IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received May 20, 2018, accepted July 16, 2018, date of publication July 31, 2018, date of current version August 20, 2018.

Digital Object Identifier 10.1109/ACCESS.2018.2860683

Nonlinear Blind Source Separation Unifying Vanishing Component Analysis and Temporal Structure

LU WANG[1], (Student Member, IEEE), AND TOMOAKI OHTSUKI[2], (Senior Member, IEEE)
[1]Graduate School of Science and Technology, Keio University, Yokohama 223–8522, Japan
[2]Department of Information and Computer Science, Keio University, Yokohama 223–8522, Japan

Corresponding author: Lu Wang (wanglu@ohtsuki.ics.keio.ac.jp)

This work was supported by the Keio Leading-Edge Laboratory of Science and Technology, Japan, under Grant KEIO-KLL-000035.

ABSTRACT Nonlinear blind source separation (BSS) is one of the unsolved problems in unsupervised learning, because the solutions are highly non-unique when there is no prior information for the mixing functions. In this paper, we present a novel approach to tackle the ill-posedness of the nonlinear BSS problem with a few assumptions. The derivation of our algorithm is inspired by the idea of an efficient layer-by-layer representation to approximate the nonlinearity. Once such a representation is built, a final output layer is constructed by solving a convex optimization problem. Relying on the multi-layer architecture, the algorithm transforms a time-invariant nonlinear BSS to the local linear problem with a tolerable computational cost. Then, the projected data can break the nonlinear problem down into the version of a generalized joint diagonalization problem in the feature space. Importantly, the parameters and forms of polynomials depend solely on the input data, which guarantee the robustness of the structure. We thus address the general problem without being restricted to any specific mixture or parametric model. Experiments show that the proposed algorithm has a higher estimation accuracy on audio data sets from the real world for separating various nonlinear mixtures.

INDEX TERMS Nonlinear BSS, vanishing component analysis, temporal structure, statistical independent.

I. INTRODUCTION

The problems of independent component analysis (ICA), blind separation of source signals have received wide attention in various fields such as speech enhancement [1], image recognition [2], wireless communication [3], and thus have been thoroughly studied in the signal processing community. Usually, the original sources are linearly or nonlinearly mixed in some ways to produce a number of observations. BSS aims at recovering independent sources from their mixtures having access only to the observations without any prior knowledge, i.e., neither the sources nor the mixing matrix is known. The foundation assumption for linear blind source separation is that the statistical independence of the sources is usually sufficient to constrain the demixing functions up to the trivial transformations such as permutation and scaling.

An obvious extension for the task of BSS is that the observations are assumed to be generated from a set of sources by a nonlinear, instantaneous and invertible function. Roughly, the blind source separation seeks to find the mixing function or its inverse, solely based on the assumption that the sources are statistically independent. However, the indeterminacies imposed by the nonlinear model are difficult to handle [4]. Without extra constraints, the solutions are non-unique and then it suffers from the inability to recover the sources such as scaling and permutation [5]. In fact, there is an infinite number of possible nonlinear decompositions of a random vector into independent components, and those decompositions are not similar to each other in any trivial way [4]. The recovery inconsistency has been tackled by adding further prior information directly in the model or as a regularization term in the optimization processing procedure.

Various attempts [6]–[8] have been proposed to provide a theoretical understanding for solving the nonlinear mixing. Despite such progress, there are still many important open problems and unexplored areas, particularly in the nonlinear spaces and systems. The captured nonlinear features are in fact growing at an enormous rate. That necessitates higher advancement of algorithms and methods to extract models,

patterns, and knowledge from nonlinear mixing. For instance, the approach that captures the topology of the space from data points is represented in [9] and [10]. Studying of various aspects of data geometry including manifold learning have been proposed in [11].

One way relies on such a flexible approximation, including multi-layer perceptron (MLP) neural network [12], [13], which is employed for estimating the nonlinear separation transform function. By restricting the smoothness of the target transforming,[1] MLP provides the regularized solutions to ensure that nonlinear ICA leads to the sources separable. However, the example presented in [14] shows that the smoothness property is not a sufficient condition for this purpose. Hyvärinen and Pajunen [5] show conformal mapping[2] may helpful. Nonlinear ICA is able to estimate a separation mapping up to the rotation when the mapping functions are restricted to the set of conformal mapping. Unfortunately, the angle preservation conditions seem very restrictive [15]. In particular, it is not realistic in the framework of the nonlinear mappings associated with the nonlinear sensor array.

A. OUR CONTRIBUTION

We present a novel separation model that relies on the temporal structure and a novel mathematical construction with a multi-layer architecture. The approach pre-processes the data using a flexible approximation that projects the data into a high dimensional feature space. Then, by considering the temporal decorrelation as the separation criterion, we can break a nonlinear problem down into a version of the generalized joint diagonalization problem in the feature space.

The derivation of our algorithm is inspired by the idea of an efficient layer-by-layer representation to approximate such nonlinearity, which is referred to as Vanishing Ideal-based NonLinear SEparation Model (ViNLisem). By using vanishing component analysis (VCA) in [16], a prominent work in machine learning, we generate a set of polynomial functions that transform a time-invariant nonlinear BSS to the local linear problem. Such transformed components are used to extract the nonlinear mixture as the flexible approximation. Similar to a well-known principle in modern deep learning [17]–[19], the layers of our architectures are built one-by-one, creating higher-and-higher level representations of the data. Once such a representation is built, a final output layer is constructed by solving a convex optimization problem [20]. Based on the multi-layered architecture, the nonlinearity of the mixing model is depicted by such polynomials. Importantly, the parameters and forms of polynomials depend solely on the input data, which guarantee the robustness of the structure. We thus address the general problem without being restricted to any specific mixture or parametric model.

[1]The function f is a smooth transformation if its derivatives of any order always exist and they are continuous.

[2]The conformal mapping is defined as a mapping which preserves orientated angles. It is often considered in the framework of functions of complex-valued variables that are restricted to plane mapping. e.g., Joukovski mapping.

In particular, the layer-by-layer representation is adaptively generated solely on the observations. As the number of spanned spaces goes up, the computational complexity grows exponentially. To overcome this obstacle, relying on the properties of vanishing components, we provide a feasible way to narrow the size of the candidate polynomial set. We thus generate the polynomial in the current layer only from the spanned space of the last layer and that of the first layer, such as $\boldsymbol{g}^{(t)}(\mathcal{S})$ is generated from the span of $F_{t-1} \times F_1$ rather than considering all the extended spaces, i.e., $F_1, F_2, \cdots, F_{t-1}$. The details are shown in Theorem 2 and Theorem 3.

In addition, using the frameworks in [21], the local temporal structure of the transformations is taken into account. The contrast function is discriminative to be designed by emphasizing the difference from the temporally i.i.d. data. On the other hand, the criterion is formulated by minimizing the second-order statistics in which the transformed components and their time lags are statistically as independent as possible. Therefore, we can break a nonlinear problem down into the version of generalized joint diagonalization problem in the feature space.

The rest of the paper is structured as follows. In Section II, we introduce some related works. The preliminary and problem formulation are given in Section III. In Section IV, we present a novel approach used for nonlinear BSS algorithm and its analysis of properties. In Section V, we discuss the computational cost of the proposed algorithm. Section VI provides experimental results to illustrate the effectiveness of the proposed algorithm. We conclude the paper briefly in Section VII.

II. THE RELATIVE WORK

One of the earliest frameworks based on temporal structures is Temporal Decorrelation source SEParation method, which is abbreviated as TDSEP in [21]. It works on the temporal structure that the separated signal and its time lags are jointly taken into account for the independence of the sources. However, for most temporal blind source separation (TBSS) methods, how to select the optimal time lags is an important problem. In this paper, we are going to show how this framework can be extended to the nonlinear case rather than solving the problem of searching the optimal time lags.

A related but different idea is exploited in approximation using multi-kernel space. Harmeling *et al.* [22]–[24], a kernelized TDSEP (KTDSEP) method was proposed for nonlinear blind source separation that the kernel functions are used for mapping the observations into the kernel spaces. They show how kernel functions are employed to linear BSS methods to solve nonlinear source separation problems. These functions, however, do not have any optimizing property in terms of the contrast function that allows them to be ranked and evaluated. In addition, the method assumes the number of kernel spaces is chosen enough to approximate the nonlinearity without technical reasons. Sprekeler *et al.* [25] claim that temporal slowness complements statistical independence well, and a combination of

these principles leads to unique solutions of the nonlinear BSS problem.

Our construction and algorithm rely on the representation learning [26]. Heldt *et al.* [27] introduced a numerically stable approximate vanishing ideal algorithm. Livni *et al.* [28] defined a family of neural networks with polynomial activation functions that the polynomial features are learned as nonlinear combinations of the original signals. Donini and Aiolli [29], used a hierarchy of base kernel in the space of polynomial. These approaches consist of using an implicit map of the data, such as the Nyström method [30], random features [31] and sketching [32], [33]. That is features interactions in possibly high-dimensional data [34]. All of these approaches have in common with the flexible approximation, which emphasizes the representation learning as the key to the challenging nonlinear problems.

III. PRELIMINARY AND PROBLEM FORMULATION

The nonlinear BSS problem is formally described as follows. The observed signals $\mathbf{x}(t) = \{x_1(t), x_2(t), \cdots, x_n(t)\}^\top$ are assumed to be generated from a set of statistically independent sources $\mathbf{s}(t) = \{s_1(t), s_2(t), \cdots, s_m(t)\}^\top$ by a nonlinear, instantaneous and invertible function

$$\mathbf{x}(t) = \mathcal{F}(\mathbf{s}(t)), \quad t = 1, 2, \cdots, T, \tag{1}$$

where $\{\cdot\}^\top$ denotes the transpose, and t is the sample (time) index. Here, T is the total number of time points. n and m refer to the number of observed signals and sources, respectively. In this paper, we set $n = m$ in general. Since we are going to exploit only the statistical independence of the sources to be retrieved, a suitable approximation of the inverse nonlinear transformation could better reproduce the independence of the sources. Then some basic definitions are introduced for problem setup. Let $\boldsymbol{f} \circ \boldsymbol{h}$ denotes the Hadamard product, such as $\boldsymbol{f} \circ \boldsymbol{h} = [f_1 h_1, \cdots, f_k h_k]^\top$, where $\boldsymbol{f} = \{f_1, \cdots, f_k\}$ and $\boldsymbol{h} = \{h_1, \cdots, h_k\}$ are two arbitrary vectors.

Definition 1 (Polynomial): A function g: $\mathbb{R}^n \to \mathbb{R}$ is called as polynomial if the linear combination is $g(\mathbf{x}) = \sum_j \beta_j \mathbf{x}^{\boldsymbol{\alpha}(j)}$, where the coefficient $\beta_j \in \mathbb{R}$, $\mathbf{x} = [x_1, \cdots, x_n]^\top$, $\mathbf{x}^{\boldsymbol{\alpha}(j)} = \prod_{i=1}^n x_i^{\alpha_i(j)}$ and $\boldsymbol{\alpha}(j) = [\alpha_1(j), \cdots, \alpha_n(j)]^\top$. □

Definition 2 (Polynomial Ring): The polynomial ring with n variables over $\mathbb{R}$ is denoted as $\mathbb{R}[x_1, \cdots, x_n]$ that the addition and multiplication operators over the polynomial ring are equivalent to addition and multiplication of functions. □

Definition 3 (Ideal): Let I be a set of polynomials in $\mathbb{R}[x_1, \cdots, x_n]$, where $\mathbb{R}[x_1, \cdots, x_n]$ is a polynomial ring with n variables. For $\forall f \in I$ and $g \in \mathbb{R}[x_1, \cdots, x_n]$. If $fg \in I$ holds, then I is defined as an ideal. □

Definition 4 (Set of Generators): Let I be an ideal. If $\forall f \in I$ there exist $h_1, \cdots, h_k \in \mathbb{R}[x_1, \cdots, x_n]$ and a set of polynomials $\{g_1, \cdots, g_k\} \subseteq I$ such that $f = \sum_i g_i h_i$, then $\{g_1, \cdots, g_k\}$ is said to generate I. □

Definition 5 (Vanishing Ideal): Given a dataset $\mathcal{S} \subset \mathbb{R}^n$, for all $\mathbf{x} \in \mathcal{S}$, the vanishing ideal of $\mathcal{S}$ is the set of polynomials that vanish on $\mathcal{S}$. i.e. $g \in I(\mathcal{S})$ iff $g(\mathbf{x}) = 0$ for $\forall \mathbf{x} \in \mathcal{S}$. □

The problem can be set up as follows. We have a set of observed signals $\mathcal{S} = \{\mathbf{x}(t)\}_{t=1}^T$ that are generated from (1). The objective is to estimate the original sources $\mathbf{s}(t)$ and the mixing functions $\mathcal{F}$ (or it's inverse function $\mathcal{G} = \mathcal{F}^{-1}$) by using the observed signals $\mathbf{x}(t)$ only.

However, without any extra constraints, the solutions of blind source separation are non-unique [5]. In this paper, a novel approach is proposed by utilizing a flexible approximation to estimate the nonlinearity of the mixing function. First, let us focus on the representation learning [26]: how can we construct a structure that provides a good approximation basis for the values attained by vanishing polynomials.

Problem 1: Given an input dataset $\mathcal{S} = \{\mathbf{x}(t)\}_{t=1}^T$, where $\mathbf{x}(t) = [x_1(t), \cdots, x_n(t)]^\top$ and $\mathcal{S} \subset \mathbb{R}^n$. The problem is to learn a set of vanishing polynomials, which is formulated as the following optimization problem

$$\begin{aligned} &\min_V \ \dim(V) \\ &\text{subject to } V = \{g_i(\mathbf{x}) = 0 \,|\, \mathbf{x} \in \mathcal{S}\}, \end{aligned} \tag{2}$$

for $i = 1, \cdots, k$, where $\dim(\cdot)$ represents the dimension of variables and V denotes the set of vanishing polynomial. Since the real data are noisy that allow us to consider a tolerate value ϵ, such that the polynomials almost vanish on $\mathcal{S}$, i.e., $\|g_i(\mathbf{x})\| \leq \epsilon$ for $\forall \mathbf{x} \in \mathcal{S}$ is satisfied, where $\|\cdot\|$ denotes the Euclidean norm. □

In Problem 1, we prefer to seek a set of polynomials such that $g_i(\mathbf{x}) \approx 0$ for all i and $\mathbf{x} \in \mathcal{S}$. These polynomials may provide a sufficient characterization of elements in $\mathcal{S}$. By utilizing the generators of vanishing polynomials, any nonlinear mixture can be approximated with the combination of coefficients and the monomials. However, such polynomials did not achieve the inversion of the $\mathcal{F}$ function directly. They provide more features with a different selection of vanishing polynomials. Finally, the sources are recovered by solving a joint diagonalization problem in the feature space.

