K. Ma, K. Huang, and N. D. Sidiropoulos, "Blind separation of quasi-stationary sources: Exploiting convex geometry in covariance domain," *IEEE Trans. Signal Process.*, vol. 63, no. 9, pp. 2306–2320, May 2015.

[3] Y.-O. Li, T. Adali, W. Wang, and V. D. Calhoun, "Joint blind source separation by multiset Canonical Correlation Analysis," *IEEE Trans. Signal Process.*, vol. 57, no. 10, pp. 3918–3929, 2009.

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

[5] W.-K. Ma, J. Bioucas-Dias, T.-H. Chan, N. Gillis, P. Gader, A. Plaza, A. Ambikapathi, and C.-Y. Chi, "A signal processing perspective on hyperspectral unmixing," *IEEE Signal Process. Mag.*, vol. 31, no. 1, pp. 67–81, Jan 2014.

[6] 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.

[7] 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.

[8] 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.

[9] 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.

[10] 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.

[11] K. Huang and X. Fu, "Detecting overlapping and correlated communities without pure nodes: Identifiability and Algorithm," in *Proceedings of ICML 2019*, vol. 97, 09–15 Jun 2019, pp. 2859–2868.

[12] G. Zhou, S. Xie, Z. Yang, J.-M. Yang, and Z. He, "Minimum-volume-constrained nonnegative matrix factorization: Enhanced ability of learning parts," *IEEE Trans. Neural Netw.*, vol. 22, no. 10, pp. 1626–1637, 2011.

[13] S. Ibrahim, X. Fu, N. Kargas, and K. Huang, "Crowdsourcing via pairwise co-occurrences: Identifiability and Algorithms," in *Proceedings of NeurIPS 2019*, 2019, pp. 7847–7857.

[14] N. Dobigeon, J.-Y. Tourneret, C. Richard, J. C. M. Bermudez, S. McLaughlin, and A. O. Hero, "Nonlinear unmixing of hyperspectral images: Models and Algorithms," *IEEE Signal Process. Mag.*, vol. 31, no. 1, pp. 82–94, 2014.

[15] 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.

[16] 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.

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

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

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

[20] A. Hyvarinen and H. Morioka, "Unsupervised feature extraction by time-contrastive learning and nonlinear ICA," in *Proceedings of NeurIPS 2016*, 2016, pp. 3765–3773.

[21] A. Hyvarinen, H. Sasaki, and R. Turner, "Nonlinear ICA using auxiliary variables and generalized contrastive learning," in *Proceedings of AISTATS 2019*, 2019, pp. 859–868.

[22] Y. Deville, "From separability/identifiability properties of bilinear and linear-quadratic mixture matrix factorization to factorization algorithms," *Digital Signal Processing*, vol. 87, pp. 21–33, 2019.

[23] D. G. Fantinato, L. T. Duarte, Y. Deville, R. Attux, C. Jutten, and A. Neves, "A second-order statistics method for blind source separation in post-nonlinear mixtures," *Signal Processing*, vol. 155, pp. 63–72, 2019.

[24] Y. Altmann, N. Dobigeon, and J.-Y. Tourneret, "Unsupervised post-nonlinear unmixing of hyperspectral images using a hamiltonian monte carlo algorithm," *IEEE Trans. Image Process.*, vol. 23, no. 6, pp. 2663–2675, 2014.

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

[26] 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.

[27] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

[28] Q. Lyu and X. Fu, "Nonlinear dependent component analysis: Identifiability and Algorithm," in *EUSIPCO 2020*, Aug 2020.

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

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

[31] 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.

[32] L. E. Larson, "Radio frequency integrated circuit technology for low-power wireless communications," *IEEE Personal Communications*, vol. 5, no. 3, pp. 11–19, 1998.

[33] S. Bermejo, C. Jutten, and J. Cabestany, "ISFET source separation: Foundations and techniques," *Sensors and Actuators B: Chemical*, vol. 113, no. 1, pp. 222–233, 2006.

[34] A. Paraschiv-Ionescu, C. Jutten, and G. Bouvier, "Source separation based processing for integrated Hall sensor arrays," *IEEE Sensors Journal*, vol. 2, no. 6, pp. 663–673, 2002.

[35] 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.

[36] J. M. P. Nascimento and J. M. B. Dias, "Does Independent Component Analysis play a role in unmixing hyperspectral data?" *IEEE Trans. Geosci. Remote Sens.*, vol. 43, no. 1, pp. 175–187, Jan 2005.

[37] T.-H. Chan, C.-Y. Chi, Y.-M. Huang, and W.-K. Ma, "A convex analysis-based minimum-volume enclosing simplex algorithm for hyperspectral unmixing," *IEEE Trans. Signal Process.*, vol. 57, no. 11, pp. 4418 –4432, Nov. 2009.

[38] R. Caron and T. Traynor, "The zero set of a polynomial," *WSMR Report*, pp. 05–02, 2005.

[39] L. Wan, M. Zeiler, S. Zhang, Y. Le Cun, and R. Fergus, "Regularization of neural networks using dropconnect," in *Proceedings of ICML 2013*, 2013, pp. 1058–1066.

[40] P. L. Bartlett and S. Mendelson, "Rademacher and Gaussian complexities: Risk bounds and structural results," *Journal of Machine Learning Research*, vol. 3, no. Nov, pp. 463–482, 2002.

[41] G. E. Hinton and R. S. Zemel, "Autoencoders, minimum description length and Helmholtz free energy," in *Proceedings of NeurIPS 1994*, 1994, pp. 3–10.

[42] D. P. Bertsekas, W. Hager, and O. Mangasarian, *Nonlinear programming*. Athena Scientific Belmont, MA, 1998.

[43] Q. Shi, M. Hong, X. Fu, and T.-H. Chang, "Penalty dual decomposition method for nonsmooth nonconvex optimization–Part II: Applications," *IEEE Trans. Signal Process.*, vol. 68, pp. 4242–4257, 2020.

[44] M. Wang, M. Zhao, J. Chen, and S. Rahardja, "Nonlinear unmixing of hyperspectral data via deep autoencoder networks," *IEEE Geosci. Remote Sens. Lett.*, vol. 16, no. 9, pp. 1467–1471, 2019.

[45] L. Li, S. Ustin, and M. Lay, "Application of multiple endmember spectral mixture analysis (mesma) to aviris imagery for coastal salt marsh mapping: a case study in china camp, ca, usa," *International Journal of Remote Sensing*, vol. 26, no. 23, pp. 5193–5207, 2005.

[46] L. Ji, P. Gong, X. Geng, and Y. Zhao, "Improving the accuracy of the water surface cover type in the 30 m from-glc product," *Remote Sensing*, vol. 7, no. 10, pp. 13 507–13 527, 2015.

[47] N. Gillis and R. Luce, "Robust near-separable nonnegative matrix factorization using linear optimization," *The Journal of Machine Learning Research*, vol. 15, no. 1, pp. 1249–1280, 2014.

[48] A. N. Sameer, K. N. Shree, and H. Murase, "Columbia object image library (COIL-20)," *Tech. Rep., technical report CUCS-005-96*, 1996.

[49] K. Y. Yeung and W. L. Ruzzo, "Details of the Adjusted Rand Index and clustering algorithms, supplement to the paper an empirical study on Principal Component Analysis for clustering gene expression data," *Bioinformatics*, vol. 17, no. 9, pp. 763–774, 2001.

[50] K. Mørken, "Numerical algorithms and digital representation, 2013," *Department of Mathematics, Centre of Mathematics for Applications, University of Oslo*, 2018.

[51] M. Mohri, A. Rostamizadeh, and A. Talwalkar, *Foundations of Machine Learning*. MIT press, 2018.

Supplemental Material of "Identifiability-Guaranteed Simplex-Structured Post-Nonlinear Mixture Learning via Autoencoder" by Qi Lyu and Xiao Fu

10 Proof of Theorem 3

In (7), $\boldsymbol{G}$ has a size of $\frac{K(K-1)}{2} \times M$. Note that when $M > \frac{K(K-1)}{2}$, the null space of $\boldsymbol{G}$ is nontrivial. Hence, we have at least one non-zero solution $\boldsymbol{h}'' = [\iota_1, \cdots, \iota_M]^\top$ such that $\boldsymbol{G}\boldsymbol{h}'' = \boldsymbol{0}$ holds. Our idea is to construct an invertible $h_m(\cdot) = \widehat{f}_m \circ g_m(\cdot)$'s where $\widehat{f}_m$'s are from a solution of (5), while some h_m's are not affine. Note that since $\widehat{f}_m$ and g_m are invertible, the constructed h_m has to be invertible.

