Tensor Decomposition Meets Blind Source Separation

Thanh Trung Le[a,b,*], Karim Abed-Meraim[b,d], Philippe Ravier[b], Olivier Buttelli[b], Ales Holobar[c]

[a] *VNU University of Engineering and Technology, Vietnam*
[b] *University of Orléans, PRISME Laboratory, France*
[c] *University of Maribor, FERI Faculty, Slovenia*
[d] *Institut Universitaire de France, France*

Abstract

In this paper, we investigate the problem of blind source separation (BSS) through the lens of tensor decomposition (TD). Two fundamental connections between TD and BSS are established, forming the basis for two novel tensor-based BSS methods, namely TenSOFO and TCBSS. The former is designed for a joint individual differences in scaling (INDSCAL) decomposition, addressing instantaneous (linear) BSS tasks; while the latter efficiently performs a constrained block term decomposition (BTD), aligning with the design of convolutive BSS. Leveraging the benefits of the alternating direction method of multipliers and the strengths of tensor representations, both TenSOFO and TCBSS prove to be effective in BSS. Our experimental results demonstrate the effectiveness of these two proposed methods in addressing both TD and BSS tasks, particularly when compared to state-of-the-art algorithms.

Keywords:

Blind source separation, tensor decomposition, block term decomposition, second-order statistics, INDSCAL decomposition, ADMM.

The material in this paper was submitted in part to present at the 2024 IEEE International Conference on Acoustics, Speech, and Signal Processing. This work was supported by the European Pro-Athena program under grant No. 20-GURE-0012. Ales Holobar was supported by the Slovenian Research and Innovation Agency (Programme No. P2-0041).

*Corresponding author

Email address: thanhletrung@vnu.edu.vn (Thanh Trung Le)

1. Introduction

In this work, we consider the following blind source separation (BSS) model:

$$x_m[t] = \sum_{r=1}^{R} y_{mr}[t] = \sum_{r=1}^{R} \sum_{\ell=0}^{L} a_{mr}[\ell] s_r[t-\ell], \tag{1}$$

where $x_m[t]$ represents the data observed at the m-th sensor ($m = 1, 2, \ldots, M$); $s_r[t]$ is the r-th source signal ($r = 1, 2, \ldots, R$); $a_{mr}[\ell]$, $\ell = 0, 1, \ldots, L$ are coefficients of the impulse response from the r-th source to the m-th sensor, and $(L+1)$ is the maximum filter length. Particularly when $L = 0$, (1) boils down to the problem of instantaneous (linear) BSS. Given a set of data observations $\{x_m[t]\}_{t\geq 0}$, it is desirable to identify the mixture process and recover the underlying source signals. In the literature, many effective methods have been proposed for BSS so far. We refer the readers to [1–3] for good references.

Over the last two decades, tensor decomposition (TD) has emerged as a powerful processing tool for analyzing multivariate and high dimensional data in both batch and adaptive settings [4–6]. With its capability to factorize multiway arrays (referred to as tensors) into basic components, TD has consistently demonstrated remarkable success in various signal processing and machine learning applications. Particularly in the context of BSS, several tensorization techniques have been introduced to transform time-series signal and data models into tensor representations, such as time-frequency tools [7], Hankelization [8], Löwnerization [9], and segmentation [10], among others. These techniques pave the way for the promising integration of tensor decomposition methods in addressing BSS tasks effectively. In this paper, we aim to investigate the problem of BSS through the lens of tensor decomposition, leveraging its advantages and advances to significantly enhance blind source separation performance.

1.1. Related Works

In the literature, several tensor-based methods have been proposed for BSS tasks [2]. Among them, many make the use of the classical canonical polyadic (CP) decomposition, which allows tensors to be expressed as a sum of rank-1 components [4]. In particular, a connection between this decomposition and joint (simultaneous) diagonalization was established in [11], which laid the foundation for various CP-based BSS algorithms. Some notable examples of CP-based BSS methods include FOOBI [12], SOBIUM [13], PARAFAC-SD [14], and DC-CPD [15]. High-order SVD or Tucker decomposition, which factorize a tensor into a smaller core tensor and a set of loading matrices, have also been employed for BSS tasks, as demonstrated in works such as [16–18]. Another tensor approach in BSS is the block component analysis or block term decomposition (BTD) which unifies the CP and Tucker decompositions [19]. Tensorization techniques like Hankelization [8], Löwnerization [9], and segmentation [10] have been developed to enable the use of BTD in BSS, particularly for source signals with specific characteristics such as exponential or polynomial behavior. Specifically, the LL1-BTD (BTD with multilinear rank-$(L, L, 1)$ terms) plays the central role in various BSS-related tasks, ensuring exact recovery of components from data observations. It has already found applications in hyperspectral unmixing [20–22], separation of biomedical signals [23–25], blind deconvolution [26, 27], and others.

The existing tensor-based BSS methods are, however, either designed for handling instantaneous BSS rather than convolutive ones or applicable only to certain classes of source signals. One potential solution is to convert convolutive mixtures into instantaneous ones, often accomplished through frequency domain representations or transformations. However, this introduces a set of new issues, including complex-valued data, permutation and scaling indeterminacies, and the consistency of filter coefficients across frequencies, to name a few [3]. Hence, there is a need to readapt current tensor-based BSS algorithms to tackle them or, alternatively, to delve deeper into exploring novel methods that can bypass such issues.

In parallel, we know that statistical properties of data play a crucial role for BSS tasks [28]. Many BSS methods have effectively utilized second-order (SO) and/or fourth-order (FO) statistics for source separation, as demonstrated in [29–32], among others. SO statistics offer insights into the correlation and linear relationships among observed signals, while FO statistics can capture the non-Gaussian nature of the sources and higher-order dependencies among the data observations. Most existing methods, however, either focus solely on one type of data statistics (e.g., SO or FO) or partially exploit their information. Consequently, our objective in this paper is to integrate SO and FO statistics with tensor analysis for BSS. This approach is expected to provide a more comprehensive set of statistical features and leverage the benefits of tensor representation, leading to enhanced robustness and accuracy in source separation performance.

1.2. Main Contributions

Following our preliminary study presented in [33, 34], the main contributions of this work can be summarized as follows. We first contribute to the literature on BSS by introducing a new tensor method that effectively leverages data statistics, namely TenSOFO (where "Ten", "SO", and "FO" stand for Tensors, Second Order statistics, and Fourth Order statistics, respectively). The proposed method is based on a novel joint (simultaneous) analysis called individual differences in scaling (INDSCAL), which is a symmetric variant of CP for third-order tensors with symmetry in two modes [35]. Particularly, this analysis offers improved interpretability as compared to the classical CP model for BSS, which uses SO and FO represented by symmetric matrices (covariance) and tensors (quadricovariance), respectively. We propose an effective alternating direction method of multipliers (ADMM)-based joint INDSCAL decomposition of two symmetric third-order tensors. In addition, we establish a fundamental link between SO-FO based BSS and joint INDSCAL decomposition, allowing us to effectively apply the proposed INDSCAL method for instantaneous BSS tasks.

In contrast to the existing BSS methods using LL1-BTD, which primarily employ low-rank constraints on the underlying source signals, the joint INDSCAL model presents an alternative approach leveraging nice properties of higher-order statistics (HOS) in data observations. Specifically, the proposed method offers several appealing features over LL1-BTD for BSS tasks, including (i) a memory-saving representation and efficient decomposition, (ii) vanishing of the Gaussian noise terms when using FO statistics, (iii) the ability to handle source signals without prior information on their rank, and among others. Instantaneous BSS methods based on LL1-BTD directly construct tensors from data observations (e.g., by tensorizing signals through Hankelization, segmentation or Löwnerization), which can result in large tensors. In contrast, TenSOFO employs data statistics to construct two third-order tensors of small size (refer to Section III for details). Consequently, TenSOFO provides a more efficient memory-saving tensor representation and decomposition, particularly advantageous for processing long time-series signals like speech and audio sampled at high rates. Additionally, TenSOFO, along with other HOS-based methods, benefits from the inherent property that higher-order cumulants of Gaussian random variables are zero, making it less sensitive to the effect of Gaussian noise. Another advantage of the proposed method over LL1-BTD is as follows: LL1-BTD based methods often require rank information of source signals, which may not always be available or is challenging to determine in practice. In contrast, the proposed method operates effectively without such information, performing well under statistical independence assumptions. Furthermore, the proposed method differs from the existing coupled tensor-based BSS method, DC-CPD [15], in several aspects. DC-CPD is designed to address joint BSS tasks involving multi-set signals. It constructs third-order tensors from cross-covariance matrices, assuming intra-set independence and inter-set dependence. Therefore, the symmetry in DC-CPD arises from concatenating these tensors and their permuted versions (see Section I in Supplementary materials of [15] for details). In contrast, our method involves the computation of covariance and quadricovari-

ance of data at pairs of time points, resulting in an intrinsic symmetric structure derived directly from these covariance matrices and quadricovariance tensors. This intrinsic symmetry facilitates the formulation of the joint INDSCAL, also known as symmetric CP model. Additionally, DC-CPD relies solely on SO statistics, whereas TenSOFO incorporates both SO and FO statistics. This distinction makes our method more comprehensive and effective in addressing BSS tasks.

Next, we propose the second tensor method aimed at dealing with convolutive BSS directly in time domain. In contrast to existing tensor-based BSS methods, our approach involves exploiting the merits of a special variant of the block term decomposition (BTD) where the loading factors are constrained to be identical. To elaborate, we establish a connection between convolutive BSS and this constrained BTD, forming the groundwork for our novel method called TCBSS (which stands for Tensor based Convolutive BSS). Specifically by exploiting second-order statistics, we first construct a third-order tensor by stacking a set of covariance matrices, and then, apply TCBSS to identify the mixing process and sources. A variant of TCBSS, incorporating fourth-order statistics, is also introduced.

By reformulating BSS within the framework of tensor decomposition, several advantageous properties of INDSCAL and BTD can be effectively leveraged for BSS tasks. Specifically, the identifiability issues inherent in BSS are closely linked to the uniqueness of solutions derived from these tensor-based approaches. The two proposed tensor models are essentially unique under mild conditions (i.e., unique up to a scale and a permutation), thus aiding in addressing identifiability issues in BSS. This is also feasible in cases when the number of sources exceeds the number of mixtures, as demonstrated by Proposition 2 in Section 3.1. Additionally, tensor decomposition allows for flexible modeling of the data, enabling the incorporation of constraints and assumptions tailored to specific BSS tasks. Moreover, the utilization of tensor models holds promise for enhancing separation performance by capturing higher-order statistical dependencies inherent in the data. This is illustrated by several

experimental results on both synthetic and real datasets in Section 5.

Beyond its primary contributions to BSS, this paper also enriches the existing tensor decomposition literature by introducing two effective optimization approaches for factorizing tensors under the joint INDSCAL and BTD models, outperforming the state-of-the-art algorithms.

1.3. Paper Organization

The remainder of this paper is structured as follows. In Section 2, we provide a brief overview of the preliminaries, covering standard notations, operations, tensor decomposition models, data statistics, and the alternating direction method of multipliers. Section 3 establishes a fundamental link between instantaneous BSS and joint INDSCAL decomposition, subsequently introducing our first method, TenSOFO. Section 4 introduces the second method, TCBSS, designed to address convolutive BSS tasks. In Section V, we present extensive experiments to demonstrate the effectiveness of these two proposed method and Section VI concludes the paper.

2. Preliminaries

2.1. Notations and Operations

In this paper, we use the following notations. Lowercase letters represent scalars (e.g., x), while boldface letters indicate vectors (e.g., $\mathbf{x}$). Matrices and tensors are denoted using boldface capital letters (e.g., $\mathbf{X}$) and bold calligraphic letters (e.g., $\boldsymbol{\mathcal{X}}$), respectively. The $(i_1, i_2, \ldots, i_N)$-th element of a tensor $\boldsymbol{\mathcal{X}}$ is denoted as $\boldsymbol{\mathcal{X}}(i_1, i_2, \ldots, i_N)$, $(\boldsymbol{\mathcal{X}})_{i_1 i_2 \ldots i_N}$, or $x_{i_1 i_2 \ldots i_N}$. The mode-n matricization of a tensor $\boldsymbol{\mathcal{X}}$ is denoted as $[\boldsymbol{\mathcal{X}}]_{(n)}$. The transpose operation is represented as $(\cdot)^\intercal$, the pseudo-inverse as $(\cdot)^\#$, and the Frobenius norm as $\|\cdot\|_F$. The mode-n product of a tensor $\boldsymbol{\mathcal{X}}$ and a matrix $\mathbf{U}$ is denoted as $\boldsymbol{\mathcal{X}} \times_n \mathbf{U}$. The concatenation of tensors $\boldsymbol{\mathcal{X}}$ and $\boldsymbol{\mathcal{Y}}$ along the last mode is represented as $\boldsymbol{\mathcal{X}} \boxplus \boldsymbol{\mathcal{Y}}$. Symbols $\circ$, $\odot$, and $\circledast$ are

used to represent the outer, Khatri-Rao, and Hadamard products, respectively. Symbols $\otimes$ and $\boxtimes$ denote the Kronecker and block-wise Kronecker products. The function "blkdiag" constructs a block diagonal matrix or tensor by arranging the inputs along its diagonal. The function "length" returns the number of entries in a vector, matrix, or tensor. The function "mat" reshapes a tensor $\boldsymbol{\mathcal{X}}$ of size $I \times J \times K \times L$ into a matrix $\mathbf{X}$ of size $IJ \times KL$ such that $\mathbf{X}((i-1)I + j, (k-1)I + l) = \boldsymbol{\mathcal{X}}(i, j, k, l)$. In the following, we present some frequently-used mathematical operations considered in this paper.

The mode-n product of a tensor $\boldsymbol{\mathcal{X}} \in \mathbb{R}^{I_1 \times I_2 \times \cdots \times I_N}$ and a matrix $\mathbf{U} \in \mathbb{R}^{J_n \times I_n}$ returns a new tensor $\boldsymbol{\mathcal{Y}} = \boldsymbol{\mathcal{X}} \times_n \mathbf{U} \in \mathbb{R}^{I_1 \times \cdots \times I_{n-1} \times J_n \times I_{n+1} \times \cdots \times I_N}$ whose elements are given by

$$\begin{aligned} &\boldsymbol{\mathcal{Y}}(i_1, \ldots, i_{n-1}, j_n, i_{n+1}, \ldots, i_N) \\ &\qquad = \sum_{i_n=1}^{I_n} \boldsymbol{\mathcal{X}}(i_1, \ldots, i_{n-1}, i_n, i_{n+1}, \ldots, i_N)\mathbf{U}(j_n, i_n). \end{aligned} \tag{2}$$

If $\boldsymbol{\mathcal{Y}} = \boldsymbol{\mathcal{X}} \times_n \mathbf{U}$, then $[\boldsymbol{\mathcal{Y}}]_{(n)} = \mathbf{U}[\boldsymbol{\mathcal{X}}]_{(n)}$ $\forall n$.

The concatenation of two tensors $\boldsymbol{\mathcal{X}} \in \mathbb{R}^{I_1 \times \cdots \times I_{N-1} \times I_N}$ and $\boldsymbol{\mathcal{Y}} \in \mathbb{R}^{I_1 \times \cdots \times I_{N-1} \times J_N}$ along the last mode yields a new tensor $\boldsymbol{\mathcal{Z}} = \boldsymbol{\mathcal{X}} \boxplus \boldsymbol{\mathcal{Y}} \in \mathbb{R}^{I_1 \times \cdots \times I_{N-1} \times (I_N + J_N)}$ whose elements are defined as

$$\boldsymbol{\mathcal{Z}}(i_1, \ldots, i_{n-1}, k) = \begin{cases} \boldsymbol{\mathcal{X}}(i_1, \ldots, i_{n-1}, k) & \text{if } k \leq I_N, \\ \boldsymbol{\mathcal{Y}}(i_1, \ldots, i_{n-1}, k) & \text{if } k > I_N. \end{cases} \tag{3}$$

The Hadamard product (aka elementwise product) of two matrices $\mathbf{A} \in \mathbb{R}^{M \times N}$ and $\mathbf{B} \in \mathbb{R}^{M \times N}$ is defined as

$$\mathbf{A} * \mathbf{B} = \begin{bmatrix} a_{11}b_{11} & \ldots & a_{1N}b_{1N} \\ \vdots & \ddots & \vdots \\ a_{M1}b_{M1} & \ldots & a_{MN}b_{MN} \end{bmatrix}. \tag{4}$$

The Kronecker product of two matrices $\mathbf{A} \in \mathbb{R}^{M \times N}$ and $\mathbf{B} \in \mathbb{R}^{P \times Q}$ results in

$$\mathbf{A} \otimes \mathbf{B} = \begin{bmatrix} a_{11}\mathbf{B} & \ldots & a_{1N}\mathbf{B} \\ \vdots & \ddots & \vdots \\ a_{M1}\mathbf{B} & \ldots & a_{MN}\mathbf{B} \end{bmatrix}. \tag{5}$$

The Khatri-Rao product (aka column-wise Kronecker product) of $\mathbf{A} = [\mathbf{a}_1, \mathbf{a}_2, \ldots, \mathbf{a}_R]$ and $\mathbf{B} = [\mathbf{b}_1, \mathbf{b}_2, \ldots, \mathbf{b}_R]$ yields

$$\mathbf{A} \odot \mathbf{B} = \big[\mathbf{a}_1 \otimes \mathbf{b}_1, \mathbf{a}_2 \otimes \mathbf{b}_2, \ldots, \mathbf{a}_R \otimes \mathbf{b}_R\big]. \tag{6}$$

When $\mathbf{B} = \mathbf{A}$, we denote (6) as $\mathbf{A}_{\odot 2} = \mathbf{A} \odot \mathbf{A}$ for short.