The procedure is implemented by finding a set of polynomials $g_1(\mathbf{x}), \cdots, g_k(\mathbf{x})$ that satisfy $\|g_i(\mathbf{x})\| \leq \epsilon$ for all $i = 1, \cdots, k$ and $\mathbf{x} \in \mathcal{S}$. Given a dataset $\mathcal{S}$, the vanishing ideal is denoted as $I(\mathcal{S})$, which is a set of polynomials vanished on $\mathcal{S}$, i.e., $g \in I(\mathcal{S})$ iff $\|g(\mathbf{x})\| \leq \epsilon$ for $\forall \mathbf{x} \in \mathcal{S}$. If a set of polynomials can generate $I(\mathcal{S})$, then this set of polynomials is referred to as a set of generators for $I(\mathcal{S})$. Hilbert basis theorem in [35] told us that a finite set of generators exists for any ideal. A finite set of generators of the ideal is an attractive mechanism for describing $I(\mathcal{S})$, since all the elements in $I(\mathcal{S})$ can be derived from this set of generators. Thus, the mixing function $\mathcal{F}$ can be approximated by finding such a finite set of generators, whose elements are named as vanishing polynomials.

Using the vanishing polynomials, the projected signals take the form of $\boldsymbol{\phi}(\mathbf{x}(t))$ that is the projection of $\mathbf{x}(t)$ in the high-dimensional feature space. The demixing process can be expressed by a linear combination of these projected signals in the following formulation.

Problem 2: Let $\{\mathbf{x}(t)\}_{t=1}^{T}$ be a set of observed signals. There is a set of polynomials g_i such that $\{g_i(\mathbf{x}(t))\}_{i=1}^{k}$ form a basis of $\mathbb{R}^n$. By using such polynomials g_i, the projected data of $\mathbf{x}(t)$ in feature space denoted as $\boldsymbol{\phi}(\mathbf{x}(t)) = \{\phi_1(\mathbf{x}(t)), \cdots, \phi_k(\mathbf{x}(t))\}$. Since the original sources $\mathbf{s}(t)$ are mutually independent, there exist a coefficient matrix $\mathbf{W}$ so as to

$$\arg\min_{\mathbf{W}} \sum_{i\neq j} \mathbf{W}_{i,:}\boldsymbol{\Sigma}_{\boldsymbol{\phi}}\mathbf{W}_{j,:}^{\top} + \sum_{i\neq j}\sum_{l=1}^{N} \mathbf{W}_{i,:}\boldsymbol{\Sigma}_{\boldsymbol{\tau}_l}\mathbf{W}_{j,:}^{\top}, \tag{3}$$

where $\mathbf{W}_{i,:}$ and $\mathbf{W}_{j,:}$ are the i-th and j-th row of matrix $\mathbf{W}$, respectively. The matrices $\boldsymbol{\Sigma}_{\boldsymbol{\phi}} = \mathbb{E}[\boldsymbol{\phi}(\mathbf{x}(t))\boldsymbol{\phi}(\mathbf{x}(t))^{\top}]$ and $\boldsymbol{\Sigma}_{\boldsymbol{\tau}_i} = \mathbb{E}[\boldsymbol{\phi}(\mathbf{x}(t))\boldsymbol{\phi}(\mathbf{x}(t+\tau_i))^{\top}]$ are defined as the covariance matrix of $\boldsymbol{\phi}(\mathbf{x}(t))$ and the covariance matrix with time lags τ_i, respectively. Thus, the signal is defined by

$$\tilde{s}_j(t) = \sum_{i=1}^{k} W_{ji}\phi_i(\mathbf{x}(t)), \tag{4}$$

for $j = 1, \cdots, k$, where W_{ji} denotes the (j, i)-th element of the coefficient matrix $\mathbf{W}$. □

Problem 2 implies that if we build a set of vanishing components, which computes such k polynomials $g_1, \cdots, g_k$, then we can recover the signals $\tilde{\mathbf{s}}(t)$ with k dimensions. Due to $k > n$, we need to select n sources from $\tilde{\mathbf{s}}(t)$, which construct the estimation of the original sources $\mathbf{s}(t)$.

Problem 3: Let $\tilde{\mathbf{s}}(t) = [\tilde{s}_1(t), \cdots, \tilde{s}_k(t)]^{\top}$ be a set of recovered signals. Since the original sources $\mathbf{s}(t)$ are mutually independent, it is also independent if the separation process in (3) is applied again to the signal $\tilde{\mathbf{s}}(t)$. i.e., $\tilde{\mathbf{s}}'(t) = \mathbf{W}'\tilde{\mathbf{s}}(t)$, where $\mathbf{W}'$ is another coefficient matrix if joint diagonalization approach is applied to the signal $\tilde{s}(t)$ again. Then, the recovered sources $\hat{\mathbf{s}}(t)$ corresponds to the first n maximum correlations (corr) in $\tilde{\mathbf{s}}(t)$

$$\begin{aligned} \hat{\mathbf{s}}(t) &= \tilde{\mathbf{s}}_{\boldsymbol{\pi},:}(t), \quad t = 1, \cdots, T, \\ \text{subject to } \boldsymbol{\pi} &= \Upsilon(\boldsymbol{\theta}; n), \\ \boldsymbol{\theta} &= \Xi_{\max}\left\{\text{corr}\left(\tilde{\mathbf{s}}(t), \tilde{\mathbf{s}}'(t)\right)\right\}, \end{aligned} \tag{5}$$

where $\tilde{\mathbf{s}}_{\boldsymbol{\pi},:}(t)$ is the vector composed of elements from $\tilde{\mathbf{s}}(t)$ indicated with the index $\boldsymbol{\pi}$. $\boldsymbol{\pi}$ is the index number of output of $\Upsilon(\boldsymbol{\theta}; n)$ that is the function to choose the maximum n values of vector $\boldsymbol{\theta}$. $\Xi_{\max}\{\text{corr}\left(\tilde{\mathbf{s}}(t), \tilde{\mathbf{s}}'(t)\right)\}$ is a function to output a vector $\boldsymbol{\theta}$ with each element being as the maximum value of each row of the matrix $\text{corr}\left(\tilde{\mathbf{s}}(t), \tilde{\mathbf{s}}'(t)\right)$.

Figure 1 shows an intuitive example for nonlinear separation using the mapping functions. Since the observations are nonlinearly mixed in the input space, we need to resort to a flexible approximation that can extract the nonlinear characteristics in the manifold $\mathcal{G}$. Here, vanishing components allow us to construct the nonlinear variants by some polynomials, such as $g_1(\mathbf{x}(t)), \cdots, g_k(\mathbf{x}(t)) \in \mathcal{G}$. *i.e.*, the data $\mathbf{x}(t)$ are mapped implicitly into the feature space that denoted as $\boldsymbol{\phi}(\mathbf{x}(t)) = [\phi_1(\mathbf{x}(t)), \cdots, \phi_k(\mathbf{x}(t))]^{\top} = [g_1(\mathbf{x}(t)), \cdots, g_k(\mathbf{x}(t))]^{\top}$. The feature space is spanned from such polynomials that enable us to work on $\mathcal{G}$. Then BSS

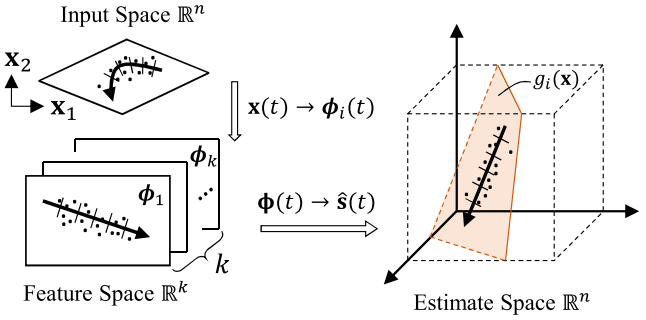

FIGURE 1. Input data $\mathbf{x}(t)$ are mapped to the manifold of $\mathcal{G} \in \mathbb{R}^k$, which is a feature space constructed by some polynomials $\{g_1, \cdots, g_k\} \subset \mathcal{G}$. Therefore, the projected points $\phi(\mathbf{x}(t))$ in feature space can make the problem linearly separable. The linear coefficient matrices in the feature space correspond to nonlinear coefficient matrices in the input space.

approaches can be applied to the projected data in the feature space, which corresponds to the nonlinear BSS approaches in the input space. Finally, due to $k > n$, we need to select n sources, which construct the estimation of the original sources $\mathbf{s}(t)$ in the estimated space. Since the parameters of the polynomials depend solely on the input data, it guarantees the robustness of the structure.

IV. NONLINEAR SEPARATION MODEL

We now turn to develop our nonlinear separation model as well as the accompanying analysis. We do the algorithm in the following stages. First, we derive a flexible approximation with multi-layer architecture, which runs in a set of polynomials that approximately equal to the value of zero. Thus the projected data in the feature space can make the problem linearly separable. Then, by taking into account the temporal structure served as a separation criterion, we can break the nonlinear problem down into a joint diagonalization problem in the feature space.

A. STRUCTURE OF MULTI-LAYER ARCHITECTURE

In order to perform a simple linear separation problem in feature space that corresponds to the nonlinear problem in input space, we need to specify how to map inputs $\mathbf{x}(1), \cdots, \mathbf{x}(T) \in \mathbb{R}^n$ into the feature space $\mathbb{R}^k$. A similar way is the kernel-based TDSEP presented by Harmeling *et al.* [24]. The difference is that our proposed method adapts to generate the polynomials, rather than assuming the number of approximate functions is chosen enough to represent the nonlinearity.

To ensure that a set of generators of $I(\mathcal{S})$ carry significant information about the input, we require the generators to be uncorrelated and the coefficients being in the null space of the matrix, which is composed of the monomials with different degree. Mathematically, this can be stated as follows.

Proposition 1: Denote the set of monomials over n variables with total degree up to d by $\mathcal{T}_d^n$. Consider the set of monomials $\mathcal{T}_d^n$ and the matrix $\mathbf{A}$ of size $T \times |\mathcal{T}_d^n|$ as follows: $\mathbf{A}_{ij} = t_j(\mathbf{x}(i))$, where $t_j(\mathbf{x}(i))$ is the j^{th} monomials in $\mathcal{T}_d^n$, which is composed of elements from $\mathbf{x}(i)$. Let $\boldsymbol{\beta}_1, \cdots, \boldsymbol{\beta}_k$

be a basis of the null space of matrix $\mathbf{A}$. Namely, for all $i = 1, \cdots, k$, we have $\mathbf{A}\boldsymbol{\beta}_i = \mathbf{0}$ and any vector $\boldsymbol{\beta}$ that satisfies $\mathbf{A}\boldsymbol{\beta} = \mathbf{0}$ can be written as a linear combination of $\boldsymbol{\beta}_i$. Then the polynomials $f_i(\mathbf{x}) = \sum_{j=1}^{|\mathcal{T}_d^n|} \beta_{ij} t_j(\mathbf{x})$, $i = 1, \cdots, k$ form a set of generators of $I(\mathcal{S})$, where β_{ij} is the coefficient for the i-th polynomial function and j-th monomial. □

Proof: Since $\mathbf{A}\boldsymbol{\beta}_i = \mathbf{0}$ is satisfied for all $i = 1, \cdots, k$, we have $f_i(\mathbf{x}) = \sum_{j=1}^{|\mathcal{T}_d^n|} \beta_{ij} t_j(\mathbf{x}) = 0$. Thus, $f_i(\mathbf{x}) \in I(\mathcal{S})$. Consider any polynomial $g(\mathbf{x})$ in the set of $I(\mathcal{S})$. Denote the coefficients for the polynomial $g(\mathbf{x})$ by $\mathbf{z} \in \mathbb{R}^{|\mathcal{T}_d^n|}$ such that the coefficients satisfy $\mathbf{A}\mathbf{z} = \mathbf{0}$. Then we have $g(\mathbf{x}) = \sum_{j=1}^{|\mathcal{T}_d^n|} z_j t_j(\mathbf{x}) = 0$. Since $\boldsymbol{\beta}_1, \cdots, \boldsymbol{\beta}_k$ is a basis of the null space of matrix $\mathbf{A}$, the coefficient vector $\mathbf{z}$ can be written as a linear combination of $\boldsymbol{\beta}_i$ as $\mathbf{z} = \sum_{i=1}^{k} \alpha_i \boldsymbol{\beta}_i$, which we also have $z_j = \sum_{i=1}^{k} \alpha_i \boldsymbol{\beta}_{ij}$. Then the polynomial $g(\mathbf{x})$ can be written by $g(\mathbf{x}) = \sum_{j=1}^{|\mathcal{T}_d^n|} z_j t_j(\mathbf{x}) = \sum_{j=1}^{|\mathcal{T}_d^n|} \sum_{i=1}^{k} \alpha_i \boldsymbol{\beta}_{ij} t_j(\mathbf{x}) = \sum_{i=1}^{k} \alpha_i f_i(\mathbf{x})$. Thus, the polynomials $f_i(\mathbf{x})$ form a set of generators of $I(\mathcal{S})$. □

The above procedure achieves the goal of finding a set of generators of $I(\mathcal{S})$. Since the real data are noisy that allow us to consider a tolerate value ϵ, such that the polynomials almost vanish on $\mathcal{S}$ if $\boldsymbol{g}(\mathbf{x}) \leq \epsilon$ is satisfied.