Assume that $\iota_m \neq 0$, we construct the corresponding h_m's as follows:

$$h_m(y_m) = \iota_m(y_m + c_m)^2 + \beta_m = \iota_m y_m^2 + 2\iota_m c_m y_m + \iota_m c_m^2 + \beta_m,$$

where c_m and β_m are certain constants, $y_m = \sum_{k=1}^{K-1} b_{m,k}s_k + a_{m,K}$ (by denoting $s_K = 1 - s_1 - \cdots - s_{K-1}$) with $b_{m,k}$ defined in Theorem 2. When $\iota_m \neq 0$, we require each $c_m \leq \min_k a_{m,k}$ or $c_m \geq \max_k a_{m,k}$ for invertibility of h_m. For those $h_m(\cdot)$'s with $\iota_m = 0$, we construct $h_m(y_m) = \omega_m y_m + \beta_m$, where $\omega_m \neq 0$. To have $\boldsymbol{h}'' \in \mathsf{null}(\boldsymbol{G})$, one only needs $\boldsymbol{\iota} = [\iota_1, \ldots, \iota_M]^\top \in \mathsf{null}(\boldsymbol{G})$.

We now show that the above construction can always make $\mathbf{1}^\top \widehat{\boldsymbol{f}}(\boldsymbol{x}) = 1$ satisfied. Consider $\sum_{m=1}^M h_m(\cdot)$ over its domain. First, the quadratic terms can be expressed as follows:

$$\begin{aligned}
&\sum_{m=1}^M \iota_m y_m^2 = \sum_{m=1}^M \iota_m \left(\sum_{k=1}^{K-1} b_{m,k}s_k + a_{m,K}\right)^2 \\
&= \sum_{m=1}^M \iota_m \left(\left(\sum_{k=1}^{K-1} b_{m,k}s_k\right)^2 + 2a_{m,K}\left(\sum_{k=1}^{K-1} b_{m,k}s_k\right) + a_{m,K}^2\right) \\
&= \sum_{m=1}^M \iota_m a_{m,K}\left(2y_m - a_{m,K}\right),
\end{aligned}$$

where the second-order term is zero because it can be written as $\mathbf{1}^\top \boldsymbol{\Sigma}\boldsymbol{G}\boldsymbol{h}''$ and $\boldsymbol{G}\boldsymbol{h}'' = \boldsymbol{0}$, in which $\boldsymbol{\Sigma} = \mathrm{Diag}(s_1^2, \ldots, s_{K-1}^2, s_1s_2, \ldots, s_{K-2}s_{K-1})$ has full rank for $\boldsymbol{s} \in \mathrm{int}\boldsymbol{\Delta}_K$.

Let us define $\mathcal{M}_1 = \{i_1, \ldots, i_{M_1}\}$ and $\mathcal{M}_2 = \{j_1, \ldots, j_{M_2}\}$ as the index sets of the nonzero and zero ι_m's, respectively (not that $\mathcal{M}_1 \cup \mathcal{M}_2 = [M]$ and $\mathcal{M}_1 \cap \mathcal{M}_2 = \emptyset$). Our construction has to respect the constraints of (5), i.e.,

$$1 = \mathbf{1}^\top \boldsymbol{h}(\boldsymbol{A}\boldsymbol{s}) = \sum_{m\in\mathcal{M}_1} [(2\iota_m a_{m,K} + 2\iota_m c_m)y_m + \beta_m - \iota_m a_{m,K}^2 + c_m^2\iota_m] + \sum_{m\in\mathcal{M}_2} (\omega_m y_m + \beta_m). \tag{31}$$

By selecting $\beta_m = \iota_m a_{m,K}^2 - c_m^2\iota_m$ for all m, the constant term of the above becomes zero. Then, we have

$$\sum_{m\in\mathcal{M}_1} (2\iota_m a_{m,K} + 2\iota_m c_m)y_m + \sum_{m\in\mathcal{M}_2} \omega_m y_m = 1, \tag{32}$$

which can be expressed as:

$$\mathbf{1}^\top \boldsymbol{\Sigma}' \boldsymbol{A}\boldsymbol{s} = 1, \tag{33}$$

where

$$\boldsymbol{\Sigma}' = \mathrm{Diag}\left(2\iota_{i_1}a_{i_1,K} + 2\iota_{i_1}c_{i_1}, \ldots, 2\iota_{i_{M_1}}a_{i_{M_1},K} + 2\iota_{i_{M_1}}c_{i_{M_1}}, \omega_{j_1}, \ldots, \omega_{j_{M_2}}\right),$$

in which, without loss of generality (w.o.l.g.), we have assumed that $i_1 \leq \ldots i_{M_1} \leq j_1 \leq \ldots \leq j_{M_2}$. Since $\mathbf{1}^\top \boldsymbol{s} = 1$, our construction only needs to satisfy:

$$\mathbf{1}^\top \boldsymbol{\Sigma}' \boldsymbol{A} = \mathbf{1}^\top \iff \boldsymbol{A}^\top \boldsymbol{z} = \mathbf{1}, \tag{34}$$

where $\boldsymbol{z} = [2\iota_{i_1}(a_{i_1,K} + c_{i_1}), \ldots, 2\iota_{i_{M_1}}(a_{i_{M_1},K} + c_{i_{M_1}}), \omega_{j_1}, \ldots, \omega_{j_{M_2}}]^\top$. By Lemma 1, a dense solution of (34) exists. Denote the dense solution as $\boldsymbol{\tau} = [\tau_1, \ldots, \tau_M]^\top$. Then, we use the following construction to make $\boldsymbol{z} = \boldsymbol{\tau}$:

$$\begin{aligned} 2\iota_m(a_{m,K} + c_m) &= \tau_m, \text{ for } m \text{ with } \iota_m \neq 0, \\ \omega_m &= \tau_m, \text{ for } m \text{ with } \iota_m = 0, \end{aligned}$$

which gives us

$$\begin{aligned} c_m &= \frac{\tau_m}{2\iota_m} - a_{m,K}, \text{ for } m \text{ with } \iota_m \neq 0, \\ \omega_m &= \tau_m, \text{ for } m \text{ with } \iota_m = 0. \end{aligned}$$

It is obvious that if ι_m has a small enough magnitude, c_m has a sufficiently large magnitude. Note that each ι_m can be scaled arbitrarily without violating $\boldsymbol{G}\boldsymbol{h}'' = \mathbf{0}$. As a result, it is guaranteed that there are c_m's such that either $c_m \leq \min_k a_{m,k}$ or $c_m \geq \max_k a_{m,k}$ is satisfied. In addition, the affine $h_m(\cdot)$'s with nonzero $\omega_m = \tau_m$ are also invertible. Since $\widehat{f}_m$ can be any continuous invertible function, there always exists $\widehat{f}_m$ such that $h_m = \widehat{f}_m \circ g_m$ (e.g., by letting $\widehat{f}_m = h_m \circ g_m^{-1}$—which is invertible). Hence, by construction, we have shown that there exists an $\widehat{\boldsymbol{f}}$ that is a solution of (5) while the elements of the corresponding $\boldsymbol{h} = \widehat{\boldsymbol{f}} \circ \boldsymbol{g}$ are not all affine.

11 Proof of Theorem 6

11.1 Finite Function Class

By the assumption that for any $\widehat{f}_m \in \mathcal{F}$ we have:

$$\sup_{\boldsymbol{x} \in \mathcal{X}} |\widehat{f}_m(x_m) - \widehat{u}_m(x_m)| < \nu, \ \forall m \in [M], \ \exists \widehat{u}_m \in \mathcal{G}^{-1}.$$

Hence, one can bound $1 - \mathbf{1}^\top \widehat{\boldsymbol{f}}(\boldsymbol{x}_\ell)$ for all ℓ by introducing $\widehat{\boldsymbol{u}}$, i.e., an inverse of $\boldsymbol{g}$ as defined in Definition 1:

$$\begin{aligned} \left|1 - \mathbf{1}^\top \widehat{\boldsymbol{f}}(\boldsymbol{x}_\ell)\right| &= \left|1 - \mathbf{1}^\top \widehat{\boldsymbol{u}}(\boldsymbol{x}_\ell) + \mathbf{1}^\top \widehat{\boldsymbol{u}}(\boldsymbol{x}_\ell) - \mathbf{1}^\top \widehat{\boldsymbol{f}}(\boldsymbol{x}_\ell)\right| \\ &\leq \left|1 - \mathbf{1}^\top \widehat{\boldsymbol{u}}(\boldsymbol{x}_\ell)\right| + \|\mathbf{1}\|_1 \left\|\widehat{\boldsymbol{u}}(\boldsymbol{x}_\ell) - \widehat{\boldsymbol{f}}(\boldsymbol{x}_\ell)\right\|_\infty \\ &= M\nu, \end{aligned}$$

where the second inequality is by the triangle inequality and Hölder's inequality, and the last equality is because $\widehat{\boldsymbol{u}}$ is an inverse of $\boldsymbol{g}$ such that we have $1 - \mathbf{1}^\top \widehat{\boldsymbol{u}}(\boldsymbol{x}_\ell) = 0$ for all $\ell = [N]$.