The block-wise Kronecker product of $\mathbf{A} = [\mathbf{A}_1, \mathbf{A}_2, \ldots, \mathbf{A}_R]$ and $\mathbf{B} = [\mathbf{B}_1, \mathbf{B}_2, \ldots, \mathbf{B}_R]$ is denoted as

$$\mathbf{A} \boxtimes \mathbf{B} = \big[\mathbf{A}_1 \otimes \mathbf{B}_1, \mathbf{A}_2 \otimes \mathbf{B}_2, \ldots, \mathbf{A}_R \otimes \mathbf{B}_R\big]. \tag{7}$$

2.2. Tensor Decomposition

We provide a brief overview of two tensor formats: INDSCAL and type-2 BTD, which are utilized in our work.

2.2.1. INDSCAL

The INdividual Differences in SCALing (INDSCAL) represents a variant of CP/PARAFAC decomposition that enables the factorization of symmetric tensors [4, 35]. Under the INDSCAL model, a third-order tensor $\boldsymbol{\mathcal{X}}$ of size $I \times I \times K$ with elements satisfying $x_{ijk} = x_{jik}$ for all i, j, k can be decomposed into two factors $\mathbf{A} \in \mathbb{R}^{I \times R}$ and $\mathbf{C} \in \mathbb{R}^{K \times R}$ (R being the tensor rank) as follows

$$\boldsymbol{\mathcal{X}} \triangleq \big[\!\big[\mathbf{A}, \mathbf{A}, \mathbf{C}\big]\!\big] = \sum_{r=1}^{R} \mathbf{a}_r \circ \mathbf{a}_r \circ \mathbf{c}_r, \tag{8}$$

where $\mathbf{a}_r$ and $\mathbf{c}_r$ are the r-th columns of $\mathbf{A}$ and $\mathbf{C}$, respectively. Its computation typically follows the same iterative procedure to computing the classical CP decomposition (i.e., CP-ALS) [4]. In CP-ALS, the two "$\mathbf{A}$" matrices are treated as separate factors, denoted as $\mathbf{A}_{\mathtt{L}}$ and $\mathbf{A}_{\mathtt{R}}$ (for left and right, respectively), and they are updated independently without an explicit constraint enforcing their equality. Despite starting with different initial estimates, the inherent symmetry of the data together with the uniqueness of CP decomposition eventually lead the two "$\mathbf{A}$" matrices to converge, up to a scaling factor. Being a special case of

CP, the uniqueness of INDSCAL is also guaranteed under mild conditions [36].

2.2.2. Type-2 BTD

The type-2 BTD is a variant of the block term decomposition (BTD) [37]. It aims to factorize a third-order tensor $\boldsymbol{\mathcal{X}} \in \mathbb{R}^{I\times J\times K}$ into a set of low multilinear-rank components $\{\boldsymbol{\mathcal{X}}_r\}_{r=1}^{R}$ as follows

$$\boldsymbol{\mathcal{X}} = \sum_{r=1}^{R} \boldsymbol{\mathcal{X}}_r = \sum_{r=1}^{R} \boldsymbol{\mathcal{G}}_r \times_1 \mathbf{A}_r \times_2 \mathbf{B}_r. \tag{9}$$

Here, $\boldsymbol{\mathcal{G}}_r \in \mathbb{R}^{L_r\times M_r\times K}$ represents the core tensor of the r-th component $\boldsymbol{\mathcal{X}}_r$, while the loading factors $\mathbf{A}_r \in \mathbb{R}^{I\times L_r}$ and $\mathbf{B}_r \in \mathbb{R}^{J\times M_r}$ are full column rank matrices. Since (9) is trilinear in $\mathbf{A} = [\mathbf{A}_1, \mathbf{A}_2, \ldots, \mathbf{A}_R]$, $\mathbf{B} = [\mathbf{B}_1, \mathbf{B}_2, \ldots, \mathbf{B}_R]$, and $\boldsymbol{\mathcal{G}} = \text{blkdiag}(\boldsymbol{\mathcal{G}}_1, \boldsymbol{\mathcal{G}}_2, \ldots, \boldsymbol{\mathcal{G}}_R)$, its computation follows the common alternating least-squares (ALS) approach [38]. Also, the type-2 BTD is essential unique under mild conditions [37]. To support our algorithm development, we present three mode-n matrix representations of $\boldsymbol{\mathcal{X}}$:

$$\big[\boldsymbol{\mathcal{X}}\big]_{(1)} = \mathbf{A}\Big[\big[\boldsymbol{\mathcal{G}}_1 \times_2 \mathbf{B}_1\big]_{(1)}^{\mathsf{T}}, \ldots, \big[\boldsymbol{\mathcal{G}}_R \times_2 \mathbf{B}_R\big]_{(1)}^{\mathsf{T}}\Big]^{\mathsf{T}}, \tag{10}$$

$$\big[\boldsymbol{\mathcal{X}}\big]_{(2)} = \mathbf{B}\Big[\big[\boldsymbol{\mathcal{G}}_1 \times_1 \mathbf{A}_1\big]_{(2)}^{\mathsf{T}}, \ldots, \big[\boldsymbol{\mathcal{G}}_R \times_1 \mathbf{A}_R\big]_{(2)}^{\mathsf{T}}\Big]^{\mathsf{T}}, \tag{11}$$

$$\big[\boldsymbol{\mathcal{X}}\big]_{(3)} = \Big[\big[\boldsymbol{\mathcal{G}}_1\big]_{(3)}, \big[\boldsymbol{\mathcal{G}}_2\big]_{(3)}, \ldots, \big[\boldsymbol{\mathcal{G}}_R\big]_{(3)}\Big]\big(\mathbf{B} \boxtimes \mathbf{A}\big)^{\mathsf{T}}. \tag{12}$$

2.3. Data Statistics

Consider a data vector $\mathbf{u}[t] \in \mathbb{R}^I$ of zero mean. We can analyze its second-order (SO) and fourth-order (FO) statistics using its covariance matrix $\mathbf{R}^{\mathbf{u}} \in \mathbb{R}^{I\times I}$ and quadricovariance

tensor $\boldsymbol{\mathcal{C}}^{\mathbf{u}} \in \mathbb{R}^{I\times I\times I\times I}$, respectively. These statistics are defined as follows:[1]

$$\mathbf{R}_{\mathbf{u}}[t,\tau] = \mathbb{E}\Big\{\mathbf{u}[t]\mathbf{u}[t-\tau]^{\top}\Big\}, \tag{13}$$

$$\Big(\boldsymbol{\mathcal{C}}_{\mathbf{u}}[t,\{\boldsymbol{\tau}\}]\Big)_{ijkl} = \mathrm{Cum}\Big\{u_i[t], u_j[t-\tau_1], u_k[t-\tau_2], u_l[t-\tau_3]\Big\}, \tag{14}$$

for the time lags τ and $\boldsymbol{\tau} \equiv \{\tau_1,\tau_2,\tau_3\}$, where $u_i[t]$ is the i-th entry of $\mathbf{u}[t]$ and the fourth-order cumulant "$\mathrm{Cum}(\cdot)$" is given by

$$\begin{aligned}\mathrm{Cum}\big\{x_i,x_j,x_k,x_l\big\} = {}& \mathbb{E}\big\{x_ix_jx_kx_l\big\} - \mathbb{E}\big\{x_ix_j\big\}\mathbb{E}\big\{x_kx_l\big\} \\ & - \mathbb{E}\big\{x_ix_k\big\}\mathbb{E}\big\{x_jx_l\big\} - \mathbb{E}\big\{x_ix_l\big\}\mathbb{E}\big\{x_jx_k\big\}.\end{aligned} \tag{15}$$

2.4. Alternating Direction Method of Multipliers

Alternating Direction Method of Multipliers (ADMM) is an effective primal-dual optimization framework designed to deal with convex constrained problems of the following form [39]

$$\min_{\mathbf{x},\mathbf{y}} f(\mathbf{x}) + g(\mathbf{y}) \quad \text{subject to} \quad \mathbf{z}(\mathbf{x},\mathbf{y}) = \mathbf{c}, \tag{16}$$

where the objective function is separable in $\mathbf{x}$ and $\mathbf{y}$, and $\mathbf{z}(\mathbf{x},\mathbf{y}) = \mathbf{c}$ represents constraints on parameters of interest. The augmented Lagrangian corresponding to (16) is given by

$$\mathcal{L}(\mathbf{x},\mathbf{y},\boldsymbol{\mu}) = f(\mathbf{x}) + g(\mathbf{y}) + \frac{\rho}{2}\big\|\mathbf{c}-\mathbf{z}(\mathbf{x},\mathbf{y})\big\|_F^2 + \boldsymbol{\mu}^{\top}\big(\mathbf{c}-\mathbf{z}(\mathbf{x},\mathbf{y})\big), \tag{17}$$

where $\rho > 0$ is a regularization parameter and $\boldsymbol{\mu}$ is the dual variable. ADMM relies on the duality theory for convex optimization, where the objective is to minimize the augmented Lagrangian w.r.t. $\mathbf{x}$, $\mathbf{y}$ and a fixed $\boldsymbol{\mu}$. Conversely, the dual function $h(\boldsymbol{\mu}) = \min_{\mathbf{x},\mathbf{y}} \mathcal{L}(\mathbf{x},\mathbf{y},\boldsymbol{\mu})$ should be maximized w.r.t. $\boldsymbol{\mu}$. Consequently, ADMM performs an alternation between minimizing $\mathcal{L}(\cdot)$ w.r.t. $\mathbf{x}$ and $\mathbf{y}$ and employing gradient ascent to maximize $h(\boldsymbol{\mu})$.

[1] In this work, we focus on tensor decomposition with real-valued data (source signals and mixtures). The definitions of $\mathbf{R}_{\mathbf{u}}$ and $\boldsymbol{\mathcal{C}}_{\mathbf{u}}$ can be derived following (13) and (14), respectively. In a very general way, it is essential to consider the conjugate complex elements when dealing with complex data at various time lags. Our proposed methods, presented in the following sections, can be readily adapted to address complex-valued scenarios. These extensions can be achieved by substituting the transpose $(\cdot)^{\top}$ with the conjugate (Hermitian) transpose $(\cdot)^{H}$, with the exception of certain reshaping and unfolding operators such as (63) and (64).

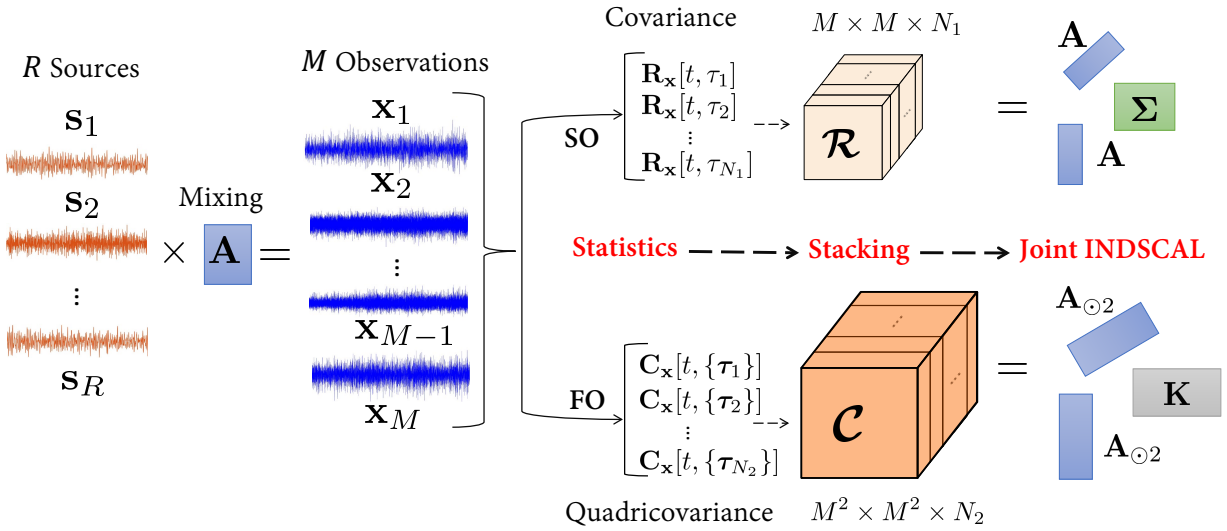


Figure 1: TenSOFO – A Joint INDSCAL Decomposition based Instantaneous BSS. TenSOFO comprises three main steps, including (i) extracting second-order (SO) and fourth-order (FO) statistics from observations to obtain two sets of covariance matrices $\{\mathbf{R}_{\mathbf{x}}[t,\tau_n]\}_{n=1}^{N_1}$ and quadricovariance matrices $\{\mathbf{C}_{\mathbf{x}}[t,\{\boldsymbol{\tau}_n\}]\}_{n=1}^{N_2}$; (ii) forming two third-order tensors $\boldsymbol{\mathcal{R}}$ and $\boldsymbol{\mathcal{C}}$, by stacking $\{\mathbf{R}_{\mathbf{x}}[t,\tau_n]\}_{n=1}^{N_1}$ and $\{\mathbf{R}_{\mathbf{x}}[t,\tau_n]\}_{n=1}^{N_1}$, respectively; and (iii) performing a joint INDSCAL decomposition of $\boldsymbol{\mathcal{R}}$ and $\boldsymbol{\mathcal{C}}$ to derive the mixing matrix $\mathbf{A}$.

3. Instantaneous BSS: Joint INDSCAL Decomposition based Method

In this section, we present a link between joint INDSCAL decomposition and instantaneous BSS. Subsequently, we introduce a novel tensor method for BSS named TenSOFO which effectively leverages SO and FO statistics. See Fig. 1 for an illustration.

3.1. Link between Instantaneous BSS and Joint INDSCAL Decomposition

The convolutive BSS model (1) can be transformed into an instantaneous BSS model when the filter length is reduced to one (i.e., $L = 0$). In this case, we can recast (1) into the following linear model

$$\mathbf{x}[t] = \mathbf{A}\mathbf{s}[t], \tag{18}$$

where $\mathbf{x}[t] = \big[x_1[t], x_2[t], \ldots, x_M[t]\big]^\top \in \mathbb{R}^M$ represents the data observation vector, $\mathbf{s}[t] = \big[s_1[t], s_2[t], \ldots, s_R[t]\big]^\top \in \mathbb{R}^R$ is the source vector, and $\mathbf{A} \in \mathbb{R}^{M\times R}$ is the mixing matrix defined as

$$\mathbf{A} = \begin{bmatrix} a_{11}[0] & \ldots & a_{1R}[0] \\ \vdots & \ddots & \vdots \\ a_{M1}[0] & \ldots & a_{MR}[0] \end{bmatrix}. \tag{19}$$