1) POLYNOMIALS OF DEGREE 1

If the vanishing polynomial is applied to the whole data $\mathcal{S}$, we have

$$\boldsymbol{g}^{(1)}(\mathcal{S}) = \left[g^{(1)}(\mathbf{x}(1)), \cdots, g^{(1)}(\mathbf{x}(T))\right]^\top = \mathbf{0}_{T\times 1}, \tag{6}$$

where $\mathcal{S} = \{\mathbf{x}(1), \cdots, \mathbf{x}(T)\}$. Firstly, the linear polynomial can be expressed as the combination of vector $\mathbf{x}(t)$ with the coefficient $\boldsymbol{\beta} \in \mathbb{R}^{n+1}$ such that

$$g^{(1)}(\mathbf{x}(t)) = \beta_0 + \sum_{i=1}^{n} \beta_i x_i(t) = \sum_{i=0}^{n} \beta_i \rho_i(\mathbf{x}(t)), \tag{7}$$

where $x_i(t)$ is the i-th element for the observations $\mathbf{x}(t)$ and $\rho_i(\mathbf{x}(t)) = x_i(t)$ for convenience. Thus, $\rho_0(\mathbf{x}(t)) = 1$ for all $\mathbf{x}(t)$. It follows that for any such polynomial we have

$$\begin{aligned}\boldsymbol{g}^{(1)}(\mathcal{S}) &= \begin{bmatrix} g^{(1)}(\mathbf{x}(1)) \\ \vdots \\ g^{(1)}(\mathbf{x}(T)) \end{bmatrix} = \begin{bmatrix} \sum_{i=0}^{n} \beta_i \rho_i(\mathbf{x}(1)) \\ \vdots \\ \sum_{i=0}^{n} \beta_i \rho_i(\mathbf{x}(T)) \end{bmatrix} \\ &= \sum_{i=0}^{n} \beta_i \boldsymbol{\rho}_i(\mathcal{S}),\end{aligned} \tag{8}$$

where $\boldsymbol{\rho}_i(\mathcal{S}) = [\rho_i(\mathbf{x}(1)), \cdots, \rho_i(\mathbf{x}(T))]^\top$.

Theorem 1: The polynomial $\boldsymbol{g}^{(1)}(\mathcal{S})$ vanishes on dataset $\mathcal{S}$ if and only if $\boldsymbol{g}^{(1)}(\mathcal{S}) = \mathbf{0}_{T\times 1}$, which requires the vector $\boldsymbol{\beta}$ would be in the null space of the $T \times (n+1)$ matrix $\mathbf{A}_1 = [\boldsymbol{\rho}_0(\mathcal{S}), \cdots, \boldsymbol{\rho}_n(\mathcal{S})]$ as

$$\mathbf{A}_1\boldsymbol{\beta} = [\boldsymbol{\rho}_0(\mathcal{S}), \cdots, \boldsymbol{\rho}_n(\mathcal{S})]\boldsymbol{\beta} = \mathbf{0}_{T\times 1}. \tag{9}$$

Then the vanishing polynomials can be obtained by searching the null space of $\mathbf{A}_1$. We maintain two sets for polynomials of degree 1: V_1 for the vanishing polynomials and F_1 for the non-vanishing polynomials. We use the notation $F_1 = \{\boldsymbol{\rho}(\mathcal{S}) : \boldsymbol{\rho} \in F_1\} \subset \mathbb{R}^T$ to denote the vectors in $\mathbb{R}^n$. We will construct F_1 such that F_1 is a set of orthogonal vectors in $\mathbb{R}^T$. Algorithm 1 describes the procedure to generate the vanishing and non-vanishing polynomials of degree 1 by the Gram-Schmidt procedure.

Algorithm 1 Generate Polynomials of Degree 1 by Gram-Schmidt Procedure

Initialization:

$F_1 = \{\boldsymbol{\rho}_0(\mathcal{S})\}$, where $\boldsymbol{\rho}_0(\mathcal{S}) = [1/\sqrt{n}, \cdots, 1/\sqrt{n}]^\top$;
$V_1 = \emptyset$;
$C_1 = \{\boldsymbol{\rho}_1(\mathcal{S}), \boldsymbol{\rho}_2(\mathcal{S}), \cdots, \boldsymbol{\rho}_n(\mathcal{S})\}$, where $\boldsymbol{\rho}_i(\mathcal{S}) = [\rho_i(\mathbf{x}(1)), \cdots, \rho_i(\mathbf{x}(T))]^\top$.

for $i = 1$ to n **do**
  $\boldsymbol{g}_i^{(1)}(\mathcal{S}) = \boldsymbol{\rho}_i(\mathcal{S}) - \sum_{\boldsymbol{\rho}\in F_1} \langle \boldsymbol{\rho}_i(\mathcal{S}), \boldsymbol{\rho}(\mathcal{S})\rangle \boldsymbol{\rho}(\mathcal{S})$
  if $\|\boldsymbol{g}_i^{(1)}(\mathcal{S})\| \leq \epsilon$ **then**
    $V_1 \leftarrow V_1 \bigcup \{\boldsymbol{g}_i^{(1)}(\mathcal{S})\}$
  else
    $F_1 \leftarrow F_1 \bigcup \{\boldsymbol{g}_i^{(1)}(\mathcal{S})/\|\boldsymbol{g}_i^{(1)}(\mathcal{S})\|\}$
  end if
end for

Output:

Vanishing polynomial set V_1;
Non-vanishing polynomials set F_1.

Considering a polynomial of degree 0, $\boldsymbol{\rho}_0(\mathcal{S}) = \mathbf{1}_{T\times 1}$ is clearly non-vanishing. We initialize $F_1 = \{\boldsymbol{\rho}_0(\mathcal{S})/\|\boldsymbol{\rho}_0(\mathcal{S})\|\}$, where $\|\cdot\|$ denotes the norm of the vector. And set $V_1 = \emptyset$ initially. Set C_1 to be a candidate set of polynomials, which is composed of polynomials of degree 1, such as $C_1 = \{\boldsymbol{\rho}_1(\mathcal{S}), \boldsymbol{\rho}_2(\mathcal{S}), \cdots, \boldsymbol{\rho}_n(\mathcal{S})\}$. To obtain the non-vanishing polynomials orthogonal to each other, it requires

$$\boldsymbol{g}_i^{(1)}(\mathcal{S}) = \boldsymbol{\rho}_i(\mathcal{S}) - \sum_{\boldsymbol{\rho}\in F_1} \langle \boldsymbol{\rho}_i(\mathcal{S}), \boldsymbol{\rho}(\mathcal{S})\rangle \boldsymbol{\rho}(\mathcal{S}). \tag{10}$$

Since the non-vanishing polynomial set F_1 only contains one element $\boldsymbol{\rho}_0(\mathcal{S})$ initially, the above equation can be simply represented as

$$\boldsymbol{g}_i^{(1)}(\mathcal{S}) = \boldsymbol{\rho}_i(\mathcal{S}) - \langle \boldsymbol{\rho}_i(\mathcal{S}), \boldsymbol{\rho}_0(\mathcal{S})\rangle \boldsymbol{\rho}_0(\mathcal{S}), \tag{11}$$

where $\langle \boldsymbol{\rho}_i(\mathcal{S}), \boldsymbol{\rho}_0(\mathcal{S})\rangle$ is the coefficient for $\boldsymbol{\rho}_0(\mathcal{S})$. We can now reformulate (11) in terms of a dual representation as

$$\begin{aligned}\boldsymbol{g}_i^{(1)}(\mathcal{S}) = [\boldsymbol{\rho}_0(\mathcal{S}), \boldsymbol{\rho}_1(\mathcal{S}), \cdots, \boldsymbol{\rho}_i(\mathcal{S}), \cdots, \boldsymbol{\rho}_n(\mathcal{S})] \cdot \\ \times [-\langle \boldsymbol{\rho}_i(\mathcal{S}), \boldsymbol{\rho}_0(\mathcal{S})\rangle, 0, \cdots, 1, \cdots, 0]^\top.\end{aligned} \tag{12}$$

Compared with (9) in Theorem 1, the vector $\boldsymbol{\beta}$ is given in the form $\boldsymbol{\beta} = [-\langle \boldsymbol{\rho}_i(\mathcal{S}), \boldsymbol{\rho}_0(\mathcal{S})\rangle, 0, \cdots, 1, \cdots, 0]^\top$. If a proper coefficient vector $\boldsymbol{\beta}$ can be searched so as to $\boldsymbol{g}_i(\mathcal{S})$ vanish on the data $\mathcal{S}$, we update $V_1 \leftarrow V_1 \bigcup \{\boldsymbol{g}_i^{(1)}(\mathcal{S})\}$. Otherwise, $F_1 \leftarrow F_1 \bigcup \{\boldsymbol{g}_i^{(1)}(\mathcal{S})/\|\boldsymbol{g}_i^{(1)}(\mathcal{S})\|\}$ is updated, where the normalization ensures that all the vectors in F_1 are orthonormalization as the normalized vectors. At the end of

this process, F_1 contains a set of linear polynomials which are non-vanishing on $\mathcal{S}$ and V_1 contains a set of linear polynomials that vanish on $\mathcal{S}$.

2) POLYNOMIALS OF DEGREE 2

To exploit the polynomials of degree 2, we need to construct a candidate set of polynomials $C_2 = \{\boldsymbol{\rho}_{i,j}(\mathcal{S})\}_{i,j=1}^{n}$, where $\boldsymbol{\rho}_{i,j}(\mathcal{S}) = [\rho_{i,j}(\mathbf{x}(1)), \cdots, \rho_{i,j}(\mathbf{x}(T))]^\top$ and $\rho_{i,j}(\mathbf{x}(t)) = x_i(t)x_j(t)$ for all i, j. Each polynomial of degree 2 takes the form

$$g^{(2)}(\mathbf{x}(t)) = \sum_{i=0}^{n} \beta_i \rho_i(\mathbf{x}(t)) + \sum_{i,j=1}^{n} \beta_{i,j}\rho_{i,j}(\mathbf{x}(t)). \quad (13)$$

By considering all the data points in $\mathcal{S}$, we have

$$\begin{aligned}\boldsymbol{g}^{(2)}(\mathcal{S}) &= \left[g^{(2)}(\mathbf{x}(1)), \cdots, g^{(2)}(\mathbf{x}(T))\right]^\top \\ &= \begin{bmatrix} \sum_{i=0}^{n} \beta_i \rho_i(\mathbf{x}(1)) + \sum_{i,j=1}^{n} \beta_{i,j}\rho_{i,j}(\mathbf{x}(1)) \\ \vdots \\ \sum_{i=0}^{n} \beta_i \rho_i(\mathbf{x}(T)) + \sum_{i,j=1}^{n} \beta_{i,j}\rho_{i,j}(\mathbf{x}(T)) \end{bmatrix} \\ &= \sum_{i=0}^{n} \beta_i \boldsymbol{\rho}_i(\mathcal{S}) + \sum_{i,j=1}^{n} \beta_{i,j}\boldsymbol{\rho}_{i,j}(\mathcal{S}). \end{aligned} \quad (14)$$

As before, we can find vanishing 2^{nd} order polynomials via the null space of the matrix: $\mathbf{A}_2 = [\mathbf{A}_1, \boldsymbol{\rho}_{1,1}(\mathcal{S}), \cdots, \boldsymbol{\rho}_{n,n}(\mathcal{S})]$. To find the null space of the matrix $\mathbf{A}_2$, we could simply continue the Gram-Schmidt procedure that we have already performed for the columns of $\mathbf{A}_1$. However, we now need to consider $n^2 + n + 1$ columns. As the degree goes up, the number of columns increases exponentially. To overcome this obstacle, relying on the properties of vanishing components, we provide an effective iterative approach to narrow the size of the candidate polynomial set.

Theorem 2: Let $\boldsymbol{g}^{(2)}(\mathcal{S})$ be a set of polynomials of degree 2. It can be constructed by two terms of degree 1 of the form $\boldsymbol{g}^{(2)}(\mathcal{S}) = \sum_{i_1,i_2} \boldsymbol{f}_{i_1}^{(1)} \circ \boldsymbol{f}_{i_2}^{(1)}$. Without loss of generality, assume that for $i_1, i_2 \le l$, where l is index number of polynomial of degree 1. We have that both $\boldsymbol{f}_{i_1}^{(1)}$ and $\boldsymbol{f}_{i_2}^{(1)}$ are non-vanishing on $\mathcal{S}$. For $i_1, i_2 > l$, either $\boldsymbol{f}_{i_1}^{(1)}$ or $\boldsymbol{f}_{i_2}^{(1)}$ vanishes. It follows that for all $i_1, i_2 > l$ we have that the polynomial $\boldsymbol{f}_{i_1}^{(1)} \circ \boldsymbol{f}_{i_2}^{(1)} = \mathbf{0}_{T\times 1}$. Thus, the polynomial $\hat{\boldsymbol{g}}^{(2)}(\mathcal{S}) = \sum_{i_1,i_2\le l} \boldsymbol{f}_{i_1}^{(1)} \circ \boldsymbol{f}_{i_2}^{(1)}$ satisfies $\hat{\boldsymbol{g}}^{(2)}(\mathcal{S}) = \boldsymbol{g}^{(2)}(\mathcal{S})$. $F_1 = \{\mathbf{p}_1^{(1)}, \cdots, \mathbf{p}_{|F_1|}^{(1)}\}$ is denoted as a non-vanishing polynomial set of degree 1, where $|F_1|$ denotes the number of elements included in the set F_1. Any polynomial of degree 1 that generated from F_1 can be expressed as

$$\boldsymbol{f}_{i_1}^{(1)} = \sum_{j_1} \alpha_{i_1,j_1}^{(1)} \mathbf{p}_{j_1}^{(1)}, \quad \boldsymbol{f}_{i_2}^{(1)} = \sum_{j_2} \alpha_{i_2,j_2}^{(1)} \mathbf{p}_{j_2}^{(1)}, \quad (15)$$

where $\alpha_{i_1,j_1}^{(1)}$ and $\alpha_{i_2,j_2}^{(1)}$ denote the coefficients that make $\boldsymbol{f}_{i_1}^{(1)} \circ \boldsymbol{f}_{i_2}^{(1)} \neq \mathbf{0}_{T\times 1}$ for all $i_1, i_2 \le l$. Then F_2 can be generated from the span of $\boldsymbol{f}_{i_1}^{(1)}$ and $\boldsymbol{f}_{i_2}^{(1)}$ for $i_1, i_2 \le l$ as

$$\begin{aligned}\hat{\boldsymbol{g}}^{(2)}(\mathcal{S}) &= \sum_{i_1,i_2\le l} \boldsymbol{f}_{i_1}^{(1)} \circ \boldsymbol{f}_{i_2}^{(1)} \\ &= \sum_{j_1,j_2} \left[\left(\mathbf{p}_{j_1}^{(1)} \circ \mathbf{p}_{j_2}^{(1)}\right) \left(\sum_{i_1,i_2\le l} \alpha_{i_1,j_1}^{(1)} \alpha_{i_2,j_2}^{(1)} \right) \right]. \end{aligned} \quad (16)$$