Then, we have:

$$\left(1 - \mathbf{1}^\top \widehat{\boldsymbol{f}}(\boldsymbol{x})\right)^2 \leq M^2\nu^2,$$

which implies an approximate feasible solution $\widehat{\boldsymbol{f}}$.

Following Lemma 3 we have:

$$\mathsf{Pr}\left[\{\boldsymbol{x}_\ell\}_{\ell=1}^N : \forall \widehat{\boldsymbol{f}} : \left|\mathcal{P}_\mathcal{D}(\widehat{\boldsymbol{f}}) - \mathcal{P}_N(\widehat{\boldsymbol{f}})\right| \leq \varepsilon\right] \geq 1 - 2d_\mathcal{F} \exp\left(\frac{-2N\varepsilon^2}{C^2}\right).$$

By the definition $\mathcal{P}_N(\widehat{\boldsymbol{f}})$, we have

$$\mathcal{P}_N(\widehat{\boldsymbol{f}}) = \frac{1}{N}\sum_{\ell=1}^{N}\left(1 - \mathbf{1}^\top \widehat{\boldsymbol{f}}(\boldsymbol{x}_\ell)\right)^2 \leq M^2\nu^2.$$

Therefore, the above means that we have the following holds with probability of at least $1-\delta$:

$$\mathbb{E}\left[\left(1 - \mathbf{1}^\top \widehat{\boldsymbol{f}}(\boldsymbol{x})\right)^2\right] \leq \varepsilon + M^2\nu^2,$$

when $N \geq \frac{C^2\log(2d_{\mathcal{F}}/\delta)}{2\varepsilon^2}$.

By applying the similar proof of Theorem 4, note that the difference is that here we have an extra error term $M^2\nu^2$. So, for the finite function class case with (27) and (28), when $\gamma = \Omega\left(\left(\frac{C^2\log(2d_{\mathcal{F}}/\delta)}{NC_\phi^4}\right)^{1/16}\right)$, we have

$$\mathbb{E}\left[\left\|\widehat{\boldsymbol{h}}''(\boldsymbol{A}\boldsymbol{s})\right\|_2^2\right] = O\left(\frac{C_\phi}{\sigma_{\min}^2(\boldsymbol{G})}\left(\frac{C^2\log\left(\frac{2d_{\mathcal{F}}}{\delta}\right)}{N} + M^2\nu^2\right)^{1/4}\right),$$

for any $\boldsymbol{s}$ such that $1-\gamma \geq s_i \geq \gamma > 0$ for all $i \in [K]$.

11.2 Neural Network Function Class

For the case where $\mathcal{F}$ is the neural networks class defined in Assumption 4, similarly we have

$$\frac{1}{N}\sum_{\ell=1}^{N}\left(1 - \mathbf{1}^\top \widehat{\boldsymbol{f}}(\boldsymbol{x}_\ell)\right)^2 \leq M^2\nu^2.$$

Thus, by applying the Rademacher complexity-based generalization bound [1, Theorem 3.3], we have

$$\mathbb{E}\left[\left\|\boldsymbol{G}\widehat{\boldsymbol{h}}''(\boldsymbol{A}\boldsymbol{s})\right\|_2^2\right] = O\left(C_\phi\left(8MzB^2C_x(MzB^2C_x+1)\sqrt{\frac{R}{N}}\right.\right.$$
$$\left.\left.+(MzB^2C_x+1)^2\sqrt{\frac{\log(1/\delta)}{2N}} + M^2\nu^2\right)^{1/2}\right),$$

when $\gamma = \Omega\left(\left(\frac{MzB^2C_x}{C_\phi}\right)^{1/4}\left(\sqrt{\frac{R}{N}} + \sqrt{\frac{\log(1/\delta)}{N}}\right)^{1/8}\right)$, which can be further simplified as (by inequality $(x+y)^{1/2} \leq x^{1/2} + y^{1/2}$ for $x, y > 0$)

$$\mathbb{E}\left[\left\|\widehat{\boldsymbol{h}}''(\boldsymbol{A}\boldsymbol{s})\right\|_2^2\right] = O\left(\frac{C_\phi MzB^2C_x\left(\sqrt{R} + \sqrt{\log(1/\delta)}\right)^{1/2}}{\sigma_{\min}^2(\boldsymbol{G})N^{1/4}} + \frac{M\nu}{\sigma_{\min}^2(\boldsymbol{G})}\right),$$

for any $\boldsymbol{s}$ such that $1-\gamma \geq s_i \geq \gamma > 0$ for all $i \in [K]$.

12 Proof of Lemma 3

The proof follows the standard uniform convergence technique in statistical learning. Let us define the following two terms:

$$\mathcal{P}_N(\widehat{\boldsymbol{f}}) = \frac{1}{N}\sum_{\ell=1}^{N}(1 - \mathbf{1}^\top \widehat{\boldsymbol{f}}(\boldsymbol{x}_\ell))^2, \tag{35a}$$

$$\mathcal{P}_{\mathcal{D}}(\widehat{\boldsymbol{f}}) = \mathbb{E}[(1 - \mathbf{1}^\top \widehat{\boldsymbol{f}}(\boldsymbol{x}_\ell))^2]. \tag{35b}$$

By the Hoeffding's inequality (see Appendix 17), for any $\widehat{\boldsymbol{f}}$, we have

$$\mathsf{Pr}\left[\left|\mathcal{P}_{\mathcal{D}}(\widehat{\boldsymbol{f}}) - \mathcal{P}_N(\widehat{\boldsymbol{f}})\right| > \varepsilon\right] \leq 2\exp(-2N\varepsilon^2/C^2). \tag{36}$$

By the union bound, one can easily show that

$$\mathsf{Pr}\left[\{\boldsymbol{x}_\ell\}_{\ell=1}^N : \exists \widehat{\boldsymbol{f}} : \left|\mathcal{P}_{\mathcal{D}}(\widehat{\boldsymbol{f}}) - \mathcal{P}_N(\widehat{\boldsymbol{f}})\right| > \varepsilon\right] \leq 2d_{\mathcal{F}} \exp\left(\frac{-2N\varepsilon^2}{C^2}\right),$$

Note that under our generative model and $\mathcal{F} \cup \mathcal{G}^{-1} \neq \emptyset$, there always exists a solution in $\mathcal{F}$ for (10). Hence, $\mathcal{P}_N(\widehat{\boldsymbol{f}}) = 0$.

13 Proof of Lemma 4

It is readily seen that

$$\frac{\partial^4 \phi(\boldsymbol{s})}{\partial s_i^4} = \sum_{m=1}^{M}(a_{m,i} - a_{m,K})^4 h_m^{(4)}\left(\boldsymbol{A}\boldsymbol{s}\right), \ \forall i \in [K-1].$$

Note that by rudimentary algebra, we have the following inequality:

$$\begin{aligned} h^{(4)} &= (f \circ g)^{(4)} \\ &= f^{(4)}(g)\cdot(g')^4 + 6f^{(3)}(g)\cdot(g')^2\cdot g'' + 3f''(g)\cdot(g'')^2 + 4f''(g)\cdot g'\cdot g^{(3)} + f'(g)\cdot g^{(4)} \\ &\leq C_f C_g^4 + 6C_f C_g^3 + 3C_f C_g^2 + 4C_f C_g^2 + C_f C_g, \end{aligned}$$

where we have dropped the subscripts of f, g and h for notational simplicity. By the definition of C_ϕ, we have

$$\left|\frac{\partial^4 \phi(\boldsymbol{s})}{\partial s_i^4}\right| \leq \left|\sum_{m=1}^{M}(a_{m,i} - a_{m,K})^4 h_m^{(4)}\left(\boldsymbol{A}\boldsymbol{s}\right)\right| \leq \left|\sum_{m=1}^{M}(2C_a)^4 h_m^{(4)}\left(\boldsymbol{A}\boldsymbol{s}\right)\right| \leq C_\phi. \tag{37}$$

For the cross derivatives, the following holds:

$$\frac{\partial^4 \phi(\boldsymbol{s})}{\partial s_i^3 \partial s_j} = \sum_{m=1}^{M}(a_{m,i} - a_{m,K})^3 (a_{m,j} - a_{m,K}) h_m^{(4)}\left(\boldsymbol{A}\boldsymbol{s}\right),$$

for all $i, j \in [K-1]$. Using similar derivation as in (37), one can attain the same bound for $\left|\frac{\partial^4 \phi(\boldsymbol{s})}{\partial s_i^3 \partial s_j}\right|$ and $\left|\frac{\partial^4 \phi(\boldsymbol{s})}{\partial s_i^2 \partial s_j^2}\right|$.