In many applications, the underlying sources are stationary, non-Gaussian and mutually statistically independent, while being individually correlated for different lags, such as [30, 40, 41]. In such cases, the computation of second-order (SO) statistics of $\mathbf{x}[t]$ leads to the following relation

$$\mathbf{R}_{\mathbf{x}}[t,\tau] = \mathbf{A}\mathbb{E}\big\{\mathbf{s}[t]\mathbf{s}[t-\tau]^\top\big\}\mathbf{A}^\top = \mathbf{A}\mathbf{R}_{\mathbf{s}}[t,\tau]\mathbf{A}^\top, \tag{20}$$

$$\mathbf{R}_{\mathbf{s}}[t,\tau] = \mathrm{diag}\Big\{\sigma_1^2[\tau], \sigma_2^2[\tau], \ldots, \sigma_R^2[\tau]\Big\}, \tag{21}$$

where $\sigma_r^2[\tau] = \mathbb{E}\big\{s_r[t]s_r[t-\tau]\big\}$. While the computation of fourth-order (FO) statistics of $\mathbf{x}[t]$ results in

$$\Big(\boldsymbol{\mathcal{C}}_{\mathbf{x}}[t,\{\boldsymbol{\tau}\}]\Big)_{ijkl} = \sum_{r_1r_2r_3r_4} \Big(\boldsymbol{\mathcal{C}}_{\mathbf{s}}[t,\{\boldsymbol{\tau}\}]\Big)_{r_1r_2r_3r_4} \mathbf{A}(i,r_1) \mathbf{A}(j,r_2)\mathbf{A}(k,r_3)\mathbf{A}(l,r_4), \tag{22}$$

for all index values. Upon transforming (22) into matrix form, the resulting expression is as follows

$$\mathbf{C}_{\mathbf{x}}[t,\{\boldsymbol{\tau}\}] = \big(\mathbf{A}\otimes\mathbf{A}\big)\widetilde{\mathbf{C}}_{\mathbf{s}}[t,\{\boldsymbol{\tau}\}]\big(\mathbf{A}\otimes\mathbf{A}\big)^\top, \tag{23}$$

where $\mathbf{C}_{\mathbf{x}}[t,\{\boldsymbol{\tau}\}] = \mathrm{mat}(\boldsymbol{\mathcal{C}}_{\mathbf{x}}[t,\{\boldsymbol{\tau}\}]) \in \mathbb{R}^{M^2\times M^2}$ and $\widetilde{\mathbf{C}}_{\mathbf{s}}[t,\{\boldsymbol{\tau}\}] = \mathrm{mat}\big(\boldsymbol{\mathcal{C}}_{\mathbf{s}}[t,\{\boldsymbol{\tau}\}]\big) \in \mathbb{R}^{R^2\times R^2}$. When the underlying sources are statistically independent, the matrix $\widetilde{\mathbf{C}}_{\mathbf{s}}[t,\{\boldsymbol{\tau}\}]$ contains at least $R^4 - R$ zeros. Thus, the expression (23) boils down to a simpler one

$$\mathbf{C}_{\mathbf{x}}[t,\{\boldsymbol{\tau}\}] = \big(\mathbf{A}\odot\mathbf{A}\big)\mathbf{C}_{\mathbf{s}}[t,\{\boldsymbol{\tau}\}]\big(\mathbf{A}\odot\mathbf{A}\big)^\top, \tag{24}$$

$$\mathbf{C}_{\mathbf{s}}[t,\{\boldsymbol{\tau}\}] = \mathrm{diag}\Big\{\kappa_1[\{\boldsymbol{\tau}\}], \kappa_2[\{\boldsymbol{\tau}\}], \ldots, \kappa_R[\{\boldsymbol{\tau}\}]\Big\}, \tag{25}$$

where $\kappa_r[\{\boldsymbol{\tau}\}] = \mathrm{Cum}\big\{s_r[t], s_r[t-\tau_1], s_r[t-\tau_2], s_r[t-\tau_3]\big\}$. To sum up, we obtain

$$\mathbf{R}_{\mathbf{x}}[t,\tau] = \mathbf{A}\mathbf{R}_{\mathbf{s}}[t,\tau]\mathbf{A}^{\top} \quad \text{and} \quad \mathbf{C}_{\mathbf{x}}[t,\{\boldsymbol{\tau}\}] = \mathbf{A}_{\odot 2}\mathbf{C}_{\mathbf{s}}[t,\{\boldsymbol{\tau}\}]\mathbf{A}_{\odot 2}^{\top}, \tag{26}$$

where $\mathbf{A}_{\odot 2} = \mathbf{A} \odot \mathbf{A}$.

In this work, we construct two tensors $\boldsymbol{\mathcal{R}} \in \mathbb{R}^{M\times M\times N_1}$ and $\boldsymbol{\mathcal{C}} \in \mathbb{R}^{M^2\times M^2\times N_2}$ from the set of N_1 matrices $\big\{\mathbf{R}_{\mathbf{x}}[t,\tau_n]\big\}_{n=1}^{N_1}$ and N_2 matrices $\big\{\mathbf{C}_{\mathbf{x}}[t,\boldsymbol{\tau}_n]\big\}_{n=1}^{N_2}$ as follows

$$\boldsymbol{\mathcal{R}} = \mathbf{R}_{\mathbf{x}}[t,\tau_1] \boxplus \mathbf{R}_{\mathbf{x}}[t,\tau_2] \cdots \boxplus \mathbf{R}_{\mathbf{x}}[t,\tau_{N_1}], \tag{27}$$

$$\boldsymbol{\mathcal{C}} = \mathbf{C}_{\mathbf{x}}[t,\{\boldsymbol{\tau}_1\}] \boxplus \mathbf{C}_{\mathbf{x}}[t,\{\boldsymbol{\tau}_2\}] \cdots \boxplus \mathbf{C}_{\mathbf{x}}[t,\{\boldsymbol{\tau}_{N_2}\}], \tag{28}$$

i.e., $\boldsymbol{\mathcal{R}}(:,:,n_1) = \mathbf{R}_{\mathbf{x}}[t,\tau_{n_1}]$ and $\boldsymbol{\mathcal{C}}(:,:,n_2) = \mathbf{C}_{\mathbf{x}}[t,\{\boldsymbol{\tau}_{n_2}\}]$ for $1 \le n_1 \le N_1$ and $1 \le n_2 \le N_2$. Here, each element of $\boldsymbol{\mathcal{R}}$ and $\boldsymbol{\mathcal{C}}$ can be expressed as

$$\boldsymbol{\mathcal{R}}(i,j,n_1) = \sum_{r=1}^{R} \mathbf{A}(i,r)\mathbf{A}(j,r)\sigma_r^2[\tau_{n_1}], \tag{29}$$

$$\boldsymbol{\mathcal{C}}(k,l,n_2) = \sum_{r=1}^{R} \mathbf{A}_{\odot 2}(k,r)\mathbf{A}_{\odot 2}(l,r)\kappa_r[\{\boldsymbol{\tau}_{n_2}\}]. \tag{30}$$

Accordingly, we can represent $\boldsymbol{\mathcal{R}}$ and $\boldsymbol{\mathcal{C}}$ by

$$\boldsymbol{\mathcal{R}} = \sum_{r=1}^{R} \mathbf{A}(:,r) \circ \mathbf{A}(:,r) \circ \boldsymbol{\Sigma}(:,r), \tag{31}$$

$$\boldsymbol{\mathcal{C}} = \sum_{r=1}^{R} \mathbf{A}_{\odot 2}(:,r) \circ \mathbf{A}_{\odot 2}(:,r) \circ \mathbf{K}(:,r), \tag{32}$$

where two matrices $\boldsymbol{\Sigma} \in \mathbb{R}^{N_1\times R}$ and $\mathbf{K} \in \mathbb{R}^{N_2\times R}$ are given by

$$\boldsymbol{\Sigma} = \begin{bmatrix} \sigma_1^2[\tau_1] & \sigma_2^2[\tau_1] & \dots & \sigma_R^2[\tau_1] \\ \sigma_1^2[\tau_2] & \sigma_2^2[\tau_2] & \dots & \sigma_R^2[\tau_2] \\ \vdots & \vdots & \vdots & \vdots \\ \sigma_1^2[\tau_{N_1}] & \sigma_2^2[\tau_{N_1}] & \dots & \sigma_R^2[\tau_{N_1}] \end{bmatrix}, \tag{33}$$

$$\mathbf{K} = \begin{bmatrix} \kappa_1[\{\boldsymbol{\tau}_1\}] & \kappa_2[\{\boldsymbol{\tau}_1\}] & \dots & \kappa_R[\{\boldsymbol{\tau}_1\}] \\ \kappa_1[\{\boldsymbol{\tau}_2\}] & \kappa_2[\{\boldsymbol{\tau}_2\}] & \dots & \kappa_R^2[\{\boldsymbol{\tau}_2\}] \\ \vdots & \vdots & \vdots & \vdots \\ \kappa_1[\{\boldsymbol{\tau}_{N_2}\}] & \kappa_2[\{\boldsymbol{\tau}_{N_2}\}] & \dots & \kappa_R[\{\boldsymbol{\tau}_{N_2}\}] \end{bmatrix}. \tag{34}$$

In other words, they admit the following INDSCAL factorization (8)

$$\boldsymbol{\mathcal{R}} = \left[\!\left[\mathbf{A}, \mathbf{A}, \boldsymbol{\Sigma}\right]\!\right] \quad \text{and} \quad \boldsymbol{\mathcal{C}} = \left[\!\left[\mathbf{A}_{\odot 2}, \mathbf{A}_{\odot 2}, \mathbf{K}\right]\!\right]. \tag{35}$$

Interestingly, under mild (and easy-to-check) conditions, the joint (coupled) INDSCAL decomposition (35) is essentially unique, as highlighted in Propositions 1 and 2.

Proposition 1 (General case)**.** *Let* $N = \min(N_1, N_2)$*. If either* $R < M$ *or* $M \leq R \leq \min(N, 2M-2)$ *or* $\max(M, N_1, N_2) \leq R \leq (2M+N-2)/2$*, then* (35) *is essentially unique.*

Proposition 2 (Underdetermined case)**.** *Let* $N = \max(N_1, N_2)$*. When*

$$\begin{cases} R \leq \dfrac{M^2 - M}{2} & \text{if } \ 2 \leq M \leq R \leq N \\ R \leq \dfrac{N + 3M - 1 - \sqrt{(N-M)^2 + 2N + 6M - 3}}{2} & \text{if } \ 2 \leq N \leq M \leq R \\ R \leq \dfrac{2M + 2N + 1 - \sqrt{8N + 8M + 1}}{2} & \text{if } \ 2 \leq M \leq M \leq R \end{cases}, \tag{36}$$

then (35) *is essentially unique.*

The conditions of Proposition 1 follow from Kruskal's condition, while those of Proposition 2 are corollaries derived from Propositions 1.1, 1.2 and 1.3 in [36]. Propositions 1 and 2 hold when the elements of the loading factors in the joint tensor decomposition randomly drawn from continuous distributions. As a result, through the joint INDSCAL decomposition of both $\boldsymbol{\mathcal{R}}$ and $\boldsymbol{\mathcal{C}}$ in (35), we directly estimate the mixing matrix $\mathbf{A}$. In next subsection, we present an efficient optimization framework to simultaneously decompose $\boldsymbol{\mathcal{R}}$ and $\boldsymbol{\mathcal{C}}$.

3.2. Optimization Framework

The joint INDSCAL decomposition of $\boldsymbol{\mathcal{R}}$ and $\boldsymbol{\mathcal{C}}$ in (35) can be obtained by solving the following constrained minimization

$$\min f(\mathbf{B}, \mathbf{K}) + g(\mathbf{A}, \boldsymbol{\Sigma}) \quad \text{subject to} \quad \mathbf{B} = \mathbf{A}_{\odot 2}, \tag{37}$$

where $f(\mathbf{B}, \mathbf{K}) = \left\|\boldsymbol{\mathcal{C}} - [\![\mathbf{B}, \mathbf{B}, \mathbf{K}]\!]\right\|_F^2$ and $g(\mathbf{A}, \boldsymbol{\Sigma}) = \left\|\boldsymbol{\mathcal{R}} - [\![\mathbf{A}, \mathbf{A}, \boldsymbol{\Sigma}]\!]\right\|_F^2$. Here, (37) can be expressed in the ADMM form, and thus, we can construct the corresponding augmented

Algorithm 1: TENSOFO

Input: $\boldsymbol{\mathcal{R}} \in \mathbb{R}^{M\times M\times N_1}$ and $\boldsymbol{\mathcal{C}} \in \mathbb{R}^{M^2\times M^2\times N_2}$

Output: Loading factors $\mathbf{A}, \boldsymbol{\Sigma}$, and $\mathbf{K}$.

Initialization:

$l = 1$, $\mathbf{U}^{(0)} = \mathbf{0}_{M^2\times R}$, any $\mathbf{A}^{(0)} \in \mathbb{R}^{M\times R}$

while *stopping criteria are not met* **do**

$$\left\{\mathbf{B}^{(l)}, \mathbf{K}^{(l)}\right\} = \underset{\mathbf{B},\mathbf{K}}{\operatorname{argmin}} \left\{ f(\mathbf{B},\mathbf{K}) + \frac{\rho_0}{2}\left\|\mathbf{B} - \mathbf{A}_{\odot 2}^{(l-1)} + \mathbf{U}^{(l-1)}\right\|_F^2 \right\} \tag{P1.1}$$

$$\left\{\mathbf{A}^{(l)}, \boldsymbol{\Sigma}^{(l)}\right\} = \underset{\mathbf{A},\boldsymbol{\Sigma}}{\operatorname{argmin}} \left\{ g(\mathbf{A},\boldsymbol{\Sigma}) + \frac{\rho_0}{2}\left\|\mathbf{B}^{(l)} - \mathbf{A}_{\odot 2} + \mathbf{U}^{(l-1)}\right\|_F^2 \right\} \tag{P1.2}$$

$$\mathbf{U}^{(l)} = \mathbf{U}^{(l-1)} + \mathbf{B}^{(l)} - \mathbf{A}_{\odot 2}^{(l)} \tag{P1.3}$$

$l = l + 1$

end

Lagrangian function with a parameter ρ_0 as follows

$$\mathcal{L}_0(\mathbf{B},\mathbf{K},\mathbf{A},\boldsymbol{\Sigma},\mathbf{Z}) = f(\mathbf{B},\mathbf{K}) + g(\mathbf{A},\boldsymbol{\Sigma}) + \left\langle \mathbf{Z}, \mathbf{B} - \mathbf{A}_{\odot 2}\right\rangle + \frac{\rho_0}{2}\left\|\mathbf{B} - \mathbf{A}_{\odot 2}\right\|_F^2, \tag{38}$$

where $\mathbf{Z} \in \mathbb{R}^{M^2\times R}$ is the dual variable. Generally, the regularization parameter ρ_0 is used as the step size in the dual update [39]. Denote by $\mathbf{U} = \mathbf{Z}/\rho_0$ the scaled version of $\mathbf{Z}$, we reformulate (38) as follows

$$\mathcal{L}_0(\mathbf{B},\mathbf{K},\mathbf{A},\boldsymbol{\Sigma},\mathbf{U}) = f(\mathbf{B},\mathbf{K}) + g(\mathbf{A},\boldsymbol{\Sigma}) + \frac{\rho_0}{2}\left\|\mathbf{B} - \mathbf{A}_{\odot 2} + \mathbf{U}\right\|_F^2 - \frac{\rho_0}{2}\left\|\mathbf{U}\right\|_F^2. \tag{39}$$

Accordingly, at each iteration l, the dual update is simply computed as $\mathbf{U}^{(l)} = \mathbf{U}^{(l-1)} + \mathbf{B}^{(l)} - \mathbf{A}_{\odot 2}^{(l)}$, and it does not involve the use of ρ_0. In particular, our proposed ADMM solver is outlined in Algorithm 1. In the following, we detail the optimization approach for minimizing (P1.1) and (P1.2).

3.2.1. Updates of $\mathbf{B}^{(l)}$ *and* $\mathbf{K}^{(l)}$

Minimization (P1.1) is equivalent to the following constrained optimization

$$\begin{aligned} &\underset{\mathbf{B}_\mathrm{L},\mathbf{B}_\mathrm{R},\mathbf{K}}{\operatorname{argmin}} \quad \left\|\boldsymbol{\mathcal{C}} - [\![\mathbf{B}_\mathrm{L},\mathbf{B}_\mathrm{R},\mathbf{K}]\!]\right\|_F^2 + \frac{\rho_0}{2}\left\|\mathbf{B}_\mathrm{L} - \mathbf{A}_{\odot 2}^{(l-1)} + \mathbf{U}^{(l-1)}\right\|_F^2 \\ &\text{subject to} \quad \mathbf{B}_\mathrm{L} = \mathbf{B}_\mathrm{R}, \end{aligned} \tag{40}$$

where the two "$\mathbf{B}$" matrices in $f(\mathbf{B},\mathbf{K})$ are considered as separate loading factors, denoted as $\mathbf{B}_{\mathtt{L}}$ and $\mathbf{B}_{\mathtt{R}}$ for the left and right, respectively. The corresponding augmented Lagrangian function is expressed as follows

$$\mathcal{L}_{\mathtt{B}}(\mathbf{B}_{\mathtt{L}},\mathbf{B}_{\mathtt{R}},\mathbf{K},\mathbf{D}) = \left\|\mathcal{C}-[\![\mathbf{B}_{\mathtt{L}},\mathbf{B}_{\mathtt{R}},\mathbf{K}]\!]\right\|_F^2 + \frac{\rho_{\mathtt{0}}}{2}\left\|\mathbf{B}_{\mathtt{L}}-\mathbf{A}_{\odot 2}^{(l-1)}+\mathbf{U}^{(l-1)}\right\|_F^2 + \frac{\rho_{\mathtt{B}}}{2}\left\|\mathbf{B}_{\mathtt{L}}-\mathbf{B}_{\mathtt{R}}+\mathbf{D}\right\|_F^2 - \frac{\rho_{\mathtt{B}}}{2}\left\|\mathbf{D}\right\|_F^2. \quad (41)$$