The operator $\circ$ denotes the Hadamard product, namely the vector $\boldsymbol{f}_{i_1}^{(1)} \circ \boldsymbol{f}_{i_2}^{(1)} = [f_{i_1,1}^{(1)} f_{i_2,1}^{(1)}, \cdots, f_{i_1,T}^{(1)} f_{i_2,T}^{(1)}]^\top$, where the degree of $\boldsymbol{f}_{i_1}^{(1)} = [f_{i_1,1}^{(1)}, \cdots, f_{i_1,T}^{(1)}]^\top$ and $\boldsymbol{f}_{i_2}^{(1)} = [f_{i_2,1}^{(1)}, \cdots, f_{i_2,T}^{(1)}]^\top$ are at most 1. □

Theorem 2 is proved in the Appendix A. It follows that $\hat{\boldsymbol{g}}^{(2)}(\mathcal{S})$ can be constructed from the span of $F_1 \times F_1$ and thus to construct F_2 and V_2, which suffices to find the null space and range on the set of candidate polynomials from $F_1 \times F_1$. Formally, let us redefine C_2 to be the set

$$C_2 = \left\{ \boldsymbol{\rho}_{i_1,i_2\le l}(\mathcal{S}) = \mathbf{p}_{j_1}^{(1)} \circ \mathbf{p}_{j_2}^{(1)} |\ \mathbf{p}_{j_1}^{(1)}, \mathbf{p}_{j_2}^{(1)} \in F_1 \right\}. \quad (17)$$

We will construct F_2 and V_2 by continuing a similar process with a polynomial of degree 1 on the candidate vectors of C_2. Note that, due to the particular structure of vanishing polynomials, as proposed in Theorem 2, $\boldsymbol{g}^{(2)}(\mathcal{S})$ can be generated from the span of $F_1 \times F_1$, i.e., from the polynomials with $i_1, i_2 \le l$ rather than the whole candidate vectors. Therefore, the remainder of $\boldsymbol{\rho}_{i_1,i_2\le l} \in C_2$ after projecting it on the current set F_2 is the polynomial $\boldsymbol{g}^{(2)}(\mathcal{S})$ defined by

$$\boldsymbol{g}^{(2)}(\mathcal{S}) = \boldsymbol{\rho}_{i_1,i_2\le l}(\mathcal{S}) - \sum_{\mathbf{p}^{(1)}\in F_2} \langle \boldsymbol{\rho}_{i_1,i_2\le l}(\mathcal{S}), \mathbf{p}^{(1)}(\mathcal{S})\rangle \mathbf{p}^{(1)}(\mathcal{S}). \quad (18)$$

It requires $|F_1| \times |F_1|$ times to evaluate all the polynomials in the candidate polynomial set C_2. Before we evaluate the polynomials of degree 2, we initialize F_2 and V_2 as $F_2 = F_1$ and $V_2 = V_1$. Then if $|\boldsymbol{g}^{(2)}(\mathcal{S})| \le \epsilon$, we have $\boldsymbol{g}^{(2)}(\mathcal{S})$ vanishes on $\mathcal{S}$. So we update $V_2 \leftarrow V_2 \bigcup \{\boldsymbol{g}^{(2)}(\mathcal{S})\}$. Otherwise, we update $F_2 \leftarrow F_2 \bigcup \{\boldsymbol{g}^{(2)}(\mathcal{S})/\|\boldsymbol{g}^{(2)}(\mathcal{S})\|\}$. At the end of this process, F_2 contains a set of polynomials of degree 1 and degree 2 that are non-vanishing on $\mathcal{S}$. In contrast, V_2 contains a set of polynomials of degree 1 and degree 2 that vanish on $\mathcal{S}$.

3) POLYNOMIALS WITH A HIGHER DEGREE

The above progress continues to a higher degree. For any polynomial of degree t, we prefer to construct the set of non-vanishing polynomials F_t only from the span of $F_{t-1} \times F_1$. At iteration t, the candidate polynomial set C_t is given in the form

$$C_t = \left\{ \boldsymbol{\rho}_{i_1,i_2,\cdots,i_t\le l}(\mathcal{S}) = \mathbf{p}_j^{(t-1)} \circ \mathbf{p}_{j_t}^{(1)} \right\}, \quad (19)$$

where $\mathbf{p}_j^{(t-1)} = \mathbf{p}_{j_1}^{(1)} \circ \mathbf{p}_{j_2}^{(1)} \cdots \circ \mathbf{p}_{j_{t-1}}^{(1)} \in F_{t-1}$ and $\mathbf{p}_{j_t}^{(1)} \in F_1$. For simple expression, the candidate polynomial set is written as $C_t = \{\boldsymbol{q}_1(\mathcal{S}), \cdots, \boldsymbol{q}_l(\mathcal{S})\}$. Then the above orthogonal processing can be given as

$$\boldsymbol{g}_i^{(t)}(\mathcal{S}) = \boldsymbol{q}_i(\mathcal{S}) - \sum_{\mathbf{p}^{(t-1)}(\mathcal{S})\in F_t} \langle \boldsymbol{q}_i(\mathcal{S}), \mathbf{p}^{(t-1)}(\mathcal{S})\rangle \mathbf{p}^{(t-1)}(\mathcal{S}). \quad (20)$$

The above processing procedure performs like a consecutive processing procedure that each time one polynomial is added to the vanishing polynomial set V_t or non-vanishing polynomial set F_t. Actually, we can operate more polynomials simultaneously with singular value decomposition (SVD). Before that, let us first introduce a property similar to Theorem 2.

Theorem 3: Let $\boldsymbol{g}^{(t)}(\mathcal{S})$ be a set of polynomials of degree t. It can be constructed as $\hat{\boldsymbol{g}}^{(t)}(\mathcal{S}) = \sum_{i_1,i_2,\cdots,i_t\leq l} \boldsymbol{f}_{i_1}^{(1)} \circ \boldsymbol{f}_{i_2}^{(1)} \circ \cdots \circ \boldsymbol{f}_{i_t}^{(1)}$. Assume that for $i_1, i_2, \cdots, i_t \leq l$, we have that $\boldsymbol{f}_{i_1}^{(1)}, \boldsymbol{f}_{i_2}^{(1)}, \cdots, \boldsymbol{f}_{i_t}^{(1)}$ are non-vanishing on $\mathcal{S}$. Denoting $F_{t-1} = \{\mathbf{p}_1^{(t-1)}, \cdots, \mathbf{p}_{|F_{t-1}|}^{(t-1)}\}$ and $F_1 = \{\mathbf{p}_1^{(1)}, \cdots, \mathbf{p}_{|F_1|}^{(1)}\}$ as a non-vanishing polynomial set of degree $t-1$ and 1, respectively. Then any polynomials $\hat{\boldsymbol{g}}^{(t)}(\mathcal{S})$ can be formulated as

$$\begin{aligned}\hat{\boldsymbol{g}}^{(t)}(\mathcal{S}) &= \sum_{i_1,i_2,\cdots,i_t\leq l} \boldsymbol{f}_{i_1}^{(1)} \circ \boldsymbol{f}_{i_2}^{(1)} \circ \cdots \circ \boldsymbol{f}_{i_t}^{(1)} \\ &= \sum_{j,j_t} \left[\left(\mathbf{p}_j^{(t-1)} \circ \mathbf{p}_{j_t}^{(1)}\right) \left(\sum_{i_t \leq l} \alpha_j^{(t-1)} \alpha_{i_t,j_t}^{(1)} \right) \right], \end{aligned} \tag{21}$$

where $\alpha_j^{(t-1)}$ and $\alpha_{i_t,j_t}^{(1)}$ denotes the coefficients that make $\mathbf{p}_j^{(t-1)} \circ \mathbf{p}_{j_t}^{(1)} \neq \mathbf{0}_{T\times 1}$ for all j, j_t. □

The theoretical proof is shown in the Appendix B. Then F_t can be generated from the span of $F_{t-1} \times F_1$. The matrix $\mathbf{A}_t$ can be formed as $\mathbf{A}_t = [\boldsymbol{g}_1^{(t)}(\mathcal{S}), \cdots, \boldsymbol{g}_{|F_t|}^{(t)}(\mathcal{S})]$. By using SVD, the matrix $\mathbf{A}_t$ can be decomposed as $\mathbf{A}_t = \mathbf{L}\mathbf{D}\mathbf{U}^\top$. Using a simple matrix operation, we then obtain

$$\mathbf{A}_t\mathbf{U} = \left[\boldsymbol{g}_1^{(t)}(\mathcal{S}), \cdots, \boldsymbol{g}_{|F_t|}^{(t)}(\mathcal{S})\right]\mathbf{U} = \mathbf{L}\mathbf{D}, \tag{22}$$

where $\boldsymbol{L} = [\boldsymbol{l}_1, \cdots, \boldsymbol{l}_T]$ and $\boldsymbol{l}_i \in \mathbb{R}^T$ for $i = 1, \cdots, T$. The above equation can be written as

$$\boldsymbol{\eta}_i^{(t)}(\mathcal{S}) = \sum_{j=1}^{|F_t|} U_{j,i} \boldsymbol{g}_j^{(t)}(\mathcal{S}) = \sum_{j=1}^{T} D_{j,i}\boldsymbol{l}_j = D_{i,i}\boldsymbol{l}_i, \tag{23}$$

where $i = 1, \cdots, |F_t|$. If $D_{i,i} < \epsilon$, we denote the polynomial $\boldsymbol{\eta}_i^{(t)}(\mathcal{S})$ vanishes, where ϵ is the tolerate value used to evaluate the polynomials how close to zero. Thus, we update $V_t \leftarrow V_t \bigcup \{\boldsymbol{\eta}_i^{(t)}(\mathcal{S})\}$. Otherwise we update $F_t \leftarrow F_t \bigcup \{\boldsymbol{\eta}_i^{(t)}(\mathcal{S}) / \|\boldsymbol{\eta}_i^{(t)}(\mathcal{S})\|\}$.

B. APPROXIMATE SIMULTANEOUS DIAGONALIZATION

After we obtain a set of polynomials that projected data in the feature space, we consider the blind source separation with temporal structure employed as the separation criterion. Thus, the nonlinear separation problem can be changed to a generalized joint diagonalization problem. An alternative technique proposed in [36] can achieve the process by implementing two steps: 1. whitening and 2. Constructing several Jacobi rotations to achieve an approximate simultaneous diagonalization of the correlation matrix set. In step 1, we find a linear transform, which can be determined by taking the inverse square root of the covariance matrix as

$$\boldsymbol{\Theta}_{\boldsymbol{\phi}} = \boldsymbol{\Sigma}_{\boldsymbol{\phi}(\mathbf{x}(t))}^{-\frac{1}{2}} = \left(\mathbb{E}\left[\boldsymbol{\phi}(\mathbf{x}(t))\boldsymbol{\phi}(\mathbf{x}(t))^\top\right]\right)^{-\frac{1}{2}}, \tag{24}$$

where $\boldsymbol{\phi}(\mathbf{x}(t)) = [g_1(\mathbf{x}(t)), \cdots, g_k(\mathbf{x}(t))]^\top$ and k is total number of vanishing polynomials. The transform $\boldsymbol{\Theta}_{\boldsymbol{\phi}}$ gives a representation of the signals $\boldsymbol{\phi}(\mathbf{x}(t))$ in a new basis and the transformed signals are denoted by $\mathbf{z}(t) = \boldsymbol{\Theta}_{\boldsymbol{\phi}}\boldsymbol{\phi}(\mathbf{x}(t)) = \boldsymbol{\Sigma}_{\boldsymbol{\phi}(\mathbf{x}(t))}^{-\frac{1}{2}}\boldsymbol{\phi}(\mathbf{x}(t))$. We defined a time-lagged correlation matrix of $\mathbf{z}(t)$ as

$$\begin{aligned}\boldsymbol{\Sigma}_{\mathbf{z}(\tau)} &= \mathbb{E}\left[\mathbf{z}(t)\mathbf{z}(t+\tau)^\top\right] \\ &= \boldsymbol{\Sigma}_{\boldsymbol{\phi}(\mathbf{x}(t))}^{-\frac{1}{2}} \mathbb{E}\left[\boldsymbol{\phi}(\mathbf{x}(t))\boldsymbol{\phi}(\mathbf{x}(t+\tau)^\top\right] \left(\boldsymbol{\Sigma}_{\boldsymbol{\phi}(\mathbf{x}(t))}^{-\frac{1}{2}}\right)^\top \\ &= \boldsymbol{\Theta}_{\boldsymbol{\phi}} \boldsymbol{\Sigma}_{\boldsymbol{\phi}(\tau)} \boldsymbol{\Theta}_{\boldsymbol{\phi}}^\top. \end{aligned} \tag{25}$$

With different time lag, we can have different correlation matrix as $\boldsymbol{\Sigma}_{\mathbf{z}(\tau_1)}, \boldsymbol{\Sigma}_{\mathbf{z}(\tau_2)}, \cdots, \boldsymbol{\Sigma}_{\mathbf{z}(\tau_N)}$, where N is the number of time lags. After the pre-whitening step, any time delayed correlation matrix can be transformed to a diagonal matrix by a rotation matrix $\mathbf{Q}$ as

$$\begin{cases} \boldsymbol{\Sigma}_{\mathbf{z}(\tau_1)} = \mathbf{Q}\boldsymbol{\Lambda}_{\mathbf{z}(\tau_1)}\mathbf{Q}^\top, \\ \boldsymbol{\Sigma}_{\mathbf{z}(\tau_2)} = \mathbf{Q}\boldsymbol{\Lambda}_{\mathbf{z}(\tau_2)}\mathbf{Q}^\top, \\ \quad\vdots \\ \boldsymbol{\Sigma}_{\mathbf{z}(\tau_N)} = \mathbf{Q}\boldsymbol{\Lambda}_{\mathbf{z}(\tau_N)}\mathbf{Q}^\top. \end{cases} \tag{26}$$