14 Proof of Lemma 5

In this section, we show the Rademacher complexity of the function class that we use. We first derive the Rademacher complexity of function class $\mathcal{F}$ which is defined in Assumption 4. By [2, Theorem 43], the Rademacher complexity of a bounded two-layer neural network $\mathcal{F}$ with z-Lipschitz activation is

$$\mathfrak{R}_N(\mathcal{F}) \leq 2zB^2C_x\sqrt{\frac{R}{N}}.$$

To further consider the Rademacher complexity of the loss function class defined in (29), first note that the Rademacher complexity has the following property [3]:

$$\mathfrak{R}_N\left(\mathcal{F}_1+\mathcal{F}_2\right)=\mathfrak{R}_N\left(\mathcal{F}_1\right)+\mathfrak{R}_N\left(\mathcal{F}_2\right),$$

where $f \in \mathcal{F}_1+\mathcal{F}_2$ means f is a linear combination of functions from $\mathcal{F}_1$ and $\mathcal{F}_2$. Hence, the complexity of function $\sum_{m=1}^{M} f_m(\cdot)$ is bounded by $M\mathfrak{R}_N(\mathcal{F})$.

Each function $f_m(\cdot)$ can be expressed as $f_m(x) = \boldsymbol{w}_2^\top \boldsymbol{\zeta}(\boldsymbol{w}_1 x)$. With the assumption that $\zeta(0) = 0$, we have $f_m(0) = 0$ and

$$\begin{aligned}
|f_m(x) - f_m(0)| &= |\boldsymbol{w}_2^\top \boldsymbol{\zeta}(\boldsymbol{w}_1 x) - \boldsymbol{w}_2^\top \boldsymbol{\zeta}(\mathbf{0})| \\
&\leq \|\boldsymbol{w}_2\|_2 \|\boldsymbol{\zeta}(\boldsymbol{w}_1 x) - \boldsymbol{\zeta}(\mathbf{0})\|_2 \\
&\leq zB\|\boldsymbol{w}_1 x - \mathbf{0}\|_2 \\
&\leq zB^2C_x, \\
\Longrightarrow |f_m(x)| &\leq zB^2C_x,
\end{aligned}$$

where the second is by the Cauchy–Schwarz inequality, the third one is by the Lipschitz continuity of $\zeta(\cdot)$, and the last one is also by the Cauchy–Schwarz inequality. Therefore, we claim that function $f_m(x_m)$ is bounded within $[-zB^2C_x,\ zB^2C_x]$.

Accordingly, $|1 - \sum_{m=1}^{M} f_m(x_m)|$ can be bounded within $[0, MzB^2C_x + 1]$. By the Lipschitz composition property of Rademacher complexity that $\mathfrak{R}_N(\phi \circ \mathcal{F}) \leq L_\phi \mathfrak{R}_N(\mathcal{F})$ where L_ϕ denotes the Lipschitz constant of ϕ. Here, the Lipschitz constant for the loss function is $L_\phi = 2(MzB^2C_x + 1)$. Thus we have

$$\mathfrak{R}_N(\mathcal{H}) \leq 2(MzB^2C_x + 1)M\mathfrak{R}_N(\mathcal{F}).$$

By plugging in $\mathfrak{R}_N(\mathcal{F})$ we have

$$\begin{aligned}
\mathfrak{R}_N(\mathcal{H}) &\leq 2(MzB^2C_x + 1)M\mathfrak{R}_N(\mathcal{F}) \\
&\leq 2M(MzB^2C_x + 1)2zB^2C_x\sqrt{\frac{R}{N}} \\
&= 4MzB^2C_x(MzB^2C_x + 1)\sqrt{\frac{R}{N}}.
\end{aligned}$$

15 Estimation Error Bound for Second-Order Cross Derivatives

First, we have the following Lemma for second-order cross derivatives.

Lemma 6 *For the cross derivative of a continuous function $\psi(x,y)$ w.r.t. both of its arguments, we have the following numerical estimation*

$$\begin{aligned}\frac{\partial^2\psi(x,y)}{\partial x\partial y} &= \frac{\psi(x+\Delta x,y+\Delta y)-\psi(x+\Delta x,y-\Delta y)}{4\Delta x\Delta y} - \frac{\psi(x-\Delta x,y+\Delta y)-\psi(x-\Delta x,y-\Delta y)}{4\Delta x\Delta y}\\ &- \frac{\Delta x^2}{6}\frac{\partial^4\psi(\xi'_{11},\xi'_{21})}{\partial x^3\partial y} - \frac{\Delta y^2}{6}\frac{\partial^4\psi(\xi'_{12},\xi'_{22})}{\partial x\partial y^3} - \frac{\Delta x^3}{48\Delta y}\left(\frac{\partial^4\psi(\xi'_{13},\xi'_{23})}{\partial x^4} - \frac{\partial^4\psi(\xi'_{14},\xi'_{24})}{\partial x^4}\right)\\ &- \frac{\Delta x\Delta y}{8}\left(\frac{\partial^4\psi(\xi'_{15},\xi'_{25})}{\partial x^2\partial y^2} - \frac{\partial^4\psi(\xi'_{16},\xi'_{26})}{\partial x^2\partial y^2}\right) - \frac{\Delta y^3}{48\Delta x}\left(\frac{\partial^4\psi(\xi'_{17},\xi'_{27})}{\partial y^4} - \frac{\partial^4\psi(\xi'_{18},\xi'_{28})}{\partial y^4}\right),\end{aligned}$$

where $\xi'_{1i}\in(x-\Delta x,x+\Delta x)$ and $\xi'_{2i}\in(y-\Delta y,y+\Delta y)$ for $i\in\{1,\cdots 8\}$.

Proof: For any two-dimensional functions whose higher-order derivatives exist, we have the following holds:

$$\begin{aligned}\psi(x+\Delta x,y+\Delta y) &= \psi(x,y) + \frac{\partial\psi(x,y)}{\partial x}\Delta x + \frac{\partial\psi(x,y)}{\partial y}\Delta y + \frac{\partial^2\psi(x,y)}{\partial x^2}\frac{\Delta x^2}{2} + \frac{\partial^2\psi(x,y)}{\partial x\partial y}\Delta x\Delta y\\ &+ \frac{\partial^2\psi(x,y)}{\partial y^2}\frac{\Delta y^2}{2} + \frac{\partial^3\psi(x,y)}{\partial x^3}\frac{\Delta x^3}{6} + \frac{\partial^3\psi(x,y)}{\partial x^2\partial y}\frac{\Delta x^2\Delta y}{2} + \frac{\partial^3\psi(x,y)}{\partial x\partial y^2}\frac{\Delta x\Delta y^2}{2}\\ &+ \frac{\partial^3\psi(x,y)}{\partial y^3}\frac{\Delta y^3}{6} + \frac{\partial^4\psi(\xi_{11},\xi_{21})}{\partial x^4}\frac{\Delta x^4}{24} + \frac{\partial^4\psi(\xi_{11},\xi_{21})}{\partial x^3\partial y}\frac{\Delta x^3\Delta y}{6}\\ &+ \frac{\partial^4\psi(\xi_{11},\xi_{21})}{\partial x^2\partial y^2}\frac{\Delta x^2\Delta y^2}{4} + \frac{\partial^4\psi(\xi_{11},\xi_{21})}{\partial x\partial y^3}\frac{\Delta x\Delta y^3}{6} + \frac{\partial^4\psi(\xi_{11},\xi_{21})}{\partial y^4}\frac{\Delta y^4}{24}.\end{aligned}$$

We have similar representations for $\psi(x+\Delta x,y-\Delta y)$, $\psi(x-\Delta x,y+\Delta y)$, and $\psi(x-\Delta x,y-\Delta y)$ following the basic rule of Taylor expansion (which are omitted for space).