Here, $\rho_{\mathtt{B}}>0$ is a regularization parameter, and $\mathbf{D}\in\mathbb{R}^{M^2\times R}$ denotes the (scaled) dual variable. To find the optimal solution of (41), the optimization process involves an iterative loop, with the i-th iteration step as follows:

$$\mathbf{K}^{(l,i)} = \left[\mathcal{C}\right]_{(3)}\left(\mathbf{B}_{\mathtt{L}}^{(l,i-1)}\odot\mathbf{B}_{\mathtt{R}}^{(l,i-1)}\right)\left(\left[\left(\mathbf{B}_{\mathtt{L}}^{(l,i-1)}\right)^{\top}\mathbf{B}_{\mathtt{L}}^{(l,i-1)}\right] * \left[\left(\mathbf{B}_{\mathtt{R}}^{(l,i-1)}\right)^{\top}\mathbf{B}_{\mathtt{R}}^{(l,i-1)}\right]\right)^{\#}, \quad (42a)$$

$$\mathbf{B}_{\mathtt{L}}^{(l,i)} = \left(\left[\mathcal{C}\right]_{(1)}\mathbf{P}_{\mathtt{R}}^{(l,i-1)} + \rho_{\mathtt{0}}\left(\mathbf{A}_{\odot 2}^{(l-1)}-\mathbf{U}^{(l-1)}\right) + \rho_{\mathtt{B}}\left(\mathbf{B}_{\mathtt{R}}^{(l,i-1)}-\mathbf{D}^{(l,i-1)}\right)\right)\left(\left(\mathbf{P}_{\mathtt{R}}^{(l,i-1)}\right)^{\top}\mathbf{P}_{\mathtt{R}}^{(l,i-1)}+(\rho_{\mathtt{0}}+\rho_{\mathtt{B}})\mathbf{I}_R\right)^{-1}, \quad (42b)$$

$$\mathbf{B}_{\mathtt{R}}^{(l,i)} = \left(\left[\mathcal{C}\right]_{(2)}\mathbf{P}_{\mathtt{L}}^{(l,i-1)} + \rho_{\mathtt{B}}\left(\mathbf{B}_{\mathtt{L}}^{(l,i-1)}-\mathbf{D}^{(l,i-1)}\right)\right)\left(\left(\mathbf{P}_{\mathtt{L}}^{(l,i-1)}\right)^{\top}\mathbf{P}_{\mathtt{L}}^{(l,i-1)}+\rho_{\mathtt{B}}\mathbf{I}_R\right)^{-1}, \quad (42c)$$

$$\mathbf{D}^{(l,i)} = \mathbf{D}^{(l,i-1)} + \mathbf{B}_{\mathtt{L}}^{(l,i)} - \mathbf{B}_{\mathtt{R}}^{(l,i)}, \quad (42d)$$

where $\mathbf{P}_{\mathtt{L}}^{(l,i-1)} = \mathbf{K}^{(l,i)}\odot\mathbf{B}_{\mathtt{L}}^{(l,i-1)}$ and $\mathbf{P}_{\mathtt{R}}^{(l,i)} = \mathbf{K}^{(l,i)}\odot\mathbf{B}_{\mathtt{R}}^{(l,i-1)}$. At the initial step ($i=0$), we set $\mathbf{K}^{(l,0)}=\mathbf{K}^{(l-1)}$ and $\mathbf{B}_{\mathtt{L}}^{(l,0)}=\mathbf{B}_{\mathtt{R}}^{(l,0)}=\mathbf{B}^{(l-1)}$. The iterative procedure (42) continues until convergence or until stopping criteria are met after I_{stop} iterations. Then, we align the two "$\mathbf{B}$" matrices as

$$\mathbf{\Lambda}_{\mathbf{B}}^{(l)} = \underset{\mathbf{\Lambda}_{\mathbf{B}}}{\operatorname{argmin}}\left\|\mathbf{B}_{\mathtt{L}}^{(l,I_{\text{stop}})}-\mathbf{B}_{\mathtt{R}}^{(l,I_{\text{stop}})}\mathbf{\Lambda}_{\mathbf{B}}\right\|_F, \quad (43)$$

where $\mathbf{\Lambda}_{\mathbf{B}}$ is a diagonal matrix. At the end, we take $\mathbf{K}^{(l)}=\mathbf{K}^{(l,I_{\text{stop}})}$ and $\mathbf{B}^{(l)}=0.5\big(\mathbf{B}_{\mathtt{L}}^{(l,I_{\text{stop}})}+\mathbf{B}_{\mathtt{R}}^{(l,I_{\text{stop}})}\mathbf{\Lambda}_{\mathbf{B}}^{(l)}\big)$.[2]

[2]In the initial stage, the iterative procedure (42) may require a large number of iterations to converge or may not converge at all. In such cases, the two matrices $\mathbf{B}_{\mathtt{L}}$ and $\mathbf{B}_{\mathtt{R}}$ might not be identical or closely aligned each other. The step (43) is essential to compute their average and determine the value of the matrix $\mathbf{B}^{(l)}$.

3.2.2. Updates of $\mathbf{A}^{(l)}$ *and* $\mathbf{\Sigma}^{(l)}$

We recast (P1.2) into the following ADMM form

$$\begin{aligned}&\underset{\mathbf{A}_{\mathtt{L}},\mathbf{A}_{\mathtt{R}},\mathbf{\Sigma}}{\operatorname{argmin}} \quad \left\|\boldsymbol{\mathcal{R}}-[\![\mathbf{A}_{\mathtt{L}},\mathbf{A}_{\mathtt{R}},\mathbf{\Sigma}]\!]\right\|_F^2+\frac{\rho_{\mathtt{0}}}{2}\left\|\mathbf{B}^{(l)}-\mathbf{A}_{\odot 2}+\mathbf{U}^{(l-1)}\right\|_F^2\\&\text{subject to}\quad \mathbf{A}_{\mathtt{L}}=\mathbf{A}_{\mathtt{R}}.\end{aligned}\tag{44}$$

In a way similar to (42), we employ an iterative procedure to update $\underline{\mathbf{A}}^{(l)}$ and $\mathbf{\Sigma}^{(l)}$. Starting at $j=0$, we initialize $\mathbf{\Sigma}^{(l,0)}=\mathbf{\Sigma}^{(l-1)}$, and obtain $\mathbf{A}_{\mathtt{L}}^{(l,0)}$ and $\mathbf{A}_{\mathtt{R}}^{(l,0)}$ from $\mathbf{A}_{\odot 2}^{(l,0)}$ which is computed as follows

$$\mathbf{A}_{\odot 2}^{(l,0)}=\Big(\big[\boldsymbol{\mathcal{R}}\big]_{(3)}^{\top}\mathbf{\Sigma}^{(l,0)}+\rho_{\mathtt{0}}\big(\mathbf{B}^{(l)}+\mathbf{U}^{(l-1)}\big)\Big)\Big(\big(\mathbf{\Sigma}^{(l,0)}\big)^{\top}\mathbf{\Sigma}^{(l,0)}+\rho_{\mathtt{0}}\mathbf{I}\Big)^{-1}.\tag{45}$$

Specifically, we know that $\mathbf{A}_{\odot 2}^{(l,0)}=\mathbf{A}_{\mathtt{L}}^{(l,0)}\odot\mathbf{A}_{\mathtt{R}}^{(l,0)}$ leads to

$$\mathbf{A}_{\odot 2}^{(l,0)}(:,r)=\mathrm{vec}\Big\{\mathbf{A}_{\mathtt{L}}^{(l,0)}(:,r)\mathbf{A}_{\mathtt{R}}^{(l,0)}(:,r)^{\top}\Big\}\ \text{with}\ 1\le r\le R.\tag{46}$$

Therefore, we can obtain the (normalized) r-th column of $\mathbf{A}_{\mathtt{L}}^{(l,0)}$ and $\mathbf{A}_{\mathtt{R}}^{(l,0)}$ from the most dominant left and right singular vectors of $\mathrm{reshape}\big\{\mathbf{A}_{\odot 2}^{(l,0)}(:,r),[M,M]\big\}$, respectively.

For short, we denote $\mathbf{F}^{(l)}=\mathbf{B}^{(l)}+\mathbf{U}^{(l-1)}$, $\mathbf{Q}_{\mathtt{L}}^{(l,j-1)}=\mathbf{\Sigma}^{(l,j)}\odot\mathbf{A}_{\mathtt{L}}^{(l,j-1)}$, $\mathbf{Q}_{\mathtt{R}}^{(l,j-1)}=\mathbf{\Sigma}^{(l,j)}\odot\mathbf{A}_{\mathtt{R}}^{(l,j-1)}$, $\mathbf{G}_{\mathtt{L}}^{(l,j-1)}=(\mathbf{I}_R\odot\mathbf{A}_{\mathtt{L}}^{(l,j-1)})\otimes\mathbf{I}_M$, $\mathbf{G}_{\mathtt{R}}^{(l,j-1)}=(\mathbf{I}_R\odot\mathbf{A}_{\mathtt{R}}^{(l,j-1)})\otimes\mathbf{I}_M$, $\mathbf{T}_{\mathtt{L}}^{(l,j-1)}=\mathrm{reshape}\{\mathbf{G}_{\mathtt{L}}^{(l,j-1)}\mathrm{vec}\{\mathbf{F}^{(l)}\},[M,R]\}$, $\mathbf{T}_{\mathtt{R}}^{(l,j-1)}=\mathrm{reshape}\{\mathbf{G}_{\mathtt{R}}^{(l,j-1)}\mathrm{vec}\{\mathbf{F}^{(l)}\},[M,R]\}$, and read

$$\begin{aligned}\mathbf{\Sigma}^{(l,j)}=\big[\boldsymbol{\mathcal{R}}\big]_{(3)}\Big(\mathbf{A}_{\mathtt{L}}^{(l,j-1)}\odot\mathbf{A}_{\mathtt{R}}^{(l,j-1)}\Big)\Big(\Big[\big(\mathbf{A}_{\mathtt{L}}^{(l,j-1)}\big)^{\top}\mathbf{A}_{\mathtt{L}}^{(l,j-1)}\Big]\\ \circledast\Big[\big(\mathbf{A}_{\mathtt{R}}^{(l,j-1)}\big)^{\top}\mathbf{A}_{\mathtt{R}}^{(l,j-1)}\Big]\Big)^{\#},\end{aligned}\tag{47a}$$

$$\begin{aligned}\mathbf{A}_{\mathtt{L}}^{(l,j)}=\Big(\big[\boldsymbol{\mathcal{R}}\big]_{(1)}\mathbf{Q}_{\mathtt{R}}^{(l,j-1)}+\rho_{\mathtt{0}}\mathbf{T}_{\mathtt{R}}^{(l,j-1)}+\rho_{\mathtt{A}}\big(\mathbf{A}_{\mathtt{R}}^{(l,j-1)}-\mathbf{E}^{(l,j-1)}\big)\Big)\\ \Big(\mathbf{H}_{\mathtt{R}}^{(l,j-1)}+\rho_{\mathtt{A}}\mathbf{I}_R\Big)^{-1},\end{aligned}\tag{47b}$$

$$\begin{aligned}\mathbf{A}_{\mathtt{R}}^{(l,j)}=\Big(\big[\boldsymbol{\mathcal{R}}\big]_{(2)}\mathbf{Q}_{\mathtt{L}}^{(l,j-1)}+\rho_{\mathtt{0}}\mathbf{T}_{\mathtt{L}}^{(l,j-1)}+\rho_{\mathtt{A}}\big(\mathbf{A}_{\mathtt{L}}^{(l,j-1)}-\mathbf{E}^{(l,j-1)}\big)\Big)\\ \Big(\mathbf{H}_{\mathtt{L}}^{(l,j-1)}+\rho_{\mathtt{A}}\mathbf{I}_R\Big)^{-1},\end{aligned}\tag{47c}$$

$$\mathbf{E}^{(l,j)}=\mathbf{E}^{(l,j-1)}+\mathbf{A}_{\mathtt{L}}^{(l,j)}-\mathbf{A}_{\mathtt{R}}^{(l,j)}.\tag{47d}$$

where $\rho_{\mathtt{A}}$ and $\mathbf{E}$ play the same role as $\rho_{\mathtt{B}}$ and $\mathbf{D}$ in (41), respectively, and

$$\mathbf{H}_{\mathtt{L}}^{(l,j-1)} = \big(\mathbf{Q}_{\mathtt{L}}^{(l,j-1)}\big)^{\top}\mathbf{Q}_{\mathtt{L}}^{(l,j-1)} + \rho_{\mathtt{0}}\big(\mathbf{G}_{\mathtt{L}}^{(l,j-1)}\big)^{\top}\mathbf{G}_{\mathtt{L}}^{(l,j-1)}, \tag{48}$$

$$\mathbf{H}_{\mathtt{R}}^{(l,j-1)} = \big(\mathbf{Q}_{\mathtt{R}}^{(l,j-1)}\big)^{\top}\mathbf{Q}_{\mathtt{R}}^{(l,j-1)} + \rho_{\mathtt{0}}\big(\mathbf{G}_{\mathtt{R}}^{(l,j-1)}\big)^{\top}\mathbf{G}_{\mathtt{R}}^{(l,j-1)}. \tag{49}$$

Once (47) meets the stopping criteria, we estimate $\mathbf{\Lambda}_{\mathbf{A}}^{(l)}$ using a way similar to (43), and then, set $\mathbf{\Sigma}^{(l)} = \mathbf{\Sigma}^{(l,J_{\text{stop}})}$ and $\mathbf{A}^{(l)} = 0.5\big(\mathbf{A}_{\mathtt{L}}^{(l,J_{\text{stop}})} + \mathbf{A}_{\mathtt{R}}^{(l,J_{\text{stop}})}\mathbf{\Lambda}_{\mathbf{A}}^{(l)}\big)$.

3.3. Stopping Criteria and Parameter Selection

Our method consists of an outer loop, and two inner loops (42) and (47). We set their maximum number of iterations to predefined values: $L_{\text{stop}} = 100$, $I_{\text{stop}} = 10$, and $J_{\text{stop}} = 10$, respectively. Following the guidelines in [39], we adopt the following stopping criteria, which rely on the *primal* and *dual* residuals

$$\big\|\mathbf{V}_{\text{cur}} - \mathbf{Z}_{\text{cur}}\big\|_F \le \varepsilon_{\text{pri}}, \quad \big\|\rho(\mathbf{Z}_{\text{cur}} - \mathbf{Z}_{\text{old}})\big\|_F \le \varepsilon_{\text{dual}}, \tag{50}$$

where "cur" and "old" represent the current and old estimates, respectively, and $\varepsilon_{\text{pri}} = \epsilon_{\text{rel}} \max\big\{\|\mathbf{V}_{\text{cur}}\|_2, \|\mathbf{Z}_{\text{cur}}\|_2\big\} + \epsilon_{\text{abs}}\sqrt{\text{length}(\mathbf{V}_{\text{cur}})}$ and $\varepsilon_{\text{dual}} = \epsilon_{\text{rel}}\|\rho\mathbf{S}_{\text{cur}}\|_2 + \epsilon_{\text{abs}}\sqrt{\text{length}(\mathbf{S}_{\text{cur}})}$. Here, $\epsilon_{\text{abs}} > 0$ and $\epsilon_{\text{rel}} > 0$ represent the absolute and relative tolerance, respectively. The primal variables are denoted by $\mathbf{V}$ and $\mathbf{Z}$, where $\mathbf{V}$ includes $(\mathbf{B}, \mathbf{B}_{\mathtt{L}}, \mathbf{A}_{\mathtt{L}})$, and $\mathbf{Z}$ includes $(\mathbf{A}_{\odot 2}, \mathbf{B}_{\mathtt{R}}, \mathbf{A}_{\mathtt{R}})$. The dual variables $(\mathbf{U}, \mathbf{D}, \mathbf{E})$ are represented by $\mathbf{S}$, while ρ represents the regularization parameters $(\rho_{\mathtt{0}}, \rho_{\mathtt{B}}, \rho_{\mathtt{A}})$ and their value can be selected by applying the following adaptive rule at each iteration

$$\rho^{(\ell+1)} = \begin{cases} \kappa\ \rho^{(\ell)} & \text{if } \big\|\mathbf{V}_{\text{cur}} - \mathbf{Z}_{\text{cur}}\big\|_F > \mu\big\|\rho^{(\ell)}(\mathbf{Z}_{\text{cur}} - \mathbf{Z}_{\text{old}})\big\|_F, \\ \rho^{(\ell)}/\kappa & \text{if } \big\|\rho^{(\ell)}(\mathbf{Z}_{\text{cur}} - \mathbf{Z}_{\text{old}})\big\|_F > \mu\big\|\mathbf{V}_{\text{cur}} - \mathbf{Z}_{\text{cur}}\big\|_F, \\ \rho^{(\ell)} & \text{otherwise.} \end{cases} \tag{51}$$

In practice, typical choices can be $\mu = 10$, $\kappa = 2$, and $\rho^{(0)} = \rho_{\mathtt{0}}^{(0)} = \rho_{\mathtt{B}}^{(0)} = \rho_{\mathtt{A}}^{(0)} = 1$, while ϵ_{abs} and ϵ_{rel} can be chosen from the ranges $[10^{-6}; 10^{-3}]$ and $[10^{-4}; 10^{-2}]$, respectively. For further details, please refer to [39].