Concatenating both the whitening matrix $\boldsymbol{\Theta}_{\boldsymbol{\phi}}$ and the rotation matrix $\mathbf{Q}$ yields the demixing matrix as

$$\mathbf{W} = \mathbf{Q}^{-1}\boldsymbol{\Theta}_{\boldsymbol{\phi}} = \mathbf{Q}^{-1}\boldsymbol{\Sigma}_{\boldsymbol{\phi}(\mathbf{x}(t))}^{-\frac{1}{2}}. \tag{27}$$

Therefore, the signal $\tilde{\mathbf{s}}(t)$ can be expressed as

$$\tilde{\mathbf{s}}(t) = \mathbf{W}\boldsymbol{\phi}(\mathbf{x}(t)) = \mathbf{Q}^{-1}\boldsymbol{\Sigma}_{\boldsymbol{\phi}(\mathbf{x}(t))}^{-\frac{1}{2}}\boldsymbol{\phi}(\mathbf{x}(t)). \tag{28}$$

Note that the dimensions of $\tilde{\mathbf{s}}(t)$ and the original source $\mathbf{s}(t)$ are k and n respectively, where $k > n$. We need to select n sources from $\tilde{\mathbf{s}}(t)$, which construct the estimation of the original sources $\mathbf{s}(t)$. Considering all the projected components, we have the demixed signals $\tilde{\mathbf{S}} = \mathbf{Q}^{-1}\boldsymbol{\Theta}_{\boldsymbol{\phi}}\boldsymbol{\Phi}$, where $\boldsymbol{\Phi} = [\boldsymbol{\phi}(\mathbf{x}(1)), \cdots, \boldsymbol{\phi}(\mathbf{x}(T))]$ and $\tilde{\mathbf{S}} = [\tilde{\mathbf{s}}(1), \cdots, \tilde{\mathbf{s}}(T)]$. Since the original sources are mutually independent, the demixed sources should be also independent even if the demixed matrix is applied to the signal $\tilde{\mathbf{s}}(t)$ again. Therefore, we can obtain another set of signal $\tilde{\mathbf{S}}' = [\tilde{\mathbf{s}}'(1), \cdots, \tilde{\mathbf{s}}'(T)]$. By employing the above temporal structure on $\tilde{\mathbf{s}}(t)$, the correlation (corr) between each row in $\tilde{\mathbf{S}}$ and each row in $\tilde{\mathbf{S}}'$ is calculated by

$$\text{corr}\left(\tilde{\mathbf{s}}(t), \tilde{\mathbf{s}}'(t)\right) = \frac{\sum_{t=1}^{T}(\tilde{s}_i - \mathbb{E}[\tilde{s}_i])(\tilde{s}'_j - \mathbb{E}[\tilde{s}'_j])}{\sqrt{\sum_{t=1}^{T}(\tilde{s}_i - \mathbb{E}[\tilde{s}_i])^2}\sqrt{\sum_{t=1}^{T}(\tilde{s}'_j - \mathbb{E}[\tilde{s}'_j])^2}}. \tag{29}$$

Then, the rows in $\tilde{\mathbf{S}}$ with the maximum n correlations are denoted as the recovered sources $\hat{\mathbf{s}}(t)$.

V. COMPUTATIONAL COMPLEXITY

In this section, we analyze the computational complexity of the algorithm. Recalling our notations, we defined two sets: V and F are sets of vanishing polynomials and non-vanishing polynomials, respectively. The subscript of F denotes the subset of non-vanishing polynomials in the corresponding degree. For example, we use the notation $F_1 \subset F$ to denote the non-vanishing polynomials with degree 1 in $\mathbb{R}^T$. $F^{[r]} = \bigcup_{i \leq r} F_i$ is defined as the union of the collection F_i up to degree r. $|F_i|$ denotes the number of polynomials in the non-vanishing polynomial set F_i. In Algorithm 1, the progress will terminate at round r when the set F_r is empty. On the other hand, the progress does not stop, then $|F^{[r]}| \geq r$ holds for any $F^{[r]} = \bigcup_{i \leq r} F_i$, because F_i should contain at least one polynomial. Since $F^{[r]}$ is a set of orthonormal non-vanishing polynomials, none of the vector in $F^{[r]}$ can be expressed as the combination of other polynomials in $F^{[r]}$. Then the rank of the matrix with the columns listed as the polynomials from $F^{[r]}$ is $|F^{[r]}|$. Consequently, we have $|F^{[r]}| \leq T$.

Suppose we have the non-vanishing polynomial set $F_1, \cdots, F_{r-1}$, the candidate polynomial set C_r is generated as $C_r = F_{r-1} \times F_1$. Let us enumerate all the non-vanishing polynomial according to the order in which they were inserted into $F^{[r-1]}$, which is listed as $F^{[r-1]} = \{\boldsymbol{g}_1(\mathcal{S}), \cdots, \boldsymbol{g}_{|F^{[r-1]}|}(\mathcal{S})\}$. Then for any polynomial from the candidate polynomial set C_r, we have

$$\boldsymbol{g}_i(\mathcal{S}) = \underbrace{\boldsymbol{\rho}_i(\mathcal{S})}_{\mathcal{O}(T)} - \underbrace{\sum_{\boldsymbol{g}(\mathcal{S}) \in F^{[r-1]}} \langle \boldsymbol{\rho}_i(\mathcal{S}), \boldsymbol{g}(\mathcal{S}) \rangle \boldsymbol{g}(\mathcal{S})}_{\mathcal{O}(T \times |F^{[r-1]}|)}. \tag{30}$$

where $\rho_i(\mathcal{S})$ is the candidate polynomial. Since $\rho_i(\mathcal{S})$ is the constant vector, it can be evaluated in time $\mathcal{O}(T)$. There are $|F^{[r-1]}|$ vector in the non-vanishing polynomial set $F^{[r-1]}$. Any polynomial $\boldsymbol{g}_i(\mathcal{S})$ can be written as a product of two polynomials from F_1 and F_{r-1} minus a linear combination of $\boldsymbol{g}_1(\mathcal{S}), \cdots, \boldsymbol{g}_{|F^{[r-1]}|}(\mathcal{S})$. Therefore, the process can be evaluated in time $\mathcal{O}(T \times |F^{[r-1]}|)$. A similar argument shows that if we take account into all the polynomials in the candidate polynomial set C_r, the evaluation of computational cost is $\mathcal{O}(T \times |F^{[r-1]}| \times |C_r|)$. Thus, considering the iteration up to the degree of r, it will take the computational cost as $\mathcal{O}(T \sum_{i=1}^{r} (|F^{[i-1]}||C_i|)) = \mathcal{O}(T \sum_{i=1}^{r} (|F^{[i-1]}||F_1||F_{i-1}|))$.

A. COMPUTATIONAL COMPLEXITY OF TEMPORAL PROCESS

Next, we consider another part of the computational cost of the temporal structure. For the observed signal $\mathbf{x}(t) \in \mathbb{R}^n$, the calculation of the covariance $\boldsymbol{\Sigma}_{\mathbf{x}}^{\frac{1}{2}} = (\frac{1}{T}\mathbf{x}\mathbf{x}^\top)^{\frac{1}{2}}$ requires $\mathcal{O}(n^2T + n^2)$. The covariance matrix with time lag τ is defined by

$$\boldsymbol{\Sigma}_{\tau(\mathbf{x})} = \mathbb{E}[\mathbf{x}(t)\mathbf{x}(t+\tau)^\top]. \tag{31}$$

Assume we need to calculate N time-lagged correlation matrices $\boldsymbol{\Sigma}_{\tau_1(\mathbf{x})}, \cdots, \boldsymbol{\Sigma}_{\tau_N(\mathbf{x})}$, it requires $\mathcal{O}(N(n^2T + n^2))$.

Simultaneous diagonalization of N matrices is implemented by the Jacobi-like technique [37]. We are going to search a unitary matrix that makes $\mathbf{Q}\boldsymbol{\Sigma}_{\tau_1(\mathbf{x})}\mathbf{Q}^\top, \cdots, \mathbf{Q}\boldsymbol{\Sigma}_{\tau_N(\mathbf{x})}\mathbf{Q}^\top$ as a collection of diagonal matrices. Considering a set $\{\mathbf{Q}\boldsymbol{\Sigma}_{\tau_1(\mathbf{x})}\mathbf{Q}^\top, \cdots, \mathbf{Q}\boldsymbol{\Sigma}_{\tau_N(\mathbf{x})}\mathbf{Q}^\top\}$ of N matrices of size $n \times n$, the process needs to take the time $\mathcal{O}(\lambda mn^2)$, where λ is the number of iterations for the simultaneous diagonalization.

After we obtain the matrix $\mathbf{Q}$, the demixing matrix is calculated as $\mathbf{W} = (\boldsymbol{\Sigma}_{\mathbf{x}}^{\frac{1}{2}}\mathbf{Q})^{-1}$, which needs the time $\mathcal{O}(2n^3 + n^2T)$. To summarize the above process, the computational cost of the temporal process is given by

$$\mathcal{O}(n^2T + n^2 + N(n^2T + n^2) + \lambda Nn^2 + 2n^3 + n^2T). \tag{32}$$

Since we have $T \gg n$, $T \gg \lambda$ and $T \gg m$, the computation time of the temporal process can be approximated as $\mathcal{O}(Nn^2T)$. We have $|V|$ vectors in the vanishing polynomial set. Then the total computational cost can be evaluated in time $\mathcal{O}(T \sum_{i=1}^{r} (|F^{[i-1]}||F_1||F_{i-1}|) + N|V|^2T)$.

TABLE 1. A comparison of the computational complexity with several integration methods.

TDSEP [21]	KTDSEP [22], [23] [24]	ViNLisem
$\mathcal{O}(Nn^2T)$	$\mathcal{O}(Nd^2T)$	$\mathcal{O}\left(T\sum_{i=1}^{r}\|F^{[i-1]}\|\|F_1\|\|F_{i-1}\| + N\|V\|^2T\right)$

As shown in Table 1, the computational complexity of TDSEP [21] for the observed signal $\mathbf{x}(t) \in \mathbb{R}^n$ is $\mathcal{O}(Nn^2T)$, where N is the number of time lags of temporal structure. Using the approximation of multi-kernel space in [22]–[24], the cost of adding the signal channels from n to the high dimensional space with d that can be evaluated in $\mathcal{O}(Nd^2T)$. Since KTDSEP method sets the number of kernel spaces initially, the parameter d is fixed rather than depending on the data itself. In contrast, the algorithm ViNLisem is not restricted to any specific mixture or parameter model, but generate the multi-layer architecture to approximate such nonlinearity solely based on the data and the degree of vanishing polynomials.

VI. EXPERIMENTS WITH REAL-WORLD DATA

In this section, experimental results of the proposed algorithms for three kinds of nonlinear mixtures are shown. The methods used for comparison and evaluation equation are presented in Section VI-*A*. Afterward, the description of data and experimental settings are shown in Section VI-*B*. The results and their performance evaluation are given in Section VI-*C*.

A. METHODS AND EVALUATION EQUATION

The separation performance of the proposed nonlinear separation method is evaluated with other six approaches on five

real audio datasets. The following shows the six methods used for comparison.

1. TDSEP [21]: Temporal decorrelation source separation relies on the estimation of simple time-lagged covariance matrices (second-order statistics), which emphasize the difference from the temporally i.i.d. case.

2. KTDSEP[3] [24]: Kernel-based TDESP was proposed by Harmeling *et al*. that transformed the source signals into kernel spaces. The approach relies on such kernels that are assumed to be chosen enough to approximate the nonlinearity of the observed signals.

3. FICA[4] [38]: Fast independent component analysis is a significant milestone for blind source separation. It recovered the statistically dependent sources by minimizing the criterion composed of the negative-entropy.

4. KICA[5] [39]: Kernel-based ICA is used to show the necessity of exploiting nonlinear ICA methods for separating nonlinear mixtures.

5. JADE[6] [40]: Joint approximate diagonalization of eigenmatrix is considered to operate on the high-order statistics of independence.

6. SOBI[7] [36]: Second-order blind identification is a technique to exploit the coherence of the source signals, which relies only on stationary second-order statistics.

To measure the performance of recovered sources, the normalized mean squared error (NMSE) is employed [41], which has the following definition

$$\mathrm{NMSE}(\mathbf{s}_i, \hat{\mathbf{s}}_i) = 10\log_{10}\left(\frac{1}{n}\sum_{i=1}^{n}\min_{\delta}\frac{\|\mathbf{s}_i - \delta\hat{\mathbf{s}}_i\|_2^2}{\|\mathbf{s}_i\|_2^2}\right), \quad (33)$$

where $\hat{\mathbf{s}}_i$ denotes the estimate of the source signal $\mathbf{s}_i$, and δ is a scalar reflecting the scalar ambiguity.

B. DATA AND EXPERIMENT SETTING

The experiments are designed on the assumption that the observed signals are mixed nonlinearly. The sources used for the following simulations include 5 real-world audio signals with different temporal properties. They are publicly available [42]. Each one has its own advantages, depending on whether one is interested in a variety of environments, in a number of microphones, or in the overlap. For instance, the data "AMI" has two kinds of sound from the cable news and network news. Another data "Multitrack" was mixed with two anonymous singers. All the sources were sampled at 8,000 Hz. The length of the samples was varied to assess how the amount of training data affects the performance of the algorithm. The general properties of the datasets are summarized in Table 2.

Three kinds of nonlinear mixture functions were investigated, including the distorted source (DS) in [6], the post-nonlinear mixture (PNL) in [15], and the generic nonlinear (GN) in [14] and [43].

[3]http://people.kyb.tuebingen.mpg.de/harmeling/code/ktdsep-0.2.tar
[4]https://research.ics.aalto.fi/ica/fastica/
[5]http://www.di.ens.fr/ fbach/kernel-ica/index.htm
[6]http://perso.telecom-paristech.fr/ cardoso/Algo/Jade/jadeR.m
[7]https://github.com/aludnam/MATLAB/blob/master/sobi/sobi.m

TABLE 2. Descriptions of real-world data [42].