By putting together, we have the following holds:

$$\begin{aligned}&\psi(x+\Delta x,y+\Delta y)-\psi(x+\Delta x,y-\Delta y)-\psi(x-\Delta x,y+\Delta y)+\psi(x-\Delta x,y-\Delta y)\\ &= \frac{\partial^2\psi(x,y)}{\partial x\partial y}4\Delta x\Delta y + \sum_{i=1}^{4}\frac{\partial^4\psi(\xi_{1i},\xi_{2i})}{\partial x^3\partial y}\frac{\Delta x^3\Delta y}{6} + \sum_{i=1}^{4}\frac{\partial^4\psi(\xi_{1i},\xi_{2i})}{\partial x\partial y^3}\frac{\Delta x\Delta y^3}{6}\\ &+ \sum_{i=1}^{4}(-1)^{i+1}\frac{\partial^4\psi(\xi_{1i},\xi_{2i})}{\partial x^4}\frac{\Delta x^4}{24} + \sum_{i=1}^{4}(-1)^{i+1}\frac{\partial^4\psi(\xi_{1i},\xi_{2i})}{\partial x^2\partial y^2}\frac{\Delta x^2\Delta y^2}{4} + \sum_{i=1}^{4}(-1)^{i+1}\frac{\partial^4\psi(\xi_{1i},\xi_{2i})}{\partial y^4}\frac{\Delta y^4}{24},\end{aligned}\tag{38}$$

where $\xi_{11}\in(x,x+\Delta x)$, $\xi_{21}\in(y,y+\Delta y)$, $\xi_{12}\in(x,x+\Delta x)$, $\xi_{22}\in(y-\Delta y,y)$, $\xi_{13}\in(x-\Delta x,x)$, $\xi_{23}\in(y-\Delta y,y)$, $\xi_{14}\in(x-\Delta x,x)$ and $\xi_{24}\in(y,y+\Delta y)$.

By the intermediate value theorem, there exists points $\xi'_{1i}\in(x-\Delta x,x+\Delta x)$ and $\xi'_{2i}\in(y-\Delta y,y+\Delta y)$ for $i\in\{1,\cdots,8\}$ such that

$$\sum_{i=1}^{4}\frac{\partial^4\psi(\xi_{1i},\xi_{2i})}{\partial x^3\partial y} = \frac{4\partial^4\psi(\xi'_{11},\xi'_{21})}{\partial x^3\partial y},\quad \sum_{i=1}^{4}\frac{\partial^4\psi(\xi_{1i},\xi_{2i})}{\partial x\partial y^3} = \frac{4\partial^4\psi(\xi'_{12},\xi'_{22})}{\partial x\partial y^3},$$

and

$$\begin{aligned}\sum_{i=1}^{4}(-1)^{i+1}\frac{\partial^4\psi(\xi_{1i},\xi_{2i})}{\partial x^4} &= 2\left(\frac{\partial^4\psi(\xi'_{13},\xi'_{23})}{\partial x^4} - \frac{\partial^4\psi(\xi'_{14},\xi'_{24})}{\partial x^4}\right),\\ \sum_{i=1}^{4}(-1)^{i+1}\frac{\partial^4\psi(\xi_{1i},\xi_{2i})}{\partial x^2\partial y^2} &= 2\left(\frac{\partial^4\psi(\xi'_{15},\xi'_{25})}{\partial x^2\partial y^2} - \frac{\partial^4\psi(\xi'_{16},\xi'_{26})}{\partial x^2\partial y^2}\right),\\ \sum_{i=1}^{4}(-1)^{i+1}\frac{\partial^4\psi(\xi_{1i},\xi_{2i})}{\partial y^4} &= 2\left(\frac{\partial^4\psi(\xi'_{17},\xi'_{27})}{\partial y^4} - \frac{\partial^4\psi(\xi'_{18},\xi'_{28})}{\partial y^4}\right).\end{aligned}$$

By combining the above and dividing Eq. (38) by $4\Delta x\Delta y$ we have (for the right hand side)

$$\frac{\partial^2\psi(x,y)}{\partial x\partial y}+\frac{\partial^4\psi(\xi'_{11},\xi'_{21})}{\partial x^3\partial y}\frac{\Delta x^2}{6}+\frac{\partial^4\psi(\xi'_{12},\xi'_{22})}{\partial x\partial y^3}\frac{\Delta y^2}{6}+\left(\frac{\partial^4\psi(\xi'_{13},\xi'_{23})}{\partial x^4}-\frac{\partial^4\psi(\xi'_{14},\xi'_{24})}{\partial x^4}\right)\frac{\Delta x^3}{48\Delta y}$$
$$+\left(\frac{\partial^4\psi(\xi'_{15},\xi'_{25})}{\partial x^2\partial y^2}-\frac{\partial^4\psi(\xi'_{16},\xi'_{26})}{\partial x^2\partial y^2}\right)\frac{\Delta x\Delta y}{8}+\left(\frac{\partial^4\psi(\xi'_{17},\xi'_{27})}{\partial y^4}-\frac{\partial^4\psi(\xi'_{18},\xi'_{28})}{\partial y^4}\right)\frac{\Delta y^3}{48\Delta x},$$

where the first term is what we aim to estimate, with the rest as error terms. □

To further show the bound, we define:

$$\begin{aligned}
\Delta\boldsymbol{s}_{ij}^{++}&=[\mathbf{0},\dots,+\Delta s_i,\dots,\mathbf{0},\dots,+\Delta s_j,\dots,\mathbf{0},-\Delta s_i-\Delta s_j]^\top,\\
\Delta\boldsymbol{s}_{ij}^{+-}&=[\mathbf{0},\dots,+\Delta s_i,\dots,\mathbf{0},\dots,-\Delta s_j,\dots,\mathbf{0},-\Delta s_i+\Delta s_j]^\top,\\
\Delta\boldsymbol{s}_{ij}^{-+}&=[\mathbf{0},\dots,-\Delta s_i,\dots,\mathbf{0},\dots,+\Delta s_j,\dots,\mathbf{0},+\Delta s_i-\Delta s_j]^\top,\\
\Delta\boldsymbol{s}_{ij}^{--}&=[\mathbf{0},\dots,-\Delta s_i,\dots,\mathbf{0},\dots,-\Delta s_j,\dots,\mathbf{0},+\Delta s_i+\Delta s_j]^\top,
\end{aligned}$$

with $\Delta s_i>0$ and $\Delta s_j>0$ for any $i,j\in[K-1]$ with $i<j$ and

$$\Delta s_i\in\mathcal{S}_i=[0,\min\{s_{i,\ell},1-s_{i,\ell}\}),\quad \Delta s_j\in\mathcal{S}_j=[0,\min\{s_{j,\ell},1-s_{j,\ell}\}).$$

Define $\boldsymbol{s}_{\widehat{\ell}}=\boldsymbol{s}_\ell+\Delta\boldsymbol{s}_{ij}^{++}$, $\boldsymbol{s}_{\widetilde{\ell}}=\boldsymbol{s}_\ell+\Delta\boldsymbol{s}_{ij}^{+-}$, $\boldsymbol{s}_{\overline{\ell}}=\boldsymbol{s}_\ell+\Delta\boldsymbol{s}_{ij}^{-+}$, and $\boldsymbol{s}_{\ell'}=\boldsymbol{s}_\ell+\Delta\boldsymbol{s}_{ij}^{--}$. Then, we have

$$\begin{aligned}
\mathbf{1}^\top\widehat{\boldsymbol{h}}(\boldsymbol{A}(\boldsymbol{s}_\ell+\Delta\boldsymbol{s}_{ij}^{++}))&=1\pm\sqrt{\varepsilon_{\widehat{\ell}}},\quad \mathbf{1}^\top\widehat{\boldsymbol{h}}(\boldsymbol{A}(\boldsymbol{s}_\ell+\Delta\boldsymbol{s}_{ij}^{+-}))=1\pm\sqrt{\varepsilon_{\widetilde{\ell}}},\\
\mathbf{1}^\top\widehat{\boldsymbol{h}}(\boldsymbol{A}(\boldsymbol{s}_\ell+\Delta\boldsymbol{s}_{ij}^{-+}))&=1\pm\sqrt{\varepsilon_{\overline{\ell}}},\quad \mathbf{1}^\top\widehat{\boldsymbol{h}}(\boldsymbol{A}(\boldsymbol{s}_\ell+\Delta\boldsymbol{s}_{ij}^{--}))=1\pm\sqrt{\varepsilon_{\ell'}}.
\end{aligned}\tag{39}$$