3.4. Performance Analysis

3.4.1. Computational Complexity

For short, we assume $N_1 = N_2 = N$. TenSOFO involves two inner ADMM loops, denoted as (42) and (47). In loop (42), the computation includes a pseudo-inverse and two inverse operations of $R \times R$ matrices, resulting in a cost of $\mathcal{O}(R^3)$ flops. Additionally, the Khatri-Rao products requires a cost of $\mathcal{O}(\max\{M^2, N\}M^2R)$ flops. Consequently, each iteration incurs a total cost of $\mathcal{O}(M^4R^2N)$ flops for updating $\mathbf{K}^{(l,i)}$, $\mathbf{B}_{\mathtt{L}}^{(l,i)}$, $\mathbf{B}_{\mathtt{R}}^{(l,i)}$, and $\mathbf{D}^{(l,i)}$. Thus, the computational complexity of (42) is $\mathcal{O}(I_{\text{stop}}M^4R^2N)$ flops. At the end of the loop (42), TenSOFO also involves the scaling step that requires $\mathcal{O}(M^2R)$ flops. Loop (47) shares a similar update rule with (42) but deals with $\boldsymbol{\mathcal{R}} \in \mathbb{R}^{M\times M\times N}$ of smaller size, leading to a complexity of $\mathcal{O}(J_{\text{stop}}M^2R^2N)$ flops. To sum up, the overall computational complexity of TenSOFO is $\mathcal{O}\big(L_{\text{stop}}\ (I_{\text{stop}}M^2 + J_{\text{stop}})M^2R^2N\big)$ flops.

3.4.2. Convergence Analysis

The convergence behavior of TenSOFO is summarized in the following theorem.

Theorem 1. *Assume that covariance matrices and quadricovariance tensors of data observations are bounded in Frobenius norm. If $\rho_{\mathit{0}} > 1$ and $\rho_{\mathit{A}}, \rho_{\mathit{B}} > 0$, the sequence $\{\mathbf{B}^{(l)}, \mathbf{K}^{(l)}, \mathbf{A}^{(l)}, \mathbf{\Sigma}^{(l)}, \mathbf{U}^{(l)}\}$ generated by TenSOFO in Algorithm 1 converges to a stationary point of $\mathcal{L}_{\mathit{0}}(\cdot)$ when l goes to infinity.*

Proof. See Appendix A. □

4. Convolutive BSS: Type-2 BTD based Method

In this section, we begin by establishing a connection between convolutive BSS and a constrained type-2 block term decomposition (BTD) factorization. Subsequently, we introduce an efficient method called TCBSS, designed for dealing with this specific BTD, and, by extension, convolutive BSS tasks.

4.1. Link between Convolutive BSS and Type-2 BTD

As suggested in [3, 40], we can reformulate the convolutive data model (1) into an instantaneous one as follows

$$\underline{\mathbf{x}}[t] = \underline{\mathbf{A}}\underline{\mathbf{s}}[t], \tag{52}$$

where $\underline{\mathbf{x}}[t] = \big[x_1[t], \ldots, x_1[t-L'+1], \ldots, x_M[t], \ldots, x_M[t-L'+1]\big]^\top \in \mathbb{R}^{ML'}$ and $\underline{\mathbf{s}}[t] = \big[s_1[t], \ldots, s_1[t-(L+L')+1], \ldots, s_R[t], \ldots, s_R[t-(L+L')+1]\big]^\top \in \mathbb{R}^{R(L+L')}$ with an extension factor for data observations $L' \geq 1$, and the new mixture $\underline{\mathbf{A}} \in \mathbb{R}^{ML' \times R(L+L')}$ is given by

$$\underline{\mathbf{A}} = \begin{bmatrix} \underline{\mathbf{A}}_{11} & \cdots & \underline{\mathbf{A}}_{1R} \\ \vdots & \ddots & \vdots \\ \underline{\mathbf{A}}_{M1} & \cdots & \underline{\mathbf{A}}_{MR} \end{bmatrix} \text{ with } \underline{\mathbf{A}}_{mr} = \begin{bmatrix} a_{mr}[0] & \ldots & a_{mr}[L] & \ldots & 0 \\ & \ddots & \ddots & \ddots & \\ 0 & & a_{mr}[0] & \ldots & a_{mr}[L] \end{bmatrix}. \tag{53}$$

For short, we denote $L_{\text{add}} = L + L'$.

We assume that individual source signals are temporally coherent while maintaining their mutual independence. Specifically, the correlation between the two sources s_i and s_j (with $i \neq j$) follows $\mathbb{E}\big\{s_i[t]s_j[t-\tau]\big\} = 0$ for all τ. In such cases, the correlation matrix corresponding to (52) is expressed as

$$\mathbf{R}_{\underline{\mathbf{x}}}[t,\tau] = \mathbb{E}\Big\{\underline{\mathbf{x}}[t]\underline{\mathbf{x}}[t-\tau]^\top\Big\} = \underline{\mathbf{A}}\,\text{blkdiag}\Big(\big\{\mathbf{R}_{\underline{\mathbf{s}}_r}[t,\tau]\big\}_{r=1}^{R}\Big)\underline{\mathbf{A}}^\top. \tag{54}$$

For multiple time lags $\{\tau_i\}_{i=1}^N$, we obtain

$$\begin{cases} \mathbf{R}_{\underline{\mathbf{x}}}[t,\tau_1] &= \underline{\mathbf{A}}\,\text{blkdiag}\Big(\big\{\mathbf{R}_{\underline{\mathbf{s}}_r}[t,\tau_1]\big\}_{r=1}^{R}\Big)\underline{\mathbf{A}}^\top, \\ \mathbf{R}_{\underline{\mathbf{x}}}[t,\tau_2] &= \underline{\mathbf{A}}\,\text{blkdiag}\Big(\big\{\mathbf{R}_{\underline{\mathbf{s}}_r}[t,\tau_2]\big\}_{r=1}^{R}\Big)\underline{\mathbf{A}}^\top, \\ & \vdots \\ \mathbf{R}_{\underline{\mathbf{x}}}[t,\tau_N] &= \underline{\mathbf{A}}\,\text{blkdiag}\Big(\big\{\mathbf{R}_{\underline{\mathbf{s}}_r}[t,\tau_N]\big\}_{r=1}^{R}\Big)\underline{\mathbf{A}}^\top. \end{cases} \tag{55}$$

Particularly by stacking $\big\{\mathbf{R}_{\underline{\mathbf{x}}}[t,\tau_i]\big\}_{i=1}^N$ and $\big\{\mathbf{R}_{\underline{\mathbf{s}}}[t,\tau_i]\big\}_{i=1}^N$ consecutively within the third mode

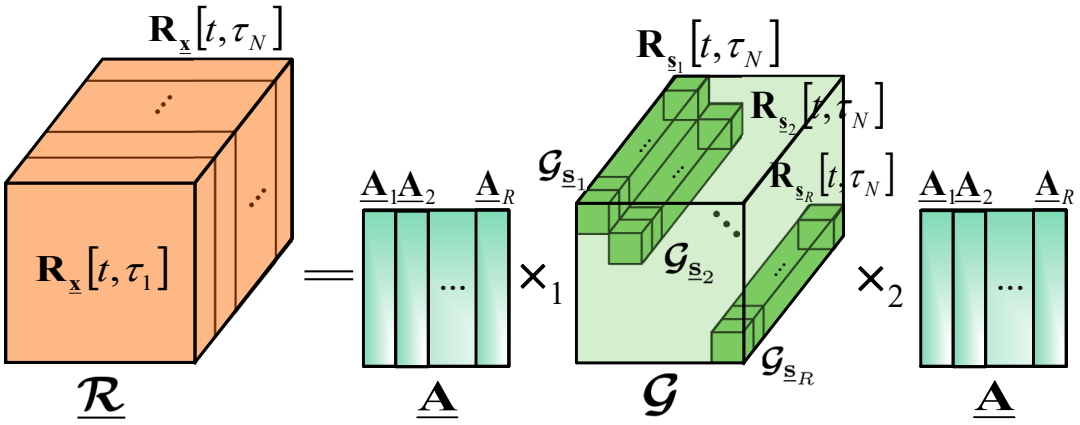


Figure 2: TCBSS – Constrained Type-2 BTD based Convolutive BSS. By extracting second-order (SO) statistics of observations and source signals, we obtain sets of covariance matrices $\{\mathbf{R}_{\underline{\mathbf{x}}}[t,\tau_n]\}_{n=1}^{N}$ and $\{\mathbf{R}_{\underline{\mathbf{s}}_r}[t,\tau_n]\}_{n,r=1}^{N,R}$. These matrices are then modeled within a constrained type-2 block term decomposition framework, enforcing two tensor factors to be identical. By performing the BTD decomposition of $\underline{\mathcal{R}}$, we derive the mixing matrix $\underline{\mathbf{A}}$.

of $\underline{\mathcal{R}} \in \mathbb{R}^{ML' \times ML' \times K}$ and $\mathcal{G} \in \mathbb{R}^{RL_{\text{add}} \times RL_{\text{add}} \times N}$, we derive the following representation

$$\underline{\mathcal{R}} = \mathcal{G} \times_1 \underline{\mathbf{A}} \times_2 \underline{\mathbf{A}} = \sum_{r=1}^{R} \mathcal{G}_{\underline{\mathbf{s}}_r} \times_1 \underline{\mathbf{A}}_r \times_2 \underline{\mathbf{A}}_r, \tag{56}$$

where $\underline{\mathbf{A}}_r = \left[\underline{\mathbf{A}}_{1r}^{\top}, \underline{\mathbf{A}}_{2r}^{\top}, \ldots, \underline{\mathbf{A}}_{Mr}^{\top}\right]^{\top}$ is the r-th block column of $\underline{\mathbf{A}}$ and $\mathcal{G}_{\underline{\mathbf{s}}_r}(:,:,i) = \mathbf{R}_{\underline{\mathbf{s}}_r}(t,\tau_i)$. See Fig. 2 for an illustration. We can see that (56) represents a variant of the type-2 BTD in (9), where the two loading factors are constrained to be identical. Interestingly, (56) is also essentially unique under mild conditions [37], as highlighted by Proposition 3. Particularly, the estimate $\hat{\mathbf{A}}$ of $\underline{\mathbf{A}}$ is unique up to trivial indeterminacies, i.e., $\hat{\mathbf{A}} = \underline{\mathbf{A}}\mathbf{\Pi}\mathbf{\Lambda}$ where $\mathbf{\Pi}$ is a block permutation matrix and $\mathbf{\Lambda}$ is a square nonsingular block-diagonal matrix.

Proposition 3. *If $ML' \geq RL_{\text{add}}$, $N \geq 3$, and $\mathcal{G}$ is generic,*[3] *then* (56) *is essentially unique.*

This result is a corollary of Theorem 6.1 in [37]. By employing the constrained type-2 BTD analysis, we can directly identify the mixing process in convolutive BSS. In next subsection, we introduce an effective optimization framework designed to deal with (56) efficiently.

[3]We call a tensor generic when its elements are drawn from a continuous probability density function.

Algorithm 2: TCBSS

Input: $\left\{\mathbf{R}_{\underline{\mathbf{x}}}[t,\tau_i]\right\}_{i=1}^{N}$
Output: Loading factor $\underline{\mathbf{A}}$ and core tensor $\boldsymbol{\mathcal{G}}$
Initialization:

- set $l=1, \alpha^{(0)}=1$, $\underline{\mathbf{U}}^{(0)}=\mathbf{0}_{ML'\times RL_{\text{add}}}$, and $\mathbf{Z}^{(0)}=\mathbf{0}_{ML'\times RL_{\text{add}}}$
- generate random matrices $\underline{\mathbf{B}}^{(0)}\in\mathbb{R}^{ML'\times RL_{\text{add}}}$ and $\boldsymbol{\mathcal{G}}_{\underline{\mathbf{s}}_r}^{(0)}\in\mathbb{R}^{L_{\text{add}}\times L_{\text{add}}\times N}$

while *stopping criteria are not met* **do**

$$\underline{\mathbf{A}}^{(l)}=\underset{\underline{\mathbf{A}}}{\operatorname{argmin}}\left\{\left\|\underline{\boldsymbol{\mathcal{R}}}-\sum_{r=1}^{R}\boldsymbol{\mathcal{G}}_{\underline{\mathbf{s}}_r}^{(l-1)}\times_1\underline{\mathbf{A}}_r\times_2\underline{\mathbf{B}}_r^{(l-1)}\right\|_F^2+\frac{\rho}{2}\left\|\underline{\mathbf{B}}^{(l-1)}-\underline{\mathbf{A}}+\underline{\mathbf{U}}^{(l-1)}\right\|_F^2\right\} \tag{P2.1}$$

$$\underline{\mathbf{B}}^{(l)}=\underset{\underline{\mathbf{B}}}{\operatorname{argmin}}\left\{\left\|\underline{\boldsymbol{\mathcal{R}}}-\sum_{r=1}^{R}\boldsymbol{\mathcal{G}}_{\underline{\mathbf{s}}_r}^{(l-1)}\times_1\underline{\mathbf{A}}_r^{(l)}\times_2\underline{\mathbf{B}}_r\right\|_F^2+\frac{\rho}{2}\left\|\underline{\mathbf{B}}-\underline{\mathbf{A}}^{(l)}+\underline{\mathbf{U}}^{(l-1)}\right\|_F^2\right\} \tag{P2.2}$$

$$\boldsymbol{\mathcal{G}}^{(l)}=\underset{\boldsymbol{\mathcal{G}}}{\operatorname{argmin}}\left\|\underline{\boldsymbol{\mathcal{R}}}-\sum_{r=1}^{R}\boldsymbol{\mathcal{G}}_{\underline{\mathbf{s}}_r}\times_1\underline{\mathbf{A}}_r^{(l)}\times_2\underline{\mathbf{B}}_r^{(l)}\right\|_F^2 \tag{P2.3}$$

$$\underline{\mathbf{U}}^{(l)}=\mathbf{Z}^{(l-1)}+\underline{\mathbf{B}}^{(l)}-\underline{\mathbf{A}}^{(l)} \tag{P2.4}$$

$$\alpha^{(l)}=\frac{1+\sqrt{1+4(\alpha^{(l-1)})^2}}{2} \tag{P2.5}$$

$$\mathbf{Z}^{(l)}=\underline{\mathbf{U}}^{(l)}+\frac{\alpha^{(l-1)}-1}{\alpha^{(l)}}\left(\underline{\mathbf{U}}^{(l)}-\underline{\mathbf{U}}^{(l-1)}\right) \tag{P2.6}$$

$l=l+1$

end

4.2. Optimization Framework

The type-2 BTD factorization (56) can be represented by

$$\min_{\boldsymbol{\mathcal{G}},\underline{\mathbf{A}},\underline{\mathbf{B}}}\left\|\underline{\boldsymbol{\mathcal{R}}}-\sum_{r=1}^{R}\boldsymbol{\mathcal{G}}_{\underline{\mathbf{s}}_r}\times_1\underline{\mathbf{A}}_r\times_2\underline{\mathbf{B}}_r\right\|_F^2\quad\text{subject to }\underline{\mathbf{A}}=\underline{\mathbf{B}}, \tag{58}$$

where $\underline{\mathbf{A}}=[\underline{\mathbf{A}}_1,\underline{\mathbf{A}}_2,\ldots,\underline{\mathbf{A}}_R]$ and $\underline{\mathbf{B}}=[\underline{\mathbf{B}}_1,\underline{\mathbf{B}}_2,\ldots,\underline{\mathbf{B}}_R]$. Here, we can formulate the augmented Lagrangian function for the minimization (58) as follows

$$\mathcal{L}(\boldsymbol{\mathcal{G}},\underline{\mathbf{A}},\underline{\mathbf{B}},\underline{\mathbf{U}})=\min_{\boldsymbol{\mathcal{G}},\underline{\mathbf{A}},\underline{\mathbf{B}}}\left\|\underline{\boldsymbol{\mathcal{R}}}-\sum_{r=1}^{R}\boldsymbol{\mathcal{G}}_{\underline{\mathbf{s}}_r}\times_1\underline{\mathbf{A}}_r\times_2\underline{\mathbf{B}}_r\right\|_F^2+\frac{\rho}{2}\left\|\underline{\mathbf{B}}-\underline{\mathbf{A}}+\underline{\mathbf{U}}\right\|_F^2-\frac{\rho}{2}\left\|\underline{\mathbf{U}}\right\|_F^2, \tag{59}$$

where $\underline{\mathbf{U}}$ is the "scaled" dual variable and $\rho>0$ is a regularization parameter. In this work, we propose Algorithm 2 to find the stationary point of $\mathcal{L}(\cdot)$. As the three sub-problems (P2.1), (P2.2) and (P2.3) are quadratic, their closed-form solutions $\underline{\mathbf{A}}^{(l)}$, $\underline{\mathbf{B}}^{(l)}$, and $\boldsymbol{\mathcal{G}}^{(l)}$ at

the l-th iteration are determined as

$$\underline{\mathbf{A}}^{(l)} = \Big(\big[\boldsymbol{\mathcal{R}}\big]_{(1)}\big(\mathbf{W}_A^{(l)}\big)^\top + \rho\big(\underline{\mathbf{B}}^{(l-1)} + \underline{\mathbf{U}}^{(l-1)}\big)\Big)\Big(\mathbf{W}_A^{(l)}\big(\mathbf{W}_A^{(l)}\big)^\top + \rho\mathbf{I}_{RL_{\text{add}}}\Big)^{-1}, \tag{60}$$

$$\underline{\mathbf{B}}^{(l)} = \Big(\big[\boldsymbol{\mathcal{R}}\big]_{(2)}\big(\mathbf{W}_B^{(l)}\big)^\top + \rho\big(\underline{\mathbf{U}}^{(l-1)} - \underline{\mathbf{A}}^{(l)}\big)\Big)\Big(\mathbf{W}_B^{(l)}\big(\mathbf{W}_B^{(l)}\big)^\top + \rho\mathbf{I}_{RL_{\text{add}}}\Big)^{-1}, \tag{61}$$

$$\Big[\big[\boldsymbol{\mathcal{G}}_{\underline{\mathbf{s}}_1}^{(l)}\big]_{(3)}, \big[\boldsymbol{\mathcal{G}}_{\underline{\mathbf{s}}_2}^{(l)}\big]_{(3)}, \ldots, \big[\boldsymbol{\mathcal{G}}_{\underline{\mathbf{s}}_R}^{(l)}\big]_{(3)}\Big] = \big[\boldsymbol{\mathcal{R}}\big]_{(3)}\Big(\big(\underline{\mathbf{B}}^{(l)} \boxtimes \underline{\mathbf{A}}^{(l)}\big)^\top\Big)^{\#}, \tag{62}$$

where $\mathbf{W}_A^{(l)}$ and $\mathbf{W}_B^{(l)}$ are given by

$$\mathbf{W}_A^{(l)} = \Big[\big[\boldsymbol{\mathcal{G}}_{\underline{\mathbf{s}}_1}^{(l-1)} \times_2 \underline{\mathbf{B}}_1^{(l-1)}\big]_{(1)}^\top, \ldots, \big[\boldsymbol{\mathcal{G}}_{\underline{\mathbf{s}}_R}^{(l-1)} \times_2 \underline{\mathbf{B}}_R^{(l-1)}\big]_{(1)}^\top\Big]^\top, \tag{63}$$

$$\mathbf{W}_B^{(l)} = \Big[\big[\boldsymbol{\mathcal{G}}_{\underline{\mathbf{s}}_1}^{(l-1)} \times_1 \underline{\mathbf{A}}_1^{(l)}\big]_{(2)}^\top, \ldots, \big[\boldsymbol{\mathcal{G}}_{\underline{\mathbf{s}}_R}^{(l-1)} \times_1 \underline{\mathbf{A}}_R^{(l)}\big]_{(2)}^\top\Big]^\top. \tag{64}$$

Inspired by the work [42], we introduce two auxiliary variables $\mathbf{Z}^{(l)}$ and α_l to accelerate the iteration procedure. The inclusion of (P2.4), (P2.5), and (P2.6) results in an accelerated augmented Lagrangian variant which can offer superior speed and estimation accuracy, as compared with state-of-the-art methods, please see Fig. 8 for an example.