Name	Scenario	Duration(s)	Microphones	Overlap
AMI[8]	News	100	16	yes
CHiME3[9]	Talker	19	6	yes
Nonspeech[10]	Wind	20	4	no
SiSEC[11]	TV order	6	16	no
Multitrack[12]	Theater	38	20	yes

1) THE DISTORTED SOURCE (DS)

In the DS mixture function of (34), each observation is a linear mixture of nonlinear distorted sources. Specifically, in the experiments the two channel mixtures were generated according to

$$\begin{aligned} x_1(t) &= a_1 s_1(t) + 3\tanh(s_2(t)/4) + 0.1 s_2(t), \\ x_2(t) &= a_2 s_2(t) + 3\tanh(s_1(t)/4) + 0.1 s_1(t), \end{aligned} \quad (34)$$

where $a_1 = a_2 = 1$. Figure 2(a) shows the scatter plot of the sources $\mathbf{s}_i(t)$ and that of the observations $\mathbf{x}_i(t)$. To see the level of nonlinear distortion in the mixing transformation, we give the scatter plot of the affine transformation of $\mathbf{s}_i(t)$ in Figure 2(b).

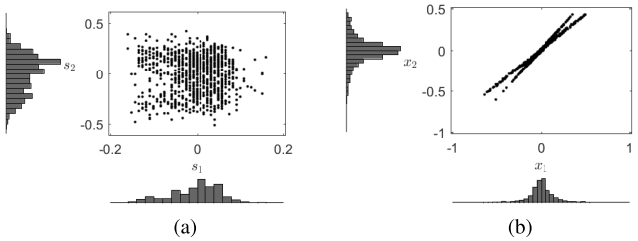

FIGURE 2. (a) The scatter plots of the original sources use the "AMI" dataset[8] in Table 2. (b) The mixture signals are generated from distorted source (DS) function. (a) Source signals. (b) Mixture signals.

2) THE POST-NONLINEAR (PNL)

The post-nonlinear mixtures constitute a particularly interesting example of the theoretical separability characterized by weak indeterminacy. The sources were the first subject to a linear mixture $\mathbf{z}(t) = \mathbf{A}\mathbf{s}(t)$, where $\mathbf{A}$ is a 2×2 mixing matrix give by

$$\mathbf{A} = \begin{pmatrix} -0.2261 & -0.1189 \\ -0.1706 & -0.2836 \end{pmatrix}. \quad (35)$$

Then each mixture component is generated from a nonlinear, invertible transformation, as the form of

$$\begin{aligned} x_1(t) &= (z_2(t) + 3z_1(t) + 6)\cos(1.5\pi)z_1(t), \\ x_2(t) &= (z_2(t) + 3z_1(t) + 6)\sin(1.5\pi)z_1(t). \end{aligned} \quad (36)$$

The sources are plotted in Figure 3(a). The mixture components are shown in Figure 3(b), where we can see the distortions caused by the nonlinearities.

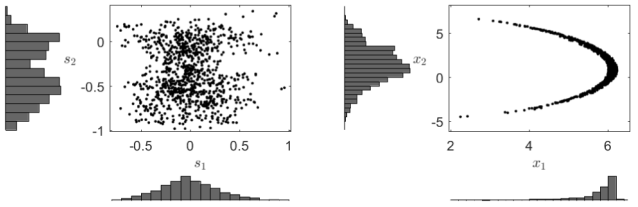

FIGURE 3. (a) The scatter plots of the original sources use the "ChiME3" dataset[9] in Table 2. (b) The mixture signals are generated from post-nonlinear (PNL) function. (a) Source signals. (b) Mixture signals.

3) THE GENERIC NONLINEAR (GN)

In the following example, at each sample t, the sources are mixed nonlinearly as

$$\begin{bmatrix} x_1(t) \\ x_2(t) \end{bmatrix} = \begin{bmatrix} \cos\alpha(s(t)) & -\sin\alpha(s(t)) \\ \sin\alpha(s(t)) & \cos\alpha(s(t)) \end{bmatrix} \begin{bmatrix} s_1(t) \\ s_2(t) \end{bmatrix}, \quad (37)$$

where $\alpha(s(t))$ is defined by the parameter model

$$\alpha(s(t)) = \alpha_0 + \gamma \times \sqrt{s_1^2(t) + s_2^2(t)}.$$

In our simulation, the parameter α_0 and γ are set to 0 and 1, respectively.

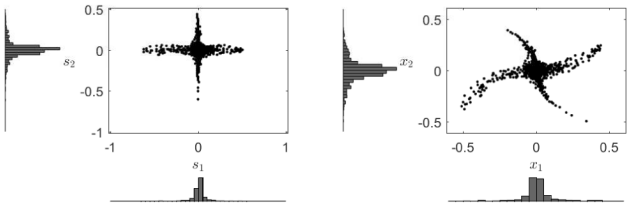

FIGURE 4. (a) The scatter plots of the original sources use the "Nonspeech" dataset[10] in Table 2. (b) The mixture signals are generated from generic nonlinear (GN) function. (a) Source signals. (b) Mixture signals.

Figure 4(a) illustrates the source signals, which is the case for the audio data of "Nonspeech" collected in Table 2. By using a mixing function given in (37), the observations are nonlinearly mixed, which is shown as an anchor-shaped structure in Figure 4(b). The mixing function (37) is not symmetric in $\mathbf{s}_1(t)$ and $\mathbf{s}_2(t)$. Thus, for every pair of sources, there are two possible mixtures and we have tested both for each source pair.

For most blind source separation method based on the temporal structure, such as TDSEP, KTDSEP and our proposed ViNLisem method, the selection of the optimal time lags is a tough problem. Clearly, the performance can be degraded if the improper delay is chosen, whereas a large number of delays always give a stable solution. Here, we got some knowledge of practical experiments, which was shown in Figure 5 that many delays always brings us to the stable side. Thus, in the following experiments, the time-shift is

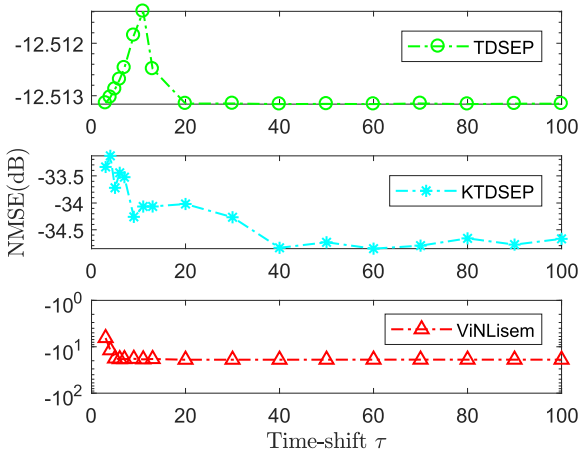

FIGURE 5. The performance indexes consider various time shift τ for the methods with temporal structure, such as TDSEP, KTDSEP, and our proposed ViNLisem.

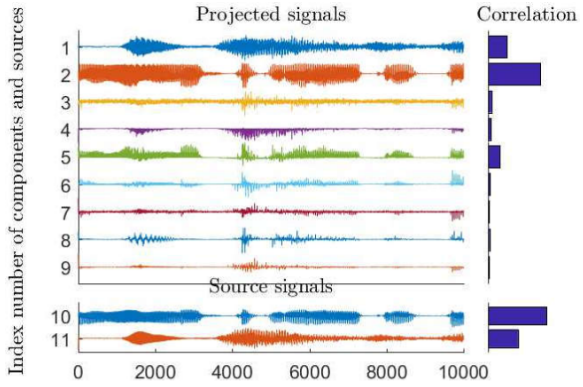

FIGURE 6. All the projected components and the original sources. The horizontal bars indicate the normalized correlation.

set as $\tau_{\text{TDSEP}} = 0, \cdots, 20$, $\tau_{\text{KTDSEP}} = 0, \cdots, 40$ and $\tau_{\text{ViNLisem}} = 0, \cdots, 7$, respectively.

In addition, for the best parameter setting, we could apply KTDSEP with a polynomial kernel of degree 9, i.e. $\mathcal{K}(\mathbf{s}_1, \mathbf{s}_2) = (\mathbf{s}_1^\top \mathbf{s}_2 + 1)^9$ and the dimensionality of kernel space set as 20. In practice, the real data are noisy that allow us to consider a tolerate value ϵ, so as to the polynomials almost vanish, i.e. $\|g_i(\mathbf{x})\| \leq \epsilon$. The parameter ϵ is used to indicate the distance between the measured polynomials and the value 0. If a bigger ϵ is selected, the polynomials will have a bigger distance from the value 0. However, if a smaller ϵ is selected, the degree of the polynomial will be higher to make the polynomial satisfying the restrict of ϵ. Then the cost time will be longer to search such polynomial. Therefore, we set the parameter $\epsilon = 0.001$ according to the experiments of the real datasets. The additive noise is generated to be white and Gaussian with uncorrelated samples whose variance was assumed to be uniform. The algorithms are performed under the signal-to-noise power ratio (SNR) varied from 5 dB to 45 dB by a step of 10 dB. To reduce the randomness effect, 20 times of Monte Carlo simulations are performed to evaluate the performance of the algorithms versus different SNR.

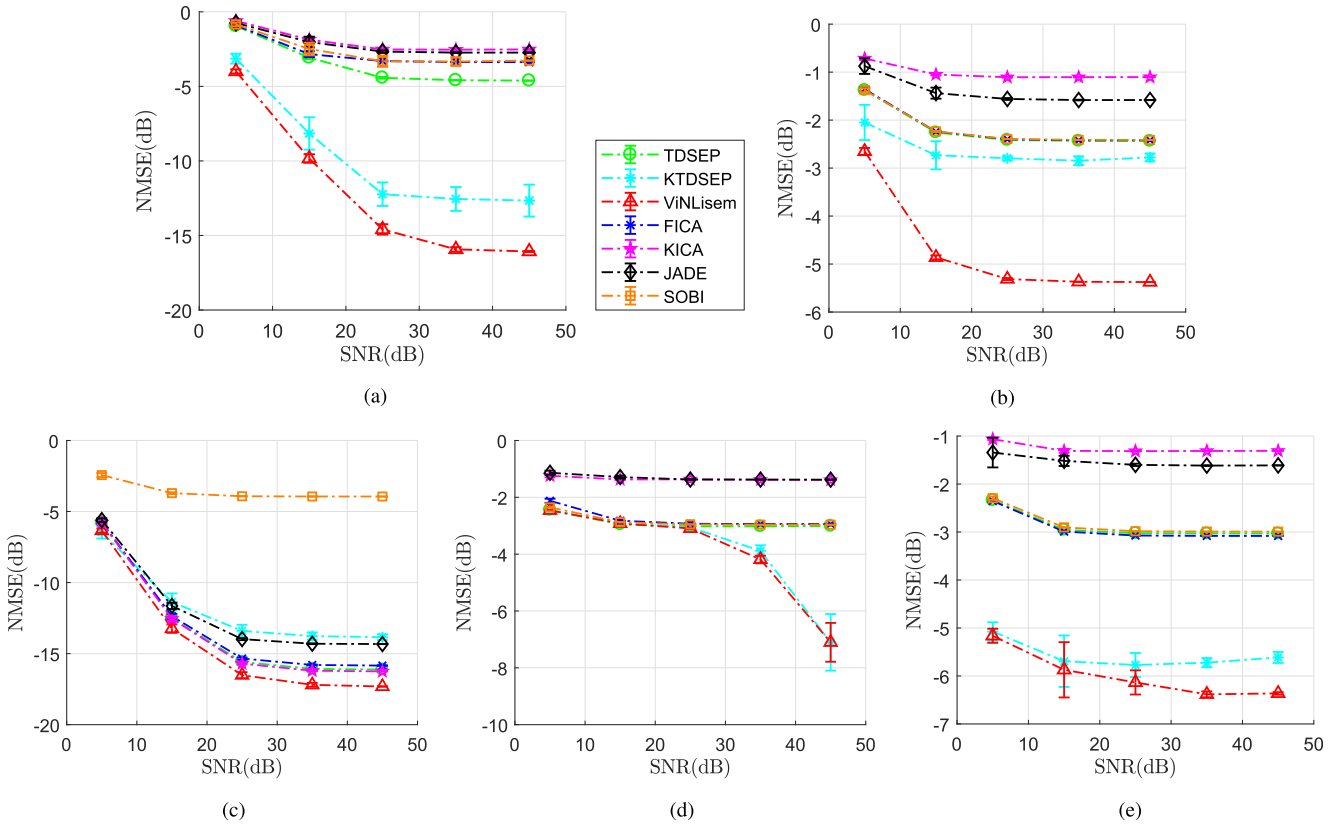

FIGURE 7. The separation performance comparison for three kinds of mixed functions in which the different dataset in Table 2 are used. (a) The accuracy for DS mixture on Data 1. (b) The accuracy for PNL mixture on Data 2. (c) The accuracy for GN mixture on Data 3. (d) The accuracy for GN mixture on Data 4. (e) The accuracy for GN mixture on Data 5.

C. RESULTS

Since our algorithm utilizes a set of polynomials to approximate the nonlinearity of mixture, we thus obtain 9 components (projected signals) adaptively for dataset "AMI" as shown in Figure 6. Then, two components with the maximum correlation are selected as described in the previous section. The best matching waveforms with the maximum correlations are shown as the first and second rows, which are denoted as the estimation of original signals $\hat{\mathbf{s}}_1$ and $\hat{\mathbf{s}}_2$, respectively. The algorithm automatically chooses two signals that turn out to reach very high correlation coefficients (cc), such as $cc(\mathbf{s}_1, \hat{\mathbf{s}}_1) = 0.9848$ and $cc(\mathbf{s}_2, \hat{\mathbf{s}}_2) = 0.9803$.

To clarify the separation performance, we use the NMSE in (33) as the error measure. We evaluate seven BSS approaches on three kinds of mixed functions with five different datasets. Figure 7 show parts of the experimental results. Similar accuracy trends were also observed with other datasets being used to testify different mixed functions with different BSS approaches. We can see from Figure 7 that the ViNLisem achieved a more accurate estimate than the other methods. In contrast, FICA and KICA optimized their estimate by having access to all the samples in one space. In addition, we also verified that for all datasets, the improved performance of the proposed approach was significant. Apart from the estimation quality, an important aspect for ViNLisem method is that the vanishing components are constructed solely on the input data without any additional constraints on the mixing functions except for invertibility.