For any continuous function $\psi(x,y)$ which has non-vanishing fourth-order partial derivatives, the second-order cross derivatives can be expressed using the following formula (see Lemma 6):

$$\begin{aligned}
\frac{\partial^2\psi(x,y)}{\partial x\partial y}=&\frac{\psi(x+\Delta x,y+\Delta y)-\psi(x+\Delta x,y-\Delta y)}{4\Delta x\Delta y}-\frac{\psi(x-\Delta x,y+\Delta y)-\psi(x-\Delta x,y-\Delta y)}{4\Delta x\Delta y}\\
&-\frac{\Delta x^2}{6}\frac{\partial^4\psi(\xi_{11},\xi_{21})}{\partial x^3\partial y}-\frac{\Delta y^2}{6}\frac{\partial^4\psi(\xi_{12},\xi_{22})}{\partial x\partial y^3}-\frac{\Delta x^3}{48\Delta y}\left(\frac{\partial^4\psi(\xi_{13},\xi_{23})}{\partial x^4}-\frac{\partial^4\psi(\xi_{14},\xi_{24})}{\partial x^4}\right)\\
&-\frac{\Delta x\Delta y}{8}\left(\frac{\partial^4\psi(\xi_{15},\xi_{25})}{\partial x^2\partial y^2}-\frac{\partial^4\psi(\xi_{16},\xi_{26})}{\partial x^2\partial y^2}\right)-\frac{\Delta y^3}{48\Delta x}\left(\frac{\partial^4\psi(\xi_{17},\xi_{27})}{\partial y^4}-\frac{\partial^4\psi(\xi_{18},\xi_{28})}{\partial y^4}\right),
\end{aligned}$$

where $\xi_{1i}\in(x-\Delta x,x+\Delta x)$ and $\xi_{2i}\in(y-\Delta y,y+\Delta y)$ for $i\in\{1,\cdots,8\}$.

Using the above formula, one can express $\frac{\partial^2\phi(\boldsymbol{s})}{\partial s_i\partial s_j}$ as follows:

$$\begin{aligned}
\frac{\partial^2\phi(\boldsymbol{s})}{\partial s_i\partial s_j}=&\frac{\pm\sqrt{\varepsilon_{\widehat{\ell}}}\mp\sqrt{\varepsilon_{\widetilde{\ell}}}\mp\sqrt{\varepsilon_{\overline{\ell}}}\pm\sqrt{\varepsilon_{\ell'}}}{4\Delta s_i\Delta s_j}-\frac{\Delta s_i^2}{6}\frac{\partial^4\phi(\boldsymbol{\xi}_{ij}^{(1)})}{\partial s_i^3\partial s_j}-\frac{\Delta s_j^2}{6}\frac{\partial^4\phi(\boldsymbol{\xi}_{ij}^{(2)})}{\partial s_i\partial s_j^3}-\frac{\Delta s_i^3}{48\Delta s_j}\left(\frac{\partial^4\phi(\boldsymbol{\xi}_{ij}^{(3)})}{\partial s_i^4}-\frac{\partial^4\phi(\boldsymbol{\xi}_{ij}^{(4)})}{\partial s_i^4}\right)\\
&-\frac{\Delta s_i\Delta s_j}{8}\left(\frac{\partial^4\phi(\boldsymbol{\xi}_{ij}^{(5)})}{\partial s_i^2\partial s_j^2}-\frac{\partial^4\phi(\boldsymbol{\xi}_{ij}^{(6)})}{\partial s_i^2\partial s_j^2}\right)-\frac{\Delta s_j^3}{48\Delta s_i}\left(\frac{\partial^4\phi(\boldsymbol{\xi}_{ij}^{(7)})}{\partial s_j^4}-\frac{\partial^4\phi(\boldsymbol{\xi}_{ij}^{(8)})}{\partial s_j^4}\right),
\end{aligned}$$

where $\boldsymbol{\xi}_{ij}^{(k)}$'s are vectors satisfying

$$\boldsymbol{\xi}_{ij}^{(k)}=\theta^{(k)}\boldsymbol{s}_{\widehat{\ell}}+(1-\theta^{(k)})\boldsymbol{s}_{\ell'}\in\mathrm{int}\Delta,k\in\{1,\cdots,8\},$$

where $\theta^{(k)}\in(0,1)$, is a vector such that $[\boldsymbol{\xi}_{ij}^{(k)}]_i\in(s_{i,\ell}-\Delta s_i,s_{i,\ell}+\Delta s_i)$ and $[\boldsymbol{\xi}_{ij}^{(k)}]_j\in(s_{j,\ell}-\Delta s_j,s_{j,\ell}+\Delta s_j)$.

Therefore, the following holds:

$$\left|\frac{\partial^2\phi(\boldsymbol{s})}{\partial s_i\partial s_j}\right| \le \frac{\sqrt{\varepsilon_{\widehat{\ell}}}+\sqrt{\varepsilon_{\widetilde{\ell}}}+\sqrt{\varepsilon_{\overline{\ell}}}+\sqrt{\varepsilon_{\ell'}}}{4\Delta s_i\Delta s_j} + \frac{\Delta s_i^2}{6}\left|\frac{\partial^4\phi(\boldsymbol{\xi}_{ij}^{(1)})}{\partial s_i^3\partial s_j}\right| + \frac{\Delta s_j^2}{6}\left|\frac{\partial^4\phi(\boldsymbol{\xi}_{ij}^{(2)})}{\partial s_i\partial s_j^3}\right| + \frac{\Delta s_i^3}{48\Delta s_j}\left(\left|\frac{\partial^4\phi(\boldsymbol{\xi}_{ij}^{(3)})}{\partial s_i^4}\right| + \left|\frac{\partial^4\phi(\boldsymbol{\xi}_{ij}^{(4)})}{\partial s_i^4}\right|\right) + \frac{\Delta s_i\Delta s_j}{8}\left(\left|\frac{\partial^4\phi(\boldsymbol{\xi}_{ij}^{(5)})}{\partial s_i^2\partial s_j^2}\right| + \left|\frac{\partial^4\phi(\boldsymbol{\xi}_{ij}^{(6)})}{\partial s_i^2\partial s_j^2}\right|\right) + \frac{\Delta s_j^3}{48\Delta s_i}\left(\left|\frac{\partial^4\phi(\boldsymbol{\xi}_{ij}^{(7)})}{\partial s_j^4}\right| + \left|\frac{\partial^4\phi(\boldsymbol{\xi}_{ij}^{(8)})}{\partial s_j^4}\right|\right).$$

By taking expectation and using Jensen's inequality,

$$\left|\mathbb{E}\left[\sum_{m=1}^{M}(a_{m,i}-a_{m,K})(a_{m,j}-a_{m,K})\widehat{h}_m''(\boldsymbol{A}\boldsymbol{s}_\ell)\right]\right| \le \frac{\sqrt{\varepsilon}}{\Delta s_i\Delta s_j} + \frac{\Delta s_i^2}{6}\left|\frac{\partial^4\phi(\boldsymbol{\xi}_{ij}^{(1)})}{\partial s_i^3\partial s_j}\right| + \frac{\Delta s_j^2}{6}\left|\frac{\partial^4\phi(\boldsymbol{\xi}_{ij}^{(2)})}{\partial s_i\partial s_j^3}\right| + \frac{\Delta s_i^3}{48\Delta s_j}\left(\left|\frac{\partial^4\phi(\boldsymbol{\xi}_{ij}^{(3)})}{\partial s_i^4}\right| + \left|\frac{\partial^4\phi(\boldsymbol{\xi}_{ij}^{(4)})}{\partial s_i^4}\right|\right) + \frac{\Delta s_i\Delta s_j}{8}\left(\left|\frac{\partial^4\phi(\boldsymbol{\xi}_{ij}^{(5)})}{\partial s_i^2\partial s_j^2}\right| + \left|\frac{\partial^4\phi(\boldsymbol{\xi}_{ij}^{(6)})}{\partial s_i^2\partial s_j^2}\right|\right) + \frac{\Delta s_j^3}{48\Delta s_i}\left(\left|\frac{\partial^4\phi(\boldsymbol{\xi}_{ij}^{(7)})}{\partial s_j^4}\right| + \left|\frac{\partial^4\phi(\boldsymbol{\xi}_{ij}^{(8)})}{\partial s_j^4}\right|\right),$$

where $\varepsilon = \varepsilon(N,\delta)$ which will be specified later.