4.3. Stopping Criteria and Parameter Selection

Our procedure stops upon convergence, reaching the maximum number of iterations $I_{\text{stop}} = 100$, or meeting the following criteria

$$\big\|\underline{\mathbf{A}}^{(l)} - \underline{\mathbf{B}}^{(l)}\big\|_F \le \varepsilon_{\text{pri}} \text{ and } \big\|\rho\big(\underline{\mathbf{A}}^{(l)} - \underline{\mathbf{A}}^{(l-1)}\big)\big\|_F \le \varepsilon_{\text{dual}}, \tag{65}$$

where $\varepsilon_{\text{pri}} = \epsilon_{\text{rel}} \max\big\{\|\underline{\mathbf{A}}^{(l)}\|_2, \|\underline{\mathbf{B}}^{(l)}\|_2\big\} + \epsilon_{\text{abs}}\sqrt{\text{length}(\mathbf{A}^{(l)})}$ and $\varepsilon_{\text{dual}} = \epsilon_{\text{rel}}\|\rho\underline{\mathbf{U}}^{(l)}\|_2 + \epsilon_{\text{abs}}\sqrt{\text{length}(\mathbf{A}^{(l)})}$. Here, $\epsilon_{\text{abs}} > 0$ and $\epsilon_{\text{rel}} > 0$ represent the absolute and relative tolerance, respectively. In practice, we can set $\rho = 1, \epsilon_{\text{abs}} = 10^{-4}$ and $\epsilon_{\text{rel}} = 10^{-2}$ for reasonable performance. To speed up the convergence rate, an adaptive parameter selection for ρ can be employed, as described in (51).

4.4. Performance Analysis

4.4.1. Computational Complexity

At each iteration, the updates for both $\underline{\mathbf{A}}^{(l)}$ in (60) and $\underline{\mathbf{B}}^{(l)}$ in (61) share the same computational process. This requires $\mathcal{O}\big(M^2L'^2L_{\text{add}}RK + ML'L_{\text{add}}^2R^2K + L_{\text{add}}^3R^3\big)$ flops. For computing $\boldsymbol{\mathcal{G}}^{(l)}$ defined in (62), the complexity is $\mathcal{O}\big(M^2L'^2L_{\text{add}}^2\ RK + M^2L'^2RL_{\text{add}}^2 \min(M^2L'^2, RL_{\text{add}}^2)\big)$ flops. The computations required for updating $\underline{\mathbf{U}}^{(l)}$, $\mathbf{Z}^{(l)}$, and α_l are relatively inexpensive, with respective complexities of $\mathcal{O}(ML'RL_{\text{add}})$, $\mathcal{O}(ML'RL_{\text{add}})$, and $\mathcal{O}(1)$. As mentioned in Proposition 3, satisfying the uniqueness condition demands $ML' \geq RL_{\text{add}}$. Accordingly, the overall complexity of TCBSS is $\mathcal{O}\big(M^2L'^2\ L_{\text{add}}^2R\max(K, L_{\text{add}}^2R)+ L_{\text{add}}^3R^3\big)$ flops at each iteration.

4.4.2. Convergence Analysis

The convergence behavior of TCBSS is summarized in the following theorem.

Theorem 2. *Assume that covariance matrices of data observations are bounded in Frobenius norm. If $\rho > 1$, the sequence $\big\{\underline{\boldsymbol{\mathcal{G}}}^{(l)}, \underline{\mathbf{A}}^{(l)}, \underline{\mathbf{B}}^{(l)}, \underline{\mathbf{U}}^{(l)}\big\}$ generated by TCBSS in Algorithm 2 converges to a stationary point of $\mathcal{L}_0(\cdot)$ when l goes to infinity.*

Proof. Proof of Theorem 2 can be obtained by applying the same mathematical framework as presented in Sec. 3.4.2 for analyzing the convergence behavior of TenSOFO. □

4.5. Extension

In this subsection, we introduce a variant of TCBSS that integrates both SO and FO statistics.

We begin by computing the FO statistic of $\underline{\mathbf{x}}[t]$ in (52) as follows

$$\mathbf{C}_{\underline{\mathbf{x}}}[t, \{\boldsymbol{\tau}\}] = \big(\underline{\mathbf{A}} \odot \underline{\mathbf{A}}\big)\,\text{blkdiag}\left(\left\{\mathbf{C}_{\underline{\mathbf{s}}_r}[t, \{\boldsymbol{\tau}\}]\right\}_{r=1}^{R}\right)\big(\underline{\mathbf{A}} \odot \underline{\mathbf{A}}\big)^{\mathsf{T}}. \tag{66}$$

By incorporating multiple time lags $\{\tau_i\}_{i=1}^{N_2}$, we derive a set of FO matrices $\{\mathbf{C}_{\underline{\mathbf{x}}}[t, \{\boldsymbol{\tau}_n\}]\}_{n=1}^{N_2}$. These matrices are then stacked into a third-order tensor $\underline{\boldsymbol{\mathcal{C}}}$ in the following manner

$$\underline{\boldsymbol{\mathcal{C}}} = \mathbf{C}_{\underline{\mathbf{x}}}[t, \{\boldsymbol{\tau}_1\}] \boxplus \mathbf{C}_{\underline{\mathbf{x}}}[t, \{\boldsymbol{\tau}_2\}] \cdots \boxplus \mathbf{C}_{\underline{\mathbf{x}}}[t, \{\boldsymbol{\tau}_{N_2}\}]. \tag{67}$$

Similar to the tensor $\underline{\boldsymbol{\mathcal{R}}}$ in (56), $\underline{\boldsymbol{\mathcal{C}}}$ also admits the type-2 BTD factorization

$$\underline{\boldsymbol{\mathcal{C}}} = \boldsymbol{\mathcal{Z}} \times_1 \underline{\mathbf{A}}_{\odot 2} \times_2 \underline{\mathbf{A}}_{\odot 2} = \sum_{r=1}^{R} \boldsymbol{\mathcal{Z}}_{\mathbf{s}_r} \times_1 (\underline{\mathbf{A}}_r \odot \underline{\mathbf{A}}_r) \times_2 (\underline{\mathbf{A}}_r \odot \underline{\mathbf{A}}_r), \tag{68}$$

where $\boldsymbol{\mathcal{Z}} = \text{blkdiag}(\boldsymbol{\mathcal{Z}}_{\underline{\mathbf{s}}_1}, \boldsymbol{\mathcal{Z}}_{\underline{\mathbf{s}}_2}, \ldots, \boldsymbol{\mathcal{Z}}_{\underline{\mathbf{s}}_R})$ with $\boldsymbol{\mathcal{Z}}_{\underline{\mathbf{s}}_r}(:,:,n) = \mathbf{C}_{\underline{\mathbf{s}}_r}[t, \{\boldsymbol{\tau}_n\}]$, and $\underline{\mathbf{A}}_r$ is the r-th block column of $\underline{\mathbf{A}}$. Accordingly, we can represent the convolutive BSS problem in the lens of a joint (coupled) type-2 BTD decomposition

$$\underline{\boldsymbol{\mathcal{R}}} = \boldsymbol{\mathcal{G}} \times_1 \underline{\mathbf{A}} \times_2 \underline{\mathbf{A}} \quad \text{and} \quad \underline{\boldsymbol{\mathcal{C}}} = \boldsymbol{\mathcal{Z}} \times_1 \underline{\mathbf{A}}_{\odot 2} \times_2 \underline{\mathbf{A}}_{\odot 2}, \tag{69}$$

where $\underline{\mathbf{A}}_{\odot 2} = \underline{\mathbf{A}} \odot \underline{\mathbf{A}}$, and thus, formulate the following ADMM optimization

$$\begin{aligned} &\min_{\boldsymbol{\mathcal{G}}, \underline{\mathbf{A}}, \boldsymbol{\mathcal{Z}}, \underline{\mathbf{B}}} \left\{ \left\| \underline{\boldsymbol{\mathcal{R}}} - \boldsymbol{\mathcal{G}} \times_1 \underline{\mathbf{A}} \times_2 \underline{\mathbf{A}} \right\|_F^2 + \left\| \underline{\boldsymbol{\mathcal{C}}} - \boldsymbol{\mathcal{Z}} \times_1 \underline{\mathbf{D}} \times_2 \underline{\mathbf{D}} \right\|_F^2 \right\} \\ &\text{subject to } \underline{\mathbf{D}} = \underline{\mathbf{A}}_{\odot 2}. \end{aligned} \tag{70}$$

In this work, we can apply the same optimization framework of TenSOFO in Algorithm 1 to solve (70). Particularly at the l-th iteration, we update parameters of interest as follows

$$\left\{ \boldsymbol{\mathcal{Z}}^{(l)}, \underline{\mathbf{D}}^{(l)} \right\} = \underset{\boldsymbol{\mathcal{Z}}, \underline{\mathbf{B}}}{\text{argmin}} \left\{ \left\| \underline{\boldsymbol{\mathcal{C}}} - \boldsymbol{\mathcal{Z}} \times_1 \underline{\mathbf{D}} \times_2 \underline{\mathbf{D}} \right\|_F^2 + \frac{\gamma}{2} \left\| \underline{\mathbf{D}} - \underline{\mathbf{A}}_{\odot 2}^{(l-1)} + \underline{\mathbf{Q}}^{(l-1)} \right\|_F^2 \right\}, \tag{71}$$

$$\left\{ \boldsymbol{\mathcal{G}}^{(l)}, \underline{\mathbf{A}}^{(l)} \right\} = \underset{\boldsymbol{\mathcal{G}}, \underline{\mathbf{A}}}{\text{argmin}} \left\{ \left\| \underline{\boldsymbol{\mathcal{R}}} - \boldsymbol{\mathcal{G}} \times_1 \underline{\mathbf{A}} \times_2 \underline{\mathbf{A}} \right\|_F^2 + \frac{\gamma}{2} \left\| \underline{\mathbf{D}}^{(l)} - \underline{\mathbf{A}}_{\odot 2} + \underline{\mathbf{Q}}^{(l-1)} \right\|_F^2 \right\}, \tag{72}$$

$$\underline{\mathbf{Q}}^{(l)} = \underline{\mathbf{Q}}^{(l-1)} + \underline{\mathbf{D}}^{(l)} - \underline{\mathbf{A}}_{\odot 2}^{(l)}, \tag{73}$$

where $\underline{\mathbf{Q}}$ is the (scaled) dual variable and γ is a regularization parameter. Sub-problems (71) and (72) can be effectively addressed by incorporating a simple regularization term $\|\cdot\|_F^2$ into the original version of TCBSS presented in Algorithm 2. For example, in the case of sub-problem (71), the closed-form solution of $\boldsymbol{\mathcal{Z}}$ keeps the same form as $\boldsymbol{\mathcal{G}}$ in (62) during each iteration. Meanwhile, the solution for $\underline{\mathbf{D}}$ is adjusted from (60) and takes the following

form

$$\underline{\mathbf{D}}_{\mathtt{L}}^{(l,i)} = \Big(\big[\underline{\boldsymbol{\mathcal{C}}}\big]_{(1)}\big(\mathbf{W}_{\mathtt{L}}^{(l)}\big)^{\top} + \rho\big(\underline{\mathbf{D}}_{\mathtt{R}}^{(l,i-1)} + \underline{\mathbf{P}}^{(l,i-1)}\big) + \gamma\big(\underline{\mathbf{A}}_{\odot 2}^{(l-1)} - \underline{\mathbf{Q}}^{(l-1)}\big)\Big) \Big(\mathbf{W}_{\mathtt{L}}^{(l)}\big(\mathbf{W}_{\mathtt{L}}^{(l)}\big)^{\top} + (\rho+\gamma)\mathbf{I}_{RL_{\text{add}}}\Big)^{-1}. \tag{74}$$

Here, the matrices $\underline{\mathbf{D}}_{\mathtt{L}}^{(l,i)}$, $\underline{\mathbf{D}}_{\mathtt{R}}^{(l,i-1)}$, $\mathbf{W}_{\mathtt{L}}^{(l,i)}$, and $\mathbf{W}_{\mathtt{R}}^{(l,i)}$ play the same role of $\underline{\mathbf{A}}^{(l)}$, $\underline{\mathbf{B}}^{(l-1)}$, $\mathbf{W}_{A}^{(l)}$, and $\mathbf{W}_{B}^{(l)}$ in Algorithm 2.

5. Numerical Experiments

5.1. Performance Assessment for TenSOFO

We evaluate the performance of TenSOFO in two aspects: (i) its effectiveness for joint INDSCAL decomposition, and (ii) its application to address BSS using SO and FO statistics.

5.1.1. INDSCAL Decomposition

We apply TenSOFO to compute joint INDSCAL decomposition of two symmetric tensors $\boldsymbol{\mathcal{R}} \in \mathbb{R}^{M\times M\times N}$ and $\boldsymbol{\mathcal{C}} \in \mathbb{R}^{M^2\times M^2\times N}$ sharing the same rank R:

$$\boldsymbol{\mathcal{R}} = \boldsymbol{\mathcal{R}}_{\text{true}} + \boldsymbol{\mathcal{N}}_{\boldsymbol{\mathcal{R}}} = \big[\!\big[\mathbf{A}, \mathbf{A}, \boldsymbol{\Sigma}\big]\!\big] + \boldsymbol{\mathcal{N}}_{\boldsymbol{\mathcal{R}}}, \tag{Tensor 1}$$

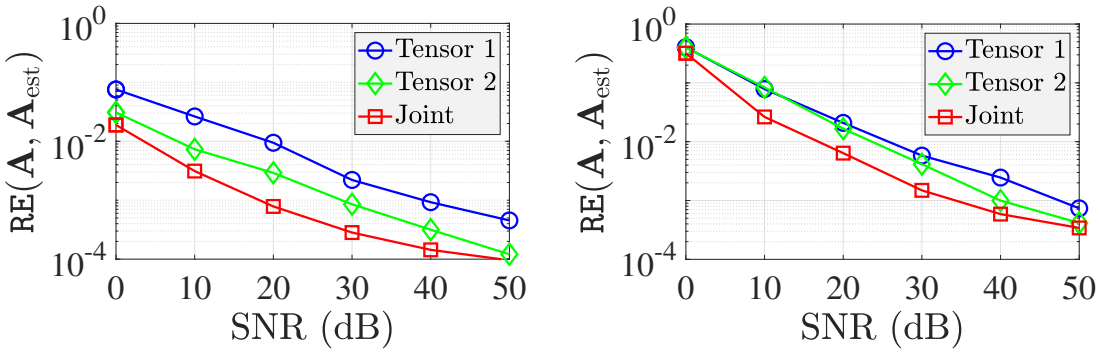
$$\boldsymbol{\mathcal{C}} = \boldsymbol{\mathcal{C}}_{\text{true}} + \boldsymbol{\mathcal{N}}_{\boldsymbol{\mathcal{C}}} = \big[\!\big[\mathbf{A}_{\odot 2}, \mathbf{A}_{\odot 2}, \mathbf{K}\big]\!\big] + \boldsymbol{\mathcal{N}}_{\boldsymbol{\mathcal{C}}}. \tag{Tensor 2}$$

Here, the tensor factors of interest $\mathbf{A} \in \mathbb{R}^{M\times R}, \boldsymbol{\Sigma} \in \mathbb{R}^{N\times R}$ and $\mathbf{K} \in \mathbb{R}^{N\times R}$ are generated as Gaussian matrices with zero-mean and unit-variance entries. $\boldsymbol{\mathcal{N}}_{\boldsymbol{\mathcal{R}}}$ and $\boldsymbol{\mathcal{N}}_{\boldsymbol{\mathcal{C}}}$ represent random Gaussian noises sharing the same SNR level, i.e., $\|\boldsymbol{\mathcal{N}}_{\boldsymbol{\mathcal{R}}}\|_F/\|\boldsymbol{\mathcal{R}}_{\text{true}}\|_F = \|\boldsymbol{\mathcal{N}}_{\boldsymbol{\mathcal{C}}}\|_F/\|\boldsymbol{\mathcal{C}}_{\text{true}}\|_F = 10^{-\frac{\text{SNR}}{20}}$. To evaluate the estimation accuracy, we measure the following relative error (RE) metric

$$\text{RE}(\mathbf{A}, \mathbf{A}_{\text{est}}) = \min_{\boldsymbol{\Pi},\boldsymbol{\Lambda}} \big\|\mathbf{A} - \mathbf{A}_{\text{est}}\boldsymbol{\Pi}\boldsymbol{\Lambda}\big\|_F \big/ \big\|\mathbf{A}\big\|_F, \tag{75}$$

where $\mathbf{A}_{\text{est}}$ refers to the estimate, $\boldsymbol{\Pi}$ and $\boldsymbol{\Lambda}$ represent the permutation and scaling matrices, respectively.