Among these methods used for comparison, we can distinguish two classes. Methods such as JADE and Fast ICA are based on statistics of order higher than two, which require at most one source can be Gaussian. This means that their performance will be poor if more than one source is close to Gaussian. However, in practice, most of the sources have distributions deviate markedly from Gaussian (e.g. speech data are strongly super-Gaussian, while images tend to be strongly sub-Gaussian). Methods of this class do not exploit any temporal or spatial structure of the sources. On the other hand, methods such as TDSEP and SOBI use only second-order statistics, and can deal with any number of Gaussian sources. However, they require sources being with temporal structure. Again, most sources of practical interest (such as speech, biomedical signals or images) do not have a temporal or spatial structure that can be used.

Note that unlike KTDSEP, ViNLisem does not assume the number of approximate functions initially, but adapt to the nonlinear approximation in the form of a multi-layer representation. Therefore, the complexity and storage requirements of the model are proportional to the number of vanishing components. The complexity of the models learned by ViNLisem is generally larger than that of the KTDSEP.

VII. CONCLUSION

Our work has three main contributions. First, the approach presents a novel mathematical construction with a multi-layer architecture. By using the layer-by-layer representation, we can approximate such nonlinearity of mixing functions. Similar to the principle of modern deep learning, the layers are generated one-by-one up to the higher-degree representations of data. Once such representations are generated, a final output layer is constructed by solving a convex optimization problem. Thus, the technique establishes a highly useful isomorphism between the projection of the data points and the multi-layer representations. By projecting a time-invariant nonlinear BSS to the local linear problem, the nonlinear problem can be linearly separable. Importantly, the parameters and forms of polynomials depend solely on the input data, which guarantees the robustness of the structures. We thus address the general problem without being restricted to any specific mixture or parametric model.

Then, the layer-by-layer representation is adaptively generated solely on the observations. As the number of spanned spaces goes up, the computational complexity grows exponentially. To overcome this obstacle, relying on the properties of vanishing polynomials, we provide a feasible way to reduce the computational cost as shown in Theorem 2 and Theorem 3. Finally, considering the temporal correlation as the separation criterion, the approach can be designed by emphasizing the difference from the temporally i.i.d. data. Therefore, we can break the nonlinear problem down into a simpler version of the generalized joint diagonalization problem in the feature space. However, due to adopting the nonlinear approximation in the form of a sample representation, the complexity and storage requirements of the model are proportional to the number of vanishing components, which is generally larger than that of the TDSEP and KTDSEP.

Appendix A
PROOF OF THEOREM 2

For instance, considering the polynomials of degree 2, we set $\rho_{i_1,i_2}(\mathbf{x}(t)) = x_{i_1}(t)x_{i_2}(t)$, for all i_1 and i_2. Thus, we now need to consider n^2+n+1 columns. As the degree goes up, the number of columns increases exponentially. To overcome this obstacle, we propose a method to reduce the computational cost relying on the underlying structure and the property of the vanishing ideal

Proof: Denoting $F_1 = \{\mathbf{p}_1^{(1)}, \cdots, \mathbf{p}_{|F_1|}^{(1)}\}$ as a non-vanishing polynomial set of degree 1, where $|F_1|$ denotes the number of elements included in the set F_1. Any polynomial of degree 1 generated from F_1 can be expressed as

$$\boldsymbol{f}_{i_1}^{(1)} = \sum_{j_1} \alpha_{i_1,j_1}^{(1)} \mathbf{p}_{j_1}^{(1)}, \quad \boldsymbol{h}_{i_2}^{(1)} = \sum_{j_2} \alpha_{i_2,j_2}^{(1)} \mathbf{p}_{j_2}^{(1)}, \tag{A.1}$$

where $\alpha_{i_1,j_1}^{(1)}$ and $\alpha_{i_2,j_2}^{(1)}$ denote the coefficients that make $\boldsymbol{f}_{i_1}^{(1)} \circ \boldsymbol{h}_{i_2}^{(1)} \neq \mathbf{0}_{T\times 1}$ for all $i_1, i_2 \leq l$. Then F_2 can be generated from the span of $\boldsymbol{f}_{i_1}^{(1)}$ and $\boldsymbol{h}_{i_2}^{(1)}$ for $i_1, i_2 \leq l$ as

$$\begin{aligned}\hat{\boldsymbol{g}}^{(2)}(\mathcal{S}) &= \sum_{i_1,i_2\leq l} \boldsymbol{f}_{i_1}^{(1)} \circ \boldsymbol{h}_{i_2}^{(1)} \\ &= \sum_{i_1,i_2\leq l} \Big(\sum_{j_1} \alpha_{i_1,j_1}^{(1)} \mathbf{p}_{j_1}^{(1)}\Big)\Big(\sum_{j_2} \alpha_{i_2,j_2}^{(1)} \mathbf{p}_{j_2}^{(1)}\Big) \\ &= \sum_{j_1,j_2} \Big[(\mathbf{p}_{j_1}^{(1)} \circ \mathbf{p}_{j_2}^{(1)})\Big(\sum_{i_1,i_2\leq l} \alpha_{i_1,j_1}^{(1)} \alpha_{i_2,j_2}^{(1)}\Big)\Big], \end{aligned} \tag{A.2}$$

where the polynomials are assumed to be composed of linear functions that each linear function is described by a coefficient vector $\boldsymbol{\alpha} \in \mathbb{R}^{n+1}$. And $\alpha_{i_1,j_1}^{(1)}$ is the coefficient that corresponds to the i_1-th element of the candidate set C_1, which is used to weight the j_1-th element $\mathbf{p}_{j_1}^{(1)}$ of the non-vanishing polynomial set F_1. Thus, we have $\hat{\boldsymbol{g}}^{(2)}(\mathcal{S})$ generated from the span of $F_1 \times F_1$ and that can be used to construct F_2 and V_2. □

Appendix B
PROOF OF THEOREM 3

A. CONSTRUCTING THE POLYNOMIALS OF DEGREE 3

Considering the polynomials of degree 3, we set $\rho_{i_1,i_2,i_3}(\mathbf{x}(t)) = x_{i_1}(t)x_{i_2}(t)x_{i_3}(t)$, for all i_1, i_2 and i_3. Then $\hat{\boldsymbol{g}}^{(3)}(\mathcal{S})$ is generated from the span of $F_1 \times F_2$.

Proof: Denoting $F_2 = \{\mathbf{p}_1^{(2)}, \cdots, \mathbf{p}_{|F_2|}^{(2)}\}$ as a non-vanishing polynomial set of degree 2, where $|F_2|$ denotes the number of elements included in the set F_2. Similarly, any polynomial of degree 3 can be expressed as

$$\boldsymbol{g}^{(3)}(\mathcal{S}) = \sum_{i_1,i_2,i_3} \alpha_{i_1,i_2,i_3} \boldsymbol{\rho}_{i_1,i_2,i_3}(\mathcal{S}). \tag{B.1}$$

The polynomial $\hat{\boldsymbol{g}}^{(3)}(\mathcal{S}) = \sum_{i_1,i_2,i_3\leq l} \boldsymbol{\rho}_{i_1,i_2,i_3}$ satisfies $\hat{\boldsymbol{g}}^{(3)}(\mathcal{S}) = \boldsymbol{g}^{(3)}(\mathcal{S})$ for $i_1, i_2, i_3 \leq l$ for assumption. Then, $\boldsymbol{g}^{(3)}(\mathcal{S})$ can be approximated as

$$\begin{aligned}&\hat{\boldsymbol{g}}^{(3)}(\mathcal{S}) \\ &= \sum_{i_1,i_2,i_3} \alpha_{i_1,i_2,i_3} \boldsymbol{\rho}_{i_1,i_2,i_3} \\ &= \sum_{i_1,i_2,i_3} \Big(\sum_{j_1} \alpha_{i_1,j_1}^{(1)} \mathbf{p}_{j_1}^{(1)}\Big)\Big(\sum_{j_2} \alpha_{i_2,j_2}^{(1)} \mathbf{p}_{j_2}^{(1)}\Big)\Big(\sum_{j_3} \alpha_{i_3,j_3}^{(1)} \mathbf{p}_{j_3}^{(1)}\Big) \\ &= \sum_{i_1,i_2,i_3} \Big(\sum_{j_1,j_2} (\alpha_{i_1,j_1}^{(1)} \alpha_{i_2,j_2}^{(1)}) \mathbf{p}_{j_1}^{(1)} \mathbf{p}_{j_2}^{(1)}\Big)\Big(\sum_{j_3} \alpha_{i_3,j_3}^{(1)} \mathbf{p}_{j_3}^{(1)}\Big). \end{aligned} \tag{B.2}$$

Since $\sum_{j_1,j_2} (\alpha_{i_1,j_1}^{(1)} \alpha_{i_2,j_2}^{(1)}) \mathbf{p}_{j_1}^{(1)} \mathbf{p}_{j_2}^{(1)}$ is in the span of $F_1 \times F_1$, thus it can be expressed as

$$\sum_{j_1,j_2} (\alpha_{i_1,j_1}^{(1)} \alpha_{i_2,j_2}^{(1)}) \mathbf{p}_{j_1}^{(1)} \mathbf{p}_{j_2}^{(1)} = \sum_{j} \alpha_j^{(2)} \mathbf{p}_j^{(2)}. \tag{B.3}$$

Then (B.2) can be written as

$$\begin{aligned}&\hat{\boldsymbol{g}}^{(3)}(\mathcal{S}) \\ &= \sum_{i_1,i_2,i_3\leq l} \Big(\sum_{j_1,j_2} (\alpha_{i_1,j_1}^{(1)} \alpha_{i_2,j_2}^{(1)}) \mathbf{p}_{j_1}^{(1)} \mathbf{p}_{j_2}^{(1)}\Big)\Big(\sum_{j_3} \alpha_{i_3,j_3}^{(1)} \mathbf{p}_{j_3}^{(1)}\Big)\end{aligned}$$

$$\begin{aligned}\hat{\boldsymbol{g}}^{(t)}(\mathcal{S}) &= \sum_{i_1,i_2,\cdots,i_t\le l} \boldsymbol{\rho}_{i_1,i_2,\cdots,i_t} \\ &= \sum_{i_1,i_2,\cdots,i_t\le l} \left(\sum_{j_1} \alpha^{(1)}_{i_1,j_1}\mathbf{p}^{(1)}_{j_1}\right)\left(\sum_{j_2} \alpha^{(1)}_{i_2,j_2}\mathbf{p}^{(1)}_{j_2}\right)\cdots\left(\sum_{j_t} \alpha^{(1)}_{i_t,j_t}\mathbf{p}^{(1)}_{j_t}\right) \\ &= \sum_{i_1,i_2,\cdots,i_t\le l} \left(\sum_{j_1,j_2,\cdots,j_{t-1}} (\alpha^{(1)}_{i_1,j_1}\alpha^{(1)}_{i_2,j_2}\cdots\alpha^{(1)}_{i_{t-1},j_{t-1}})\mathbf{p}^{(1)}_{j_1}\mathbf{p}^{(1)}_{j_2}\cdots\mathbf{p}^{(1)}_{j_{t-1}}\right)\left(\sum_{j_t} \alpha^{(1)}_{i_t,j_t}\mathbf{p}^{(1)}_{j_t}\right) \end{aligned} \tag{B.6}$$

Since $\sum_{j_1,j_2,\cdots,j_{t-1}} (\alpha^{(1)}_{i_1,j_1}\alpha^{(1)}_{i_2,j_2}\cdots\alpha^{(1)}_{i_{t-1},j_{t-1}})\mathbf{p}^{(1)}_{j_1}\mathbf{p}^{(1)}_{j_2}\cdots\mathbf{p}^{(1)}_{j_{t-1}}$ is in the span of $F_{t-2}\times F_1$, thus it can be expressed as

$$\sum_{j_1,j_2,\cdots,j_{t-1}} (\alpha^{(1)}_{i_1,j_1}\alpha^{(1)}_{i_2,j_2}\cdots\alpha^{(1)}_{i_{t-1},j_{t-1}})\mathbf{p}^{(1)}_{j_1}\mathbf{p}^{(1)}_{j_2}\cdots\mathbf{p}^{(1)}_{j_{t-1}} = \sum_j \alpha^{(t-1)}_j \mathbf{p}^{(t-1)}_j. \tag{B.7}$$

Then (B.6) can be rewritten as

$$\begin{aligned}\hat{\boldsymbol{g}}^{(t)}(\mathcal{S}) &= \sum_{i_1,i_2,\cdots,i_t\le l} \left(\sum_{j_1,j_2,\cdots,j_{t-1}} (\alpha^{(1)}_{i_1,j_1}\alpha^{(1)}_{i_2,j_2}\cdots\alpha^{(1)}_{i_{t-1},j_{t-1}})\mathbf{p}^{(1)}_{j_1}\mathbf{p}^{(1)}_{j_2}\cdots\mathbf{p}^{(1)}_{j_{t-1}}\right)\left(\sum_{j_t} \alpha^{(1)}_{i_t,j_t}\mathbf{p}^{(1)}_{j_t}\right) \\ &= \sum_{i_t\le l}\left(\sum_j \alpha^{(t-1)}_j \mathbf{p}^{(t-1)}_j\right)\left(\sum_{j_t} \alpha^{(1)}_{i_t,j_t}\mathbf{p}^{(1)}_{j_t}\right) \\ &= \sum_{j,j_t} \mathbf{p}^{(t-1)}_j \mathbf{p}^{(1)}_{j_t}\left(\sum_{i_t\le l} \alpha^{(t-1)}_j \alpha^{(1)}_{i_t,j_t}\right), \end{aligned} \tag{B.8}$$

where the polynomials are assumed to be composed of linear functions that each linear function is described by a coefficient vector $\boldsymbol{\alpha}\in\mathbb{R}^{n+1}$, and $\alpha^{(1)}_{i_t,j_t}$ is the coefficient that corresponds to the i_t-th element of the candidate set C_1, and that is used to weight the j_1-th element $\mathbf{p}^{(1)}_{j_1}$ of the non-vanishing polynomial set F_1. Therefore, we can generate $\hat{\boldsymbol{g}}^{(t)}(\mathcal{S})$ only in the span of $F_{t-1}\times F_1$ rather than considering all the extension space. □

$$\begin{aligned}&= \sum_{i_3\le l}\left(\sum_j \alpha^{(2)}_j \mathbf{p}^{(2)}_j\right)\left(\sum_{j_3} \alpha^{(1)}_{i_3,j_3}\mathbf{p}^{(1)}_{j_3}\right) \\ &= \sum_{j,j_3} \mathbf{p}^{(2)}_j \mathbf{p}^{(1)}_{j_3}\left(\sum_{i_3\le l} \alpha^{(2)}_j \alpha^{(1)}_{i_3,j_3}\right). \end{aligned} \tag{B.4}$$

Thus, $\hat{\boldsymbol{g}}^{(3)}(\mathcal{S})$ is generated from the span of $F_2\times F_1$ that can be used to construct F_3 and V_3.