We are interested in finding the optimal upper bound

$$\inf_{\Delta s_i,\Delta s_j}\frac{\sqrt{\varepsilon_0}}{\Delta s_i\Delta s_j} + \frac{\Delta s_i^2}{6}\left|\frac{\partial^4\phi(\boldsymbol{\xi}_{ij}^{(1)})}{\partial s_i^3\partial s_j}\right| + \frac{\Delta s_j^2}{6}\left|\frac{\partial^4\phi(\boldsymbol{\xi}_{ij}^{(2)})}{\partial s_i\partial s_j^3}\right| + \frac{\Delta s_i^3}{48\Delta s_j}\left(\left|\frac{\partial^4\phi(\boldsymbol{\xi}_{ij}^{(3)})}{\partial s_i^4}\right| + \left|\frac{\partial^4\phi(\boldsymbol{\xi}_{ij}^{(4)})}{\partial s_i^4}\right|\right) + \frac{\Delta s_i\Delta s_j}{8}\left(\left|\frac{\partial^4\phi(\boldsymbol{\xi}_{ij}^{(5)})}{\partial s_i^2\partial s_j^2}\right| + \left|\frac{\partial^4\phi(\boldsymbol{\xi}_{ij}^{(6)})}{\partial s_i^2\partial s_j^2}\right|\right) + \frac{\Delta s_j^3}{48\Delta s_i}\left(\left|\frac{\partial^4\phi(\boldsymbol{\xi}_{ij}^{(7)})}{\partial s_j^4}\right| + \left|\frac{\partial^4\phi(\boldsymbol{\xi}_{ij}^{(8)})}{\partial s_j^4}\right|\right).$$

Without loss of generality, we assume that $\Delta s = \Delta s_i = \Delta s_j$, with its feasible domain

$$\Delta s \in [0, \min\{s_{i,\ell}, s_{i,\ell}, 1-s_{i,\ell}, 1-s_{j,\ell}\}),$$

and we have the following

$$\inf_{\Delta s}\frac{\sqrt{\varepsilon_0}}{\Delta s^2} + \frac{\Delta s^2}{6}\left(\left|\frac{\partial^4\phi(\boldsymbol{\xi}_{ij}^{(1)})}{\partial s_i^3\partial s_j}\right| + \left|\frac{\partial^4\phi(\boldsymbol{\xi}_{ij}^{(2)})}{\partial s_i\partial s_j^3}\right|\right) + \frac{\Delta s^2}{48}\left(\left|\frac{\partial^4\phi(\boldsymbol{\xi}_{ij}^{(3)})}{\partial s_i^4}\right| + \left|\frac{\partial^4\phi(\boldsymbol{\xi}_{ij}^{(4)})}{\partial s_i^4}\right|\right) + \frac{\Delta s^2}{8}\left(\left|\frac{\partial^4\phi(\boldsymbol{\xi}_{ij}^{(5)})}{\partial s_i^2\partial s_j^2}\right| + \left|\frac{\partial^4\phi(\boldsymbol{\xi}_{ij}^{(6)})}{\partial s_i^2\partial s_j^2}\right|\right) + \frac{\Delta s^2}{48}\left(\left|\frac{\partial^4\phi(\boldsymbol{\xi}_{ij}^{(7)})}{\partial s_j^4}\right| + \left|\frac{\partial^4\phi(\boldsymbol{\xi}_{ij}^{(8)})}{\partial s_j^4}\right|\right).$$

Define τ as

$$\tau := 8\left(\left|\frac{\partial^4\phi(\boldsymbol{\xi}_{ij}^{(1)})}{\partial s_i^3\partial s_j}\right| + \left|\frac{\partial^4\phi(\boldsymbol{\xi}_{ij}^{(2)})}{\partial s_i\partial s_j^3}\right|\right) + \left(\left|\frac{\partial^4\phi(\boldsymbol{\xi}_{ij}^{(3)})}{\partial s_i^4}\right| + \left|\frac{\partial^4\phi(\boldsymbol{\xi}_{ij}^{(4)})}{\partial s_i^4}\right|\right) + 6\left(\left|\frac{\partial^4\phi(\boldsymbol{\xi}_{ij}^{(5)})}{\partial s_i^2\partial s_j^2}\right| + \left|\frac{\partial^4\phi(\boldsymbol{\xi}_{ij}^{(6)})}{\partial s_i^2\partial s_j^2}\right|\right) + \left(\left|\frac{\partial^4\phi(\boldsymbol{\xi}_{ij}^{(7)})}{\partial s_j^4}\right| + \left|\frac{\partial^4\phi(\boldsymbol{\xi}_{ij}^{(8)})}{\partial s_j^4}\right|\right).$$

So the minimum is obtained at

$$\Delta s^* \in \left\{ \left(\frac{48\sqrt{\varepsilon_0}}{\tau} \right)^{1/4}, \min\{s_{i,\ell}, s_{j,\ell}, 1 - s_{i,\ell}, 1 - s_{j,\ell}\} \right\},$$

which gives the following minimum

$$\inf_{\Delta s} \frac{\sqrt{\varepsilon_0}}{\Delta s^2} + \frac{\tau \Delta s^2}{48} \leq \min \left\{ \frac{\sqrt{3}\varepsilon_0^{1/4}\sqrt{\tau}}{6}, \frac{\sqrt{\varepsilon_0}}{\kappa^2} + \frac{\kappa^2 \tau}{6} \right\}, \tag{40}$$

where $\kappa = \min\{s_{i,\ell}, s_{j,\ell}, 1 - s_{i,\ell}, 1 - s_{j,\ell}\}$. Similarly we have

$$\kappa \geq \gamma.$$

Hence, if

$$\left(\frac{48\sqrt{\varepsilon_0}}{\tau} \right)^{1/4} \leq \gamma,$$

the bound can be further simplified. If we have

$$\left(\frac{48\sqrt{\varepsilon_0}}{16 C_\phi} \right)^{1/4} \leq \gamma,$$

then (40) can be bounded by

$$\left| \mathbb{E} \left[\sum_{m=1}^{M} (a_{m,i} - a_{m,K})(a_{m,j} - a_{m,K}) \widehat{h}_m''(\boldsymbol{A}\boldsymbol{s}_\ell) \right] \right| \leq \frac{\sqrt{3C_\phi}\varepsilon_0^{1/4}}{6}.$$

Given that N is fixed, one may pick $\varepsilon = \sqrt{\frac{C^2 \log(2d_{\mathcal{F}}/\delta)}{2N}}$, which gives the conclusion that if

$$\gamma \geq \left(\frac{3}{C_\phi} \right)^{1/4} \left(\frac{C^2 \log\left(\frac{2d_{\mathcal{F}}}{\delta}\right)}{2N} \right)^{1/16},$$

we have

$$\left| \mathbb{E} \left[\sum_{m=1}^{M} (a_{m,i} - a_{m,K})(a_{m,j} - a_{m,K}) \widehat{h}_m''(\boldsymbol{A}\boldsymbol{s}_\ell) \right] \right| \leq \frac{\sqrt{3C_\phi} \left(\frac{C^2 \log\left(\frac{2d_{\mathcal{F}}}{\delta}\right)}{2N} \right)^{1/8}}{6}. \tag{41}$$

16 Proof of Proposition 2

First, define $\mathcal{C}_{\boldsymbol{\theta}^*}(\boldsymbol{x}_\ell) = \mathbf{1}^\top \widehat{\boldsymbol{f}}(\boldsymbol{x}_\ell) - 1$ where $\widehat{\boldsymbol{f}}$ is a solution of (15). For neural network $\mathcal{F}$ defined in Assumption 4, by selecting ϵ as in Eq. (30), when

$$N = \Omega \left(\frac{M^4 z^4 B^8 C_x^4 \left(\sqrt{R} + \sqrt{\log(1/\delta)} \right)^2}{\epsilon^2} \right),$$

we have $\left| \mathbb{E}[\mathcal{C}_{\boldsymbol{\theta}^*}(\boldsymbol{x})] \right|^2 \leq \varepsilon$.

To bound $v_{\infty,\boldsymbol{\theta}^*}$, we need first to bound the Rademacher complexity of function $\boldsymbol{q} \circ \boldsymbol{f}$ with $q_m, f_m \in \mathcal{F}$ which is defined in Assumption 4. The Rademacher complexity of the function class $\mathcal{K} = \{q \circ f(x) \mid q, f \in \mathcal{F}\}$ could be bounded by

$$\mathfrak{R}_N(\mathcal{K}) \leq zB^2\mathfrak{R}_N(\mathcal{F}) \leq 2z^2B^4C_x\sqrt{\frac{R}{N}},$$

where the first inequality is by the function composition property of Rademacher complexity [3] and any function in $\mathcal{F}$ is zB^2-Lipschitz continuous.

Next, the Rademacher complexity of the function class $\mathcal{K}'$ such that $\mathcal{K}' = \{q \circ f(x) - x \mid q, f \in \mathcal{F}\}$, where $\mathcal{F}$ is defined in Assumption 4, is bounded by

$$\mathfrak{R}_N(\mathcal{K}') = \mathfrak{R}_N(\mathcal{K}) + 0 \leq 2z^2B^4C_x\sqrt{\frac{R}{N}},$$

due to the linearity of Rademacher complexity [3] and the fact that the function $-x$ is a singleton function, whose Rademacher complexity is 0.