In this experiment, we set the values of M and N to 5 and 100, respectively. We consider two cases of the tensor rank R: one with $R = 3$ (less than M) and another with $R = 7$ (greater



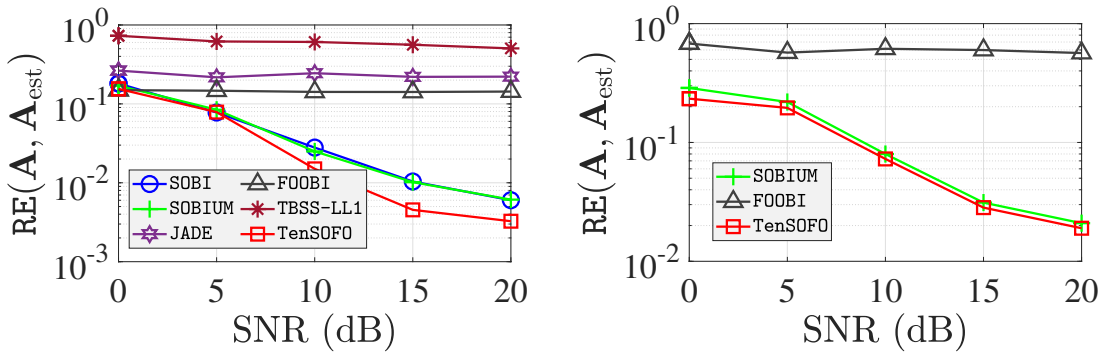
(a) Overdetermination: Rank $R = 3 < M$ (b) Underdetermination: Rank $R = 7 > M$

Figure 3: INDSCAL decomposition: data dimension $M = 5$, number of tensor slices $N = 100$, individual decomposition of $\boldsymbol{\mathcal{R}}$ (——), $\boldsymbol{\mathcal{C}}$ (——), and joint decomposition of both $\boldsymbol{\mathcal{R}}$ and $\boldsymbol{\mathcal{C}}$ (——).

than M). The performance of TenSOFO is conducted across varying SNR levels within the range $[0, 50]$ dB. Fig. 3 illustrates the performance comparison between TenSOFO and the classical INDSCAL method (i.e., CP-ALS) for each $\boldsymbol{\mathcal{R}}$ and $\boldsymbol{\mathcal{C}}$. The experimental results indicate that TenSOFO performs effectively in both cases, with better performance at the same SNR level when $R < M$. Moreover, the joint decomposition significantly improves the accuracy of estimating $\mathbf{A}$ as compared to the individual decomposition approach. The improved performance of TenSOFO in estimating $\mathbf{A}$ is primarily attributed to its joint decomposition approach. Since tensors $\boldsymbol{\mathcal{R}}$ and $\boldsymbol{\mathcal{C}}$ both contain information about $\mathbf{A}$, their integration simulates the beneficial effect of regularization and data augmentation for both $\boldsymbol{\mathcal{R}}$ and $\boldsymbol{\mathcal{C}}$. Consequently, results obtained through joint INDSCAL decomposition consistently outperform those from individual decompositions of $\boldsymbol{\mathcal{R}}$ and $\boldsymbol{\mathcal{C}}$. Furthermore, the proposed ADMM optimization also contributes to improved convergence rate and estimation accuracy as compared to the standard ALS approach.

5.1.2. Instantaneous BSS

We then demonstrate the effectiveness of TenSOFO for instantaneous BSS tasks in comparison with several widely-used BSS algorithms, namely SOBI [30], JADE [29], SO-



(a) Overdetermined BSS: Number of sources $R = 3 < M$.

(b) Underdetermined BSS: Number of sources $R = 7 > M$.

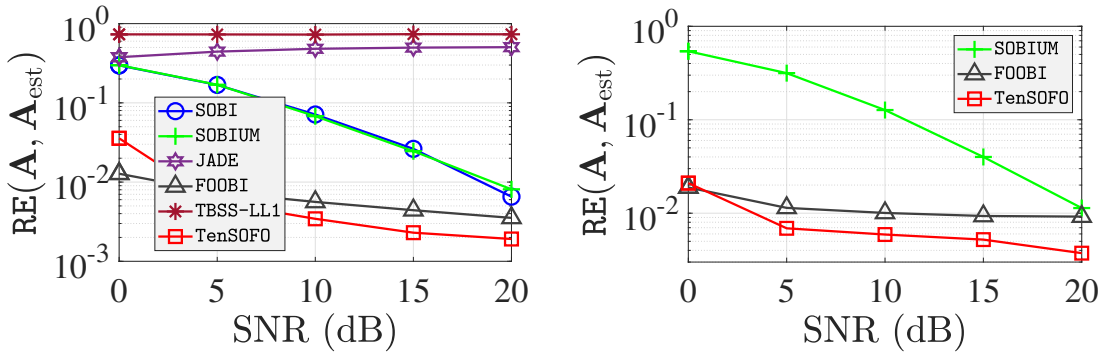
Figure 4: Instantaneous BSS tasks with $\mathbf{s}_r = \mathbf{f}_r * \mathbf{g}_r$: The number of sensors $M = 5$, data samples $T = 10000$, and time lags $N = 5$.

BIUM [13], FOOBI [12], and TBSS-LL1 (using segmentation) [10]. Since SOBI, JADE, and TBSS-LL1 are not optimally designed for handling underdetermined BSS tasks, we abstain from conducting a performance comparison among them in such cases. In the following, we present two scenarios where TenSOFO demonstrates advantages over the compared BSS algorithms.

In particular, we investigate the following BSS model

$$\mathbb{R}^M \ni \mathbf{x}[t] = \mathbf{A}\mathbf{s}[t] + \mathbf{n}[t], \ t = 0, 1, \ldots, T-1, \tag{76}$$

where the noise vector is $\mathbf{n}[t] \sim \mathcal{N}(\mathbf{0}, \sigma_n^2 \mathbf{I}_M)$ and the mixing matrix $\mathbf{A} \in \mathbb{R}^{M \times R}$ is generated as a Gaussian matrix with zero-mean and unit-variance entries. We explore two cases for the source matrix $\mathbf{S} = [\mathbf{s}_1^\top, \mathbf{s}_2^\top, \ldots, \mathbf{s}_R^\top]^\top \in \mathbb{R}^{R \times T}$: (i) each source signal $\mathbf{s}_r$ results from convoluting a kernel/filter $\mathbf{f}_r$ of length $L_f \ll T$ with a random coefficient vector $\mathbf{g}_r$ of length $T - L_f + 1$ (i.e., $\mathbf{s}_r = \mathbf{f}_r * \mathbf{g}_r$); and (ii) each source signal $\mathbf{s}_r$ is derived from filtering a non-Gaussian random process $\mathbf{v}_r$ by a first-order autoregressive (AR) model with coefficient ψ_r, denoted as $\mathbf{s}_r = \text{filter}(\mathbf{v}_r, \text{AR}(1, \psi_r))$. In the first case, $\mathbf{f}_r$ and $\mathbf{g}_r$ are generated as normal (Gaussian) and folded-normal random vectors, respectively. The filter length L_f is set to 30. In the second case, we define the non-Gaussian process for the r-th source $\mathbf{s}_r$ using a power of



(a) Overdetermined BSS: Number of sources $R = 3 < M$

(b) Underdetermined BSS: Number of sources $R = 7 > M$

Figure 5: Instantaneous BSS tasks with $\mathbf{s}_r = \text{filter}(\mathbf{v}_r, \text{AR}(1, \psi_r))$ with a non-Gaussian process $\mathbf{v}_r$, the number of sensors $M = 5$, data samples $T = 10000$, and time lags $N = 5$.

normal Gaussian distribution (i.e., $\mathbf{v}_t(i) = |y_i|^p$ where $y_i \sim \mathcal{N}(0,1), p > 1$). Here, we set the value of p to 6, and we arrange the set of coefficients $\{\psi_r\}_{r=1}^R$ to be evenly distributed within the range $[0.1, 0.9]$.[4] The hyperparameters of the compared BSS algorithms are kept at their default values. SOBI, SOBIUM, and TenSOFO require a predefined number of time lags, we set its value to $N = 5$. To evaluate the estimation accuracy of the BSS algorithms, we reuse the error metric $\text{RE}(\mathbf{A}, \mathbf{A}_{\text{est}})$ in (75).

In Fig. 4, the performance of BSS algorithms is illustrated for the case $\mathbf{s}_r = \mathbf{f}_r * \mathbf{g}_r$. In the overdetermined BSS task ($R < M$), the experimental results in Fig. 4(a) indicate that BSS algorithms utilizing second-order statistics, including SOBI, SOBIUM, and TenSOFO, effectively reconstruct the mixing matrix $\mathbf{A}$. The accuracy of their estimation improves with higher SNR. Notably, TenSOFO demonstrates slightly better estimation accuracy than SOBI and SOBIUM. Conversely, two fourth-order (FO)-based BSS algorithms (JADE and FOOBI) and tensor-based algorithm TBSS-LL1 exhibit worse performance. Fig. 4(b) illustrates the performance of TenSOFO, SOBIUM and FOOBI for underdetermined BSS. We can see that TenSOFO and SOBIUM succeed, while FOOBI fails in this task, consistent with the

[4]In MATLAB, we employ the command "`linspace`$(0.1, 0.9, R)$" to generate a sequence of R values for the set $\{\psi_r\}_{r=1}^R$ (e.g., if $R = 3$, then $\psi_1 = 0.1, \psi_2 = 0.5$, and $\psi_3 = 0.9$).

previous overdetermined BSS task.

In Fig. 5, the performance of BSS algorithms is illustrated for the second case where $\mathbf{s}_r = \text{filter}(\mathbf{v}_r, \text{AR}(1, \psi_r))$. The experimental results in Fig. 5 (a) indicate that when $R < M$, TenSOFO outperforms other methods, providing better estimation accuracy when compared to the second-best FOOBI at high SNR levels. JADE and TBSS-LL1 become less effective, whereas SOBI and SOBIUM offer moderate performance in this context. When $R > M$, both FOOBI and TenSOFO perform well in underdetermined BSS, although their estimation accuracy is lower than when dealing with $R < M$. The BSS algorithm using only SO statistics, SOBIUM, is less effective than FOOBI and TenSOFO in this case. In summary, the results presented in both Figs. 4 and 5 demonstrate that the proposed algorithm, TenSOFO, can leverage both SO and FO statistics to enhance the BSS performance in scenarios where algorithms using only SO or FO statistics may fail short.

5.1.3. Fetal Electrocardiogram Extraction via Instantaneous BSS

In this experiment, we employ TenSOFO to extract the fetal electrocardiogram (fetal ECG) from cutaneous potential recordings acquired from the mother's skin. This separation is crucial for analyzing the health and condition of the fetus [43]. The ECG dataset used in our investigation contains five abdominal and three thoracic recordings.[5] These signals were recorded using eight skin electrodes placed on various regions of a pregnant woman's body over a duration of 10 seconds, with a sampling frequency of 250 Hz. Fig. 6 illustrates the five abdominal recordings used in this study.

As indicated in [44], ECG signals can be formulated using the following linear model

$$\mathbf{x}[t] = \mathbf{A}\mathbf{s}[t] + \mathbf{n}[t], \ t = 0, 1, \ldots, T-1, \tag{77}$$

where $\mathbf{x}[t] = [x_1[t], \ldots, x_M[t]]^\top$ represents observations, $\mathbf{s}[t] = [s_1[t], \ldots, s_R(t)]^\top$ denotes the underlying sources, $\mathbf{A} \in \mathbb{R}^{M \times R}$ represents the propagation from R sources to M elec-

[5]Access the ECG recordings at: https://ftp.esat.kuleuven.be/pub/SISTA/data/biomedical/

Figure 6: Real ECG recordings obtained from the maternal abdominal region.

trodes, and $\mathbf{n}[t]$ is the noise. Consequently, the separation of fetal heartbeats from maternal heartbeats in the ECG recordings can be seen as an instantaneous BSS task [44].

As the ground truth is not available and the effectiveness of the Lowner technique has been demonstrated in [9] for this task, we follow its experimental setup and compare our separation results against it. Specifically, we use 05 recordings, each contains 500 data samples, corresponding to electrodes located on the mother's abdominal region. These ECG recordings are normalized to unit norm. The Lowner-based method requires no preprocessing and we keep their algorithmic parameters by default. For TenSOFO, we set the number of time lags N to 3. The experimental results are depicted in Fig. 7. It is worth noting that, the fetal heart rate consistently appears higher than the mother's heart rate. Both methods successfully separate fetal heartbeats from maternal heartbeats, with our method yielding similar results to the Lowner technique.

5.2. Performance Assessment for TCBSS

We assess the performance of TCBSS by (i) analyzing its convergence rate for type-2 BTD factorization and (ii) applying it to the problem of EMG decomposition, serving as an

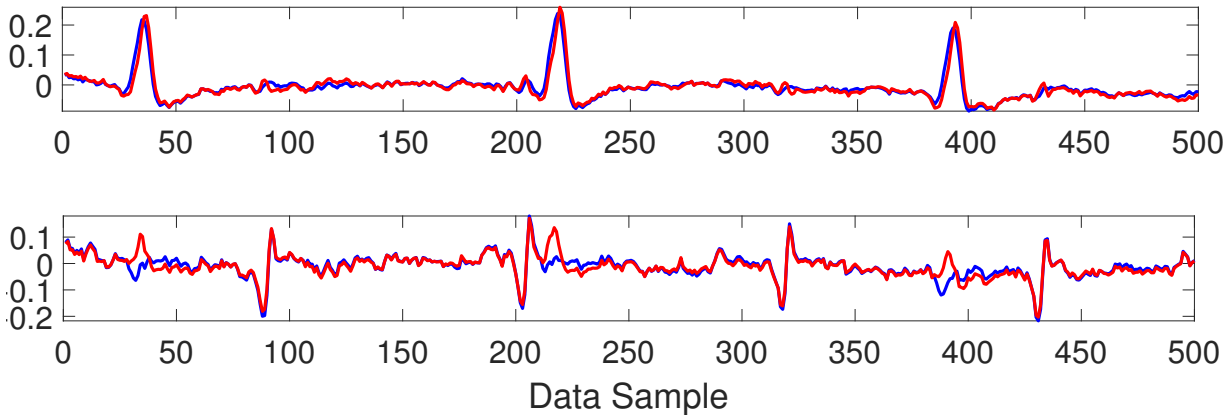


Figure 7: ECG source separation using TenSOFO (——) and Lowner-based method (——): maternal heartbeats (above) and fetal heartbeats (below).

illustrative example of convolutive BSS.