B. CONSTRUCTING THE POLYNOMIALS OF HIGHER DEGREE

Similar to the above processing procedure, any polynomial of degree t can be expressed as

$$\boldsymbol{g}^{(t)}(\mathcal{S}) = \sum_{i_1,i_2,\cdots,i_t} \alpha_{i_1,i_2,\cdots,i_t}\boldsymbol{\rho}_{i_1,i_2,\cdots,i_t}(\mathcal{S}). \tag{B.5}$$

The polynomial $\boldsymbol{g}^{(t)}(\mathcal{S}) = \sum_{i_1,i_2,\cdots,i_t} \boldsymbol{\rho}_{i_1,i_2,\cdots,i_t}$ satisfies $\hat{\boldsymbol{g}}^{(t)}(\mathcal{S}) = \boldsymbol{g}^{(t)}(\mathcal{S})$ for $i_1, i_2, \cdots, i_t \le l$. Denoting $F_{t-1} = \{\mathbf{p}^{(t-1)}_1, \cdots, \mathbf{p}^{(t-1)}_{|F_{t-1}|}\}$, $\hat{\boldsymbol{g}}^{(t)}(\mathcal{S})$ can be written as (B.6). □

REFERENCES

[1] S. Gannot, E. Vincent, S. Markovich-Golan, and A. Ozerov, "A consolidated perspective on multimicrophone speech enhancement and source separation," *IEEE/ACM Trans. Audio, Speech, Language Process.*, vol. 25, no. 4, pp. 692–730, Apr. 2017.

[2] S.-I. Amari, S. C. Douglas, A. Cichocki, and H. H. Yang, "Novel online adaptive learning algorithms for blind deconvolution using the natural gradient approach," *IFAC Proc. Vol.*, vol. 30, no. 11, pp. 1007–1012, 1997.

[3] W. A. Gardner, "A new method of channel identification," *IEEE Trans. Commun.*, vol. 39, no. 6, pp. 813–817, Jun. 1991.

[4] A. Hyvärinen and H. Morioka, "Nonlinear ICA of temporally dependent stationary sources," in *Proc. Int. Conf. Artif. Intell. Statist. (AISTATS)*, vol. 54, 2017, pp. 460–469.

[5] A. Hyvärinen and P. Pajunen, "Nonlinear independent component analysis: Existence and uniqueness results," *Neural Netw.*, vol. 12, no. 3, pp. 429–439, 1999.

[6] K. Zhang and L. Chan, "Minimal nonlinear distortion principle for nonlinear independent component analysis," *J. Mach. Learn. Res.*, vol. 9, pp. 2455–2487, Nov. 2008.

[7] G. Deco and W. Brauer, "Nonlinear higher-order statistical decorrelation by volume-conserving neural architectures," *Neural Netw.*, vol. 8, no. 4, pp. 525–535, 1995.

[8] L. Dinh, D. Krueger, and Y. Bengio. (Apr. 2015). "NICE: Nonlinear independent components estimation." [Online]. Available: https://arxiv.org/abs/1410.8516

[9] Y. Wu, T. K. Doyle, and C. Fyfe, "Multi-layer topology preserving mapping for k-means clustering," in *Proc. Int. Conf. Intell. Data Eng. Automated Learn.*, 2011, pp. 84–91.

[10] B. Qi, V. John, Z. Liu, and S. Mita, "Pedestrian detection from thermal images: A sparse representation based approach," *Infr. Phys. Technol.*, vol. 76, pp. 157–167, May 2016.

[11] J. B. Tenenbaum, V. de Silva, and J. C. Langford, "A global geometric framework for nonlinear dimensionality reduction," *Science*, vol. 290, no. 5500, pp. 2319–2322, Dec. 2000.

[12] H. H. Yang, S.-I. Amari, and A. Cichocki, "Information–theoretic approach to blind separation of sources in non-linear mixture," *Signal Process.*, vol. 64, no. 3, pp. 291–300, 1998.

[13] G. Marques and L. Almeida, "Separation of nonlinear mixtures using pattern repulsion," in *Proc. Int. Workshop Independent Compon. Anal. Signal Separat.*, 1999, pp. 277–282.

[14] B. Ehsandoust, M. Babaie-Zadeh, B. Rivet, C. Jutten, "Blind source separation in nonlinear mixtures: Separability and a basic algorithm," *IEEE Trans. Signal Process.*, vol. 65, no. 16, pp. 4339–4352, Aug. 2017.

[15] P. Comon and C. Jutten, *Handbook of Blind Source Separation: Independent Component Analysis and Applications*, 1st ed. New York, NY, USA: Academic, Feb. 2010.
[16] R. Livni, D. Lehavi, S. Schein, H. Nachliely, S. Shalev-Shwartz, and A. Globerson, "Vanishing component analysis," in *Proc. Int. Conf. Mach. Learn. (ICML)*, 2013, pp. 597–605.
[17] A. A. Nugraha, A. Liutkus, and E. Vincent, "Multichannel audio source separation with deep neural networks," *IEEE/ACM Trans. Audio, Speech, Language Process.*, vol. 24, no. 9, pp. 1652–1664, Sep. 2016.
[18] S. Uhlich *et al.*, "Improving music source separation based on deep neural networks through data augmentation and network blending," in *Proc. Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, Mar. 2017, pp. 261–265.
[19] E. M. Grais, G. Roma, A. J. R. Simpson, and M. D. Plumbley, "Discriminative enhancement for single channel audio source separation using deep neural networks," in *Proc. Int. Conf. Latent Variable Anal. Signal Separat.*, 2017, pp. 236–246.
[20] R. Livni, S. Shalev-Shwartz, and O. Shamir. (2013). "An algorithm for training polynomial networks." [Online]. Available: https://arxiv.org/abs/1304.7045
[21] A. Ziehe and K.-R. Müller, "TDSEP—An efficient algorithm for blind separation using time structure," in *Proc. Int. Conf. Artif. Neural Netw. (ICANN)*, 1998, pp. 675–680.
[22] S. Harmeling, A. Ziehe, M. Kawanabe, B. Blankertz, and K.-R. Müller, "Nonlinear blind source separation using kernel feature spaces," in *Proc. Int. Workshop Independ. Compon. Anal. Blind Signal Separat.*, 2001, pp. 102–107.
[23] S. Harmeling, A. Ziehe, M. Kawanabe, and K.-R. Müller, "Kernel feature spaces and nonlinear blind souce separation," in *Proc. Adv. Neural Inf. Process. Syst.*, 2002, pp. 761–768.
[24] S. Harmeling, A. Ziehe, M. Kawanabe, and K. Müller, "Kernel-based nonlinear blind source separation," *Neural Comput.*, vol. 15, no. 5, pp. 1089–1124, May 2003.
[25] H. Sprekeler, T. Zito, and L. Wiskott, "An extension of slow feature analysis for nonlinear blind source separation," *J. Mach. Learn. Res.*, vol. 15, pp. 921–947, Mar. 2014.
[26] Y. Bengio, A. Courville, and P. Vincent, "Representation learning: A review and new perspectives," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 35, no. 8, pp. 1798–1828, Aug. 2013.
[27] D. Heldt, M. Kreuzer, S. Pokutta, and H. Poulisse, "Approximate computation of zero-dimensional polynomial ideals," *J. Symbolic Comput.*, vol. 44, no. 11, pp. 1566–1591, 2009.
[28] R. Livni, S. Shalev-Shwartz, and O. Shamir, "On the computational efficiency of training neural networks," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 27, 2014, pp. 855–863.
[29] M. Donini and F. Aiolli, "Learning deep kernels in the space of dot product polynomials," *Mach. Learn.*, vol. 106, nos. 9–10, pp. 1245–1269, 2017.
[30] C. K. I. Williams and M. Seeger, "Using the Nyström method to speed up kernel machines," in *Proc. Int. Conf. Neural Inf. Process. Syst.*, 2001, pp. 661–667.
[31] P. Kar and H. Karnick, "Random feature maps for dot product kernels," in *Proc. Int. Conf. Artif. Intell. Statist.*, vol. 22, 2012, pp. 583–591.
[32] N. Pham and R. Pagh, "Fast and scalable polynomial kernels via explicit feature maps," in *Proc. Int. Conf. Knowl. Discovery Data Mining*, 2013, pp. 239–247.
[33] H. Avron, H. Nguyen, and D. Woodruff, "Subspace embeddings for the polynomial kernel," in *Proc. Adv. Neural Inf. Process. Syst.*, 2014, pp. 2258–2266.
[34] M. Blondel, M. Ishihata, A. Fujino, and N. Ueda, "Polynomial networks and factorization machines: New insights and efficient training algorithms," in *Proc. Int. Conf. Mach. Learn.*, vol. 48, 2016, pp. 850–858.
[35] D. Cox, J. Little, and D. O'Shea, *Ideals, Varieties, and Algorithms: An Introduction to Computational Algebraic Geometry and Commutative Algebra*, vol. 10, 3rd ed. New York, NY, USA: Springer-Verlag, 2007.
[36] A. Belouchrani, K. Abed-Meraim, J.-F. Cardoso, and E. Moulines, "A blind source separation technique using second-order statistics," *IEEE Trans. Signal Process.*, vol. 45, no. 2, pp. 434–444, Feb. 1997.
[37] J.-F. Cardoso and A. Souloumiac, "Jacobi angles for simultaneous diagonalization," *SIAM J. Matrix Anal. Appl.*, vol. 17, no. 1, pp. 161–164, Jan. 1996.
[38] A. Hyvärinen, J. Karhunen, and E. Oja, *Independent Component Analysis*. Hoboken, NJ, USA: Wiley, 2001.
[39] F. R. Bach and M. I. Jordan, "Kernel independent component analysis," *J. Mach. Learn. Res.*, vol. 3, pp. 1–48, Jan. 2002.
[40] J.-F. Cardoso, "High-order contrasts for independent component analysis," *Neural Comput.*, vol. 11, no. 1, pp. 157–192, 1999.
[41] L. Zhen, D. Peng, Z. Yi, Y. Xiang, and P. Chen, "Underdetermined blind source separation using sparse coding," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 28, no. 12, pp. 3102–3108, Dec. 2017.
[42] J. Roux and E. Vincent, "A categorization of robust speech processing datasets," Mitsubishi Electr. Res. Labs, Cambridge, MA, USA, Tech. Rep. TR2014–116, Aug. 2014.
[43] M. Babaie-Zadeh, "On blind source separation in convolutive and nonlinear mixtures," Ph.D. dissertation, Dept. Elect. Eng., Sharif Univ. Technol., Tehran, Iran, Sep. 2002. [Online]. Available: http://sharif.edu/~mbzadeh/Publications/PublicationFiles/PhD_Thesis/2002/MBzadehPhDthesisEnglish.pdf

LU WANG received the B.E. and M.E. degrees from the Faculty of Electrical Engineering, Heilongjiang University, Harbin, China, in 2008 and 2011, respectively. She is currently pursuing the Ph.D. degree with the Graduate School of Science and Technology, Keio University, Yokohama, Japan. Her research interests include blind source separation, model mining, and knowledge discovery from massive data with a recent emphasis on the improvement of noisy speech and biomedical engineering applications.

TOMOAKI OHTSUKI (SM'01) received the B.E., M.E., and Ph.D. degrees in electrical engineering from Keio University, Yokohama, Japan, in 1990, 1992, and 1994, respectively. From 1994 to 1995, he was a Post-Doctoral Fellow and a Visiting Researcher in electrical engineering with Keio University. From 1993 to 1995, he was a Special Researcher of Fellowships with the Japan Society for the promotion of science for Japanese junior scientists. From 1995 to 2005, he was with the Science University of Tokyo. From 1998 to 1999, he was with the Department of Electrical Engineering and Computer Sciences, University of California at Berkeley. In 2005, he joined Keio University, where he is currently a Professor. He is also involved in research on wireless communications, optical communications, signal processing, and information theory. He was a recipient of the 1997 Inoue Research Award for Young Scientist, the 1997 Hiroshi Ando Memorial Young Engineering Award, the Ericsson Young Scientist Award 2000, the 2002 Funai Information and Science Award for Young Scientist, the IEEE the 1st Asia–Pacific Young Researcher Award 2001, the 5th International Communication Foundation Research Award, the 2011 IEEE SPCE Outstanding Service Award, the 27th TELECOM System Technology Award, the ETRI Journal's 2012 Best Reviewer Award, and the 9th International Conference on Communications and Networking (CHINACOM) Best Paper Award in China, in 2014.

He has published over 160 journal papers and 370 international conference papers. He was the Vice President of the Communications Society of the IEICE. He is a fellow of the IEICE. He has served the General-Co Chair and Symposium Co-Chair of many conferences, including IEEE GLOBECOM 2008, SPC, IEEE ICC2011, CTS, IEEE GCOM2012, and IEEE SPAWC. He also served as the Chair of the IEEE Communications Society and the Signal Processing for Communications and Electronics Technical Committee. He served as a Technical Editor for the *IEEE Wireless Communications Magazine* and an Editor of *Physical Communications* (Elsevier). He is currently serving as an Area Editor for the IEEE TRANSACTIONS ON *Vehicular* Technology and an Editor of the IEEE COMMUNICATIONS SURVEYS AND TUTORIALS. He gave tutorials and keynote speech at many international conferences including IEEE VTC and IEEE PIMRC.