Following similar proof of Lemma 5, the Rademacher complexity for the function class

$$\mathcal{K}'' = \left\{\|\boldsymbol{q}\left(\boldsymbol{f}(\boldsymbol{x})\right) - \boldsymbol{x}\|_2^2 \mid q_m, f_m \in \mathcal{F},\ \forall m = [M]\right\}$$

can be derived as follows. First, the function $|q_m \circ f_m(x_m) - x_m|$ is upper bounded by $z^2B^4C_x + C_x$. Thus we have

$$\mathfrak{R}_N(\mathcal{K}'') \leq 2M(z^2B^4C_x + C_x)\mathfrak{R}_N(\mathcal{K}') \leq 4Mz^2B^4C_x(z^2B^4C_x + C_x)\sqrt{\frac{R}{N}}.$$

Consequently,

$$\mathbb{E}\left[\|\boldsymbol{q}(\boldsymbol{f}(\boldsymbol{x})) - \boldsymbol{x}\|_2^2\right] \leq \frac{1}{N}\sum_{\ell=1}^{N}\|\boldsymbol{q}(\boldsymbol{f}(\boldsymbol{x}_\ell)) - \boldsymbol{x}_\ell\|_2^2 + 2\mathfrak{R}_N(\mathcal{K}'') + M(z^2B^4C_x + C_x)^2\sqrt{\frac{\log(1/\delta)}{2N}}.$$

Let $\epsilon = 2\mathfrak{R}_N(\mathcal{K}'') + M(z^2B^4C_x + C_x)^2\sqrt{\frac{\log(1/\delta)}{2N}}$, we have

$$N = \Omega\left(\frac{M^2z^8B^{16}C_x^4\left(\sqrt{R} + \sqrt{\log(1/\delta)}\right)^2}{\epsilon^2}\right).$$

For the finite function class, by assuming that $\|\boldsymbol{q}\left(\boldsymbol{f}(\boldsymbol{x})\right) - \boldsymbol{x}\|_2^2 \leq C_r$, we have (16) hold when $N \geq \max\left\{\frac{C^2\log(2d_{\mathcal{F}}/\delta)}{2\varepsilon^2}, \frac{C_r^2\log(2d_{\mathcal{F}}/\delta)}{2\varepsilon^2}\right\}$.

17 Analytical Tools

In this section, we introduce two analytical tools used in our proof, namely, the Hoeffding's inequality and the Rademacher complexity—depending on what kind of nonlinear function $\boldsymbol{f} \in \mathcal{F}$ that is used in the learning formulation (10). The Hoeffding's concentration theorem is as follows [4]:

Theorem 7 *Let $Z_1, \cdots, Z_N$ be sequence of i.i.d. random variables. Assume that $\mathbb{E}[Z_i] = \mu$ and $\mathbb{P}[a \leq Z_i \leq b] = 1$ for all i. Then for any $\epsilon > 0$*

$$\mathbb{P}\left[\left|\frac{1}{N}\sum_{i=1}^{N}Z_i - \mu\right| > \epsilon\right] \leq 2\exp\left(-2N\epsilon^2/(b-a)^2\right).$$

This will be used when we consider $f_m \in \mathcal{F}$ where $|\mathcal{F}| < \infty$.

We will also consider function classes where $|\mathcal{F}| = \infty$, in particular, neural networks. For this class, we will leverage the Rademacher complexity. Let $\mathcal{G}$ be a family of functions $\boldsymbol{g} : \mathcal{X} \in \mathbb{R}^M \to y \in \mathbb{R}$ where $y \in [a, b]$. Denote $\{\boldsymbol{x}_\ell\}_{\ell=1}^N$ a fixed i.i.d. sample of size N sampled from $\mathcal{X}$ following a certain distribution $\mathcal{D}$. Then, the empirical Rademacher complexity [3] of function class $\mathcal{G}$ with respect to $\{\boldsymbol{x}_\ell\}_{\ell=1}^N$ is defined as

$$\widehat{\mathfrak{R}}_{\boldsymbol{X}}(\mathcal{G}) = \mathbb{E}_{\boldsymbol{\sigma}}\left[\sup_{\boldsymbol{g}\in\mathcal{G}} \frac{1}{N}\sum_{\ell=1}^{N} \sigma_\ell \boldsymbol{g}(\boldsymbol{x}_\ell)\right],$$

where $\boldsymbol{X} = [\boldsymbol{x}_1, \ldots, \boldsymbol{x}_N]$, σ_i is independent uniform random variables taking $\{-1, +1\}$. Intuitively, for more complex families $\mathcal{G}$, the vectors $\boldsymbol{g}(\boldsymbol{x}_1), \ldots, \boldsymbol{g}(\boldsymbol{x}_N)$ are less similar to each other, and thus $\widehat{\mathfrak{R}}_{\boldsymbol{X}}(\mathcal{G})$ is expected to be larger—since the chance that $+\boldsymbol{g}(\boldsymbol{x}_\ell)$ and $-\boldsymbol{g}(\boldsymbol{x}_{\ell'})$ for $\ell \neq \ell'$ cancel each other is smaller.

The Rademacher complexity of $\mathcal{G}$ is the expectation of the empirical Rademacher complexity over all sample sets of size N following the same distribution:

$$\mathfrak{R}_N(\mathcal{G}) = \mathbb{E}_{\boldsymbol{X}\sim\mathcal{D}^N}\left[\widehat{\mathfrak{R}}_{\boldsymbol{X}}(\mathcal{G})\right],$$

where $\mathcal{D}^N$ denotes the joint distribution of N samples.

18 Additional Experiment

In Fig. 8 shows the runtime of the proposed method is plotted as M increases. In the experiment, we generate $5,000$ samples and the nonlinear functions for each dimension are randomly selected from $g_m(x) = \alpha\cdot\text{sigmoid}(x)+\beta x$ or $g_m(x) = \alpha\cdot\tanh(x)+\beta x$ with α and β drawn from the normal distribution and uniform distribution $[-0.5, 0.5]$, respectively. We use a one-hidden-layer network with 32, 64 and 128 neurons to model each dimension's $f_m(\cdot)$. We run the following settings with (K, M) being $(5, 5)$, $(5, 10)$, $(10, 20)$ and $(20, 30)$, respectively.

One can see that as M becomes larger, the runtime to reach $\frac{1}{N}\sum_{\ell=1}^{N} |\mathbf{1}^\top \boldsymbol{f}(\boldsymbol{x}_\ell) - 1|^2 < 10^{-5}$ increases rapidly when $R = 256$. The time increase is more moderate when $R = 128$ and $R = 64$. This shows a trade-off between the expressiveness of the employed neural network (i.e., larger R means that the corresponding neural network is more expressive) and the computational cost. How to better balance these two aspects is a meaningful future direction.

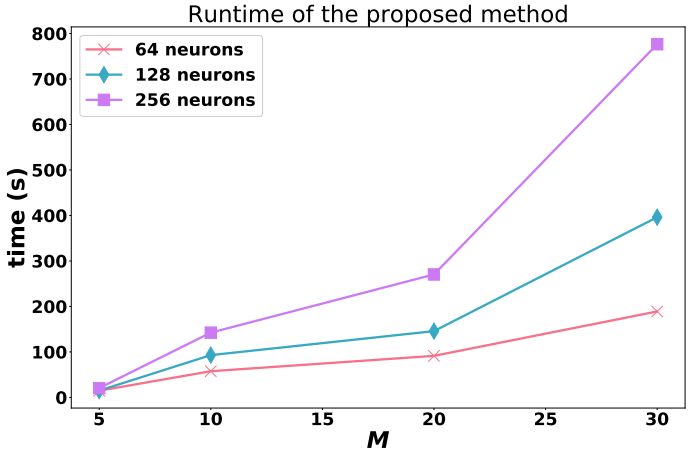

Figure 8: Runtime of the proposed method as M increases.

References

[1] M. Mohri, A. Rostamizadeh, and A. Talwalkar, *Foundations of Machine Learning*. MIT press, 2018.

[2] P. Liang, "CS229T/STAT231: Statistical Learning Theory," 2016.

[3] P. L. Bartlett and S. Mendelson, "Rademacher and Gaussian complexities: Risk bounds and structural results," *Journal of Machine Learning Research*, vol. 3, no. Nov, pp. 463–482, 2002.

[4] S. Shalev-Shwartz and S. Ben-David, *Understanding machine learning: From theory to algorithms*. Cambridge university press, 2014.