5.2.1. Type-2 BTD Factorization

We demonstrate its effectiveness in the context of type-2 BTD factorization. To illustrate this, we generate a third-order tensor $\boldsymbol{\mathcal{X}} \in \mathbb{R}^{M\times M\times K}$ as follows

$$\boldsymbol{\mathcal{X}} = \boldsymbol{\mathcal{X}}_{tr} + \sigma_n \boldsymbol{\mathcal{N}} = \sum_{r=1}^{R} \boldsymbol{\mathcal{G}}_r \times_1 \mathbf{A}_r \times_2 \mathbf{A}_r + \sigma_n \boldsymbol{\mathcal{N}}. \tag{78}$$

Here, the elements of $\boldsymbol{\mathcal{G}}_r \in \mathbb{R}^{L\times L\times K}$, $\boldsymbol{\mathcal{N}} \in \mathbb{R}^{M\times M\times K}$, and $\mathbf{A}_r \in \mathbb{R}^{M\times L}$ are derived from random Gaussian variables with zero mean and unit variance. The parameter $\sigma_n > 0$ is introduced to control the noise level. To measure the estimation accuracy, we use the following error metrics

$$\mathrm{ER}(\boldsymbol{\mathcal{X}}_{\mathrm{tr}}, \boldsymbol{\mathcal{X}}_{\mathrm{est}}) = \left\|\boldsymbol{\mathcal{X}}_{\mathrm{tr}} - \boldsymbol{\mathcal{X}}_{\mathrm{est}}\right\|_F \Big/ \left\|\boldsymbol{\mathcal{X}}_{\mathrm{tr}}\right\|_F, \tag{79}$$

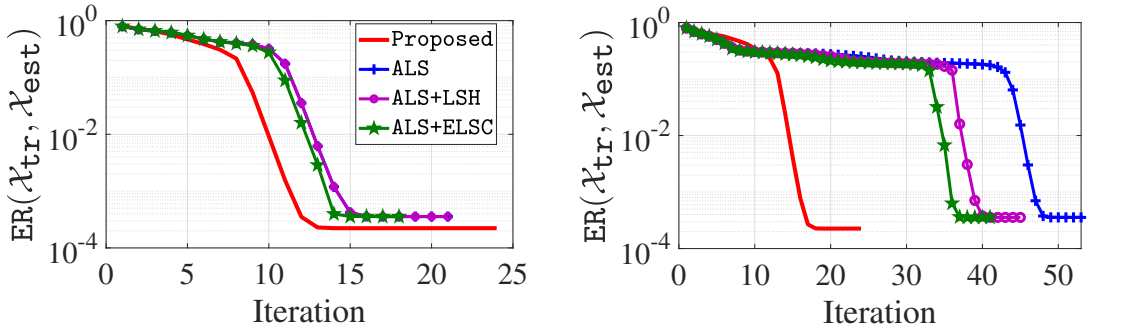
$$\mathrm{RE}(\mathbf{A}_{\mathrm{tr}}, \mathbf{A}_{\mathrm{est}}) = \min_{\boldsymbol{\Pi},\boldsymbol{\Lambda}} \left\|\mathbf{A}_{\mathrm{tr}} - \mathbf{A}_{\mathrm{est}}\boldsymbol{\Pi}\boldsymbol{\Lambda}\right\|_F \Big/ \left\|\mathbf{A}_{\mathrm{tr}}\right\|_F, \tag{80}$$

where $\boldsymbol{\mathcal{X}}_{\mathrm{est}}$, $\mathbf{A}_{\mathrm{est}}$ are the reconstructed tensor and loading factor, $\boldsymbol{\Pi}$ is a block permutation matrix, and $\boldsymbol{\Lambda}$ is a square nonsingular block-diagonal matrix.

In this investigation, we consider two cases: a small tensor with $M = 10, K = 100, R =$

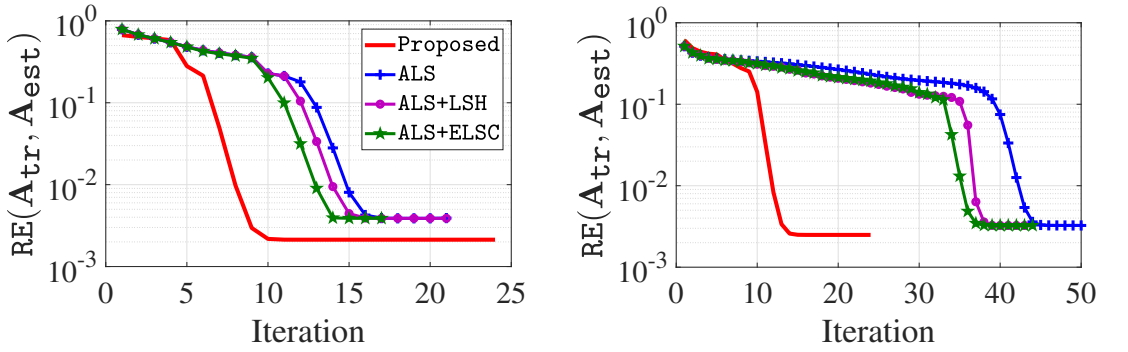
Table 1: Running time of type-2 BTD algorithms.

Setup \ Method	ALS	ALS+LSH	ALS+ELSC	TCBSS
$M = 10, K = 100, R = 2,$ $L = 2$ and $\sigma_n = 10^{-2}$	0.22(s)	0.24(s)	0.25(s)	0.17(s)
$M = 50, K = 100, R = 5,$ $L = 4$ and $\sigma_n = 10^{-2}$	0.36(s)	0.39(s)	0.43(s)	0.28(s)



(a) $M = 10, K = 100, R = 2, L = 2$ and $\sigma_n = 10^{-2}$ (b) $M = 50, K = 100, R = 5, L = 4$ and $\sigma_n = 10^{-2}$

Figure 8: Convergence rate of type-2 BTD algorithms w.r.t. ER($\boldsymbol{\mathcal{X}}_{\text{tr}}, \boldsymbol{\mathcal{X}}_{\text{est}}$).



(a) $M = 10, K = 100, R = 2, L = 2$ and $\sigma_n = 10^{-2}$ (b) $M = 50, K = 100, R = 5, L = 4$ and $\sigma_n = 10^{-2}$

Figure 9: Convergence rate of type-2 BTD algorithms w.r.t. RE($\mathbf{A}_{\text{tr}}, \mathbf{A}_{\text{est}}$).

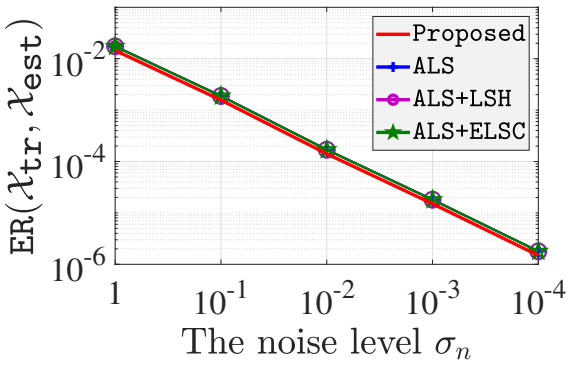


Figure 10: Effect of the noise level σ_n on the performance of type-2 BTD algorithms: $M = 10, K = 100, R = 2$, and $L = 2$.

$2, L = 2$ and a larger one with $M = 50, K = 100, R = 5, L = 4$. The noise level σ_n is selected from the range $[10^{-4}, 1]$. The performance of TCBSS is compared with the widely-used ALS method [38] and its variants, namely ALS+LSH [45] and ALS+ELSC [46], as depicted in Figs. 8, 9 and 10. While their running times are reported in Tab. 1. The results indicate that the proposed algorithm not only achieves faster convergence but also provides improved estimation accuracy in both setups.

5.2.2. EMG Decomposition via Convolutive BSS

We apply TCBSS to the problem of EMG decomposition and compare its performance with other tensor-based BSS algorithms, namely PARAFAC-SD [14] and TBSS-LL1 [10]. Synthetic EMG signals are simulated using the convolutive BSS model (1), where $a_{mr}[\ell]$ represents the action potential (AP) of the r-th motor unit (MU) at the m-th sensor. The duration of the APs is 35 (i.e., $L = 34$). The source $s_r[t] = \sum_j \delta[t - \psi_{rj}]$ corresponds to the spike train of the r-th MU, featuring spikes at times ψ_{rj}, with $\delta[\cdot]$ representing the Dirac delta function. Specifically, the m-th measurement $\mathbf{x}_m$ is formulated as

$$\mathbb{R}^T \ni \mathbf{x}_m = \sum_{r=1}^{R} \mathbf{y}_{mr} + \mathbf{n}_m = \sum_{r=1}^{R} \mathbf{a}_{mr} * \mathbf{s}_r + \mathbf{n}_m, \tag{81}$$

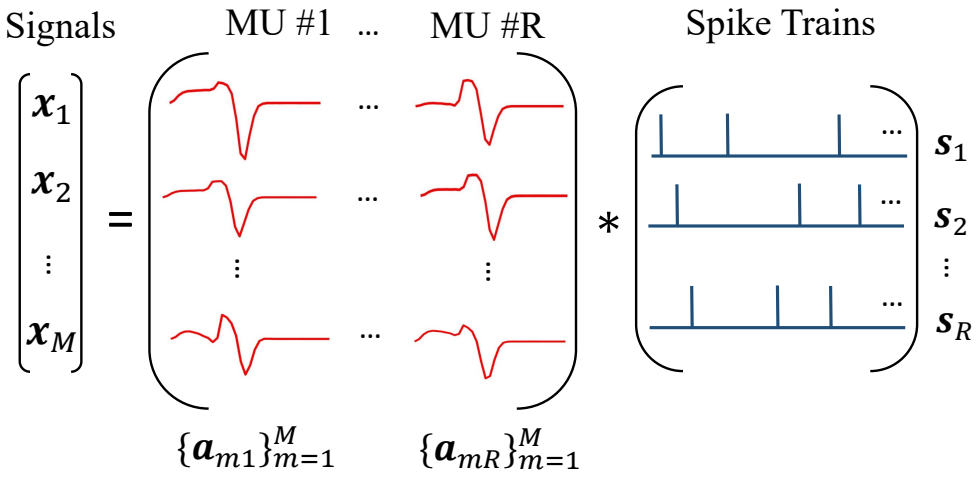


Figure 11: Synthetic EMG signals (without noise): Convolutive mixture model with M measurements and R sources.

where $\mathbf{n}_m$ is Gaussian noise. See Fig. 11 for an illustration. The objective of the EMG decomposition is to identify the underlying sources $\{\mathbf{y}_{mr}\}_{m=1,r=1}^{M,R}$ (i.e., spike trains $\{\mathbf{s}_r\}_{r=1}^{R}$ and MUs $\{\mathbf{a}_{mr}\}_{m=1,r=1}^{M,R}$) from $\{\mathbf{x}_m\}_{m=1}^{M}$. For further details, we refer the reader to [47].

In this task, three EMG measurements (M = 3) are comprised of two sources (R = 2) with an excitation level set at 5% of the maximum voluntary contraction. The sampling frequency is f_s = 2048 Hz. Each measurement contains a sequence of data samples with a total length of T = 10000 samples, corresponding to an approximate duration of 5 seconds. Four SNR levels are considered, including 0 dB, 10 dB, 20 dB and ∞ dB (without noise). The total number of spikes from the two sources is 106. To satisfy the uniqueness condition $ML' \geq R(L + L')$, we set the extension factor L' of TCBSS to $2L$. Additionally, we set the number of time lags K to 30. As $R < M$, once obtaining the mixing matrix $\underline{\mathbf{A}}$, the underlying sources are then determined up to an unknown filter using the ordinary least-squares (OLS) estimation method (i.e., $\hat{\underline{\mathbf{s}}} = \underline{\mathbf{A}}^{\#}\underline{\mathbf{x}}$). Subsequently, we identify the timings of activations of each MU, thanks to the form of spike (impulse) trains. Detected spikes exhibiting magnitude ten times lower than that of the spike with the highest magnitude are identified as artifact

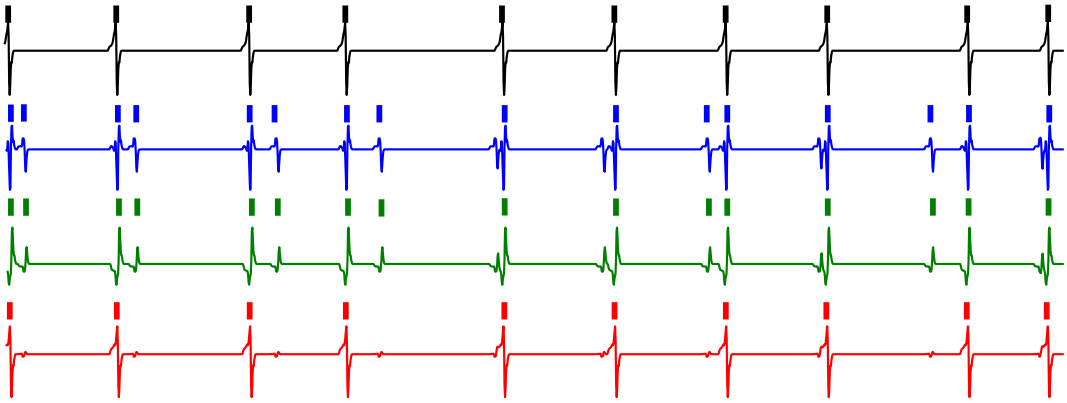


Figure 12: EMG decompositon: Illustration of a 1-second segment of decomposition results (i.e., spike train (ı) and waveform (−)) related to the first component $\mathbf{y}_{11} = \mathbf{a}_{11} * \mathbf{s}_1$ in the noise-free case: Ground truth (——), estimates by PARAFAC-SD (——), TBSS-LL1 (——), and the proposed TCBSS (——).

and will be excluded from the final detection result.

Table 2: EMG spike train detection performance.

Method	PARAFAC-SD		TBSS-LL1		TCBSS	
Metric / SNR	SEN	FDR	SEN	FDR	SEN	FDR
0 dB	87.74%	65.68%	85.85%	67.5%	86.79%	10.07%
10 dB	95.82%	49.50%	94.43%	50.98%	89.62%	9.52%
20 dB	100%	45.44%	100%	44.21%	99.06%	4.55%
∞ dB	100%	43.01%	100%	36.32%	100%	0%

For performance evaluation, we employ sensitivity (SEN) and false discovery rate (FDR) as key metrics in this task, which are defined as follows:

$$\text{SEN} = \frac{\text{TP}}{\text{TP} + \text{FN}} \quad \text{and} \quad \text{FDR} = \frac{\text{FP}}{\text{FP} + \text{TP}}. \tag{82}$$

Here, true positive (TP) represents the number of correctly identified spikes, false negative (FN) denotes the number of spikes left unidentified, and false positive (FP) refers to the number of incorrectly identified spikes. The algorithm's performance is considered better

when the value of sensitivity (SEN) is higher and the value of false discovery rate (FDR) is lower.

The decomposition results are shown in Fig. 12 and Tab. 2. At SNR = ∞ dB (without noise), TCBSS successfully recovers the underlying EMG sources and two spike trains are detected correctly without errors (i.e., SEN = 100%) and no additional impulses (spikes) are introduced (i.e., FDR = 0%). While PARAFAC-SD and TBSS-LL1 correctly identify the underlying spike trains, their false discovery rate (FDR) is reasonably high due to the presence of several other impulses in the recovered sources, as illustrated in Fig. 12 (blue and green lines). The performance of all algorithms degrades when SNR decreases. However, TCBSS remains effective in this task, with reasonable values for both SEN and FDR, much better than that of PARAFAC-SD and TBSS-LL1.

5.2.3. Separation of Convolutive Speech Mixtures

Our main objective is to demonstrate the use of TCBSS for separating sound signals in a reverberant/convolutive environment. We use a publicly available dataset that contains 18 real sound sources, sampled at a rate of 16 KHz over a time range from 24 to 30 seconds.[6] The dataset includes various sources, including speeches from both male and female speakers as well as musical sounds like piano, guitar, trumpet, and others.

In this work, our experiment involves three sound sources (male speech, female speech, and piano) recorded by three microphones, as illustrated in Fig. 13. These sounds are then mixed in a virtual room using real-world measured room impulse responses (RIRs) to simulate a realistic convolutive environment. The mixing process is facilitated by the function"`roommix.m`", which can be accessed at "https://sound.media.mit.edu/ica-bench/". We compute a set of 20 covariance matrices $\{\mathbf{R}_{\mathbf{x}}[t, \tau_n]\}_{n=1}^{20}$ to construct the third-order tensor $\underline{\boldsymbol{\mathcal{R}}}$, as in (56). Subsequently, the mixing matrix $\underline{\hat{\mathbf{A}}}$ is estimated by TCBSS, and the least-square estimate of the source vector is given by $\underline{\hat{\mathbf{s}}} = \underline{\hat{\mathbf{A}}}^{\#} \underline{\mathbf{x}}$. Due to the inherent ambiguity of

[6]Access the speech dataset at: http://dimitri.nion.free.fr/bss/BSS.html

block-term decomposition, the recovered sources are identified up to a filter. For comparison purpose, we resolve this ambiguity by determining the block permutation matrix $\mathbf{\Pi}$ and block diagonal matrix $\mathbf{\Lambda}$ through optimization, specifically by solving $\operatorname{argmin}_{\mathbf{\Pi},\mathbf{\Lambda}} \|\hat{\mathbf{A}} - \underline{\mathbf{A}}\mathbf{\Pi}\mathbf{\Lambda}\|_F^2$ where $\underline{\mathbf{A}}$ incorporates measured RIRs.[7] We then choose three sources among recovered ones based on their correlations, selecting those with the smallest correlation coefficients. The experimental results are illustrated in Fig. 14. Despite the fact that the recovered sources are not a perfect match to the ground truth (due to the filter ambiguity), it is evident that the three sources are well separated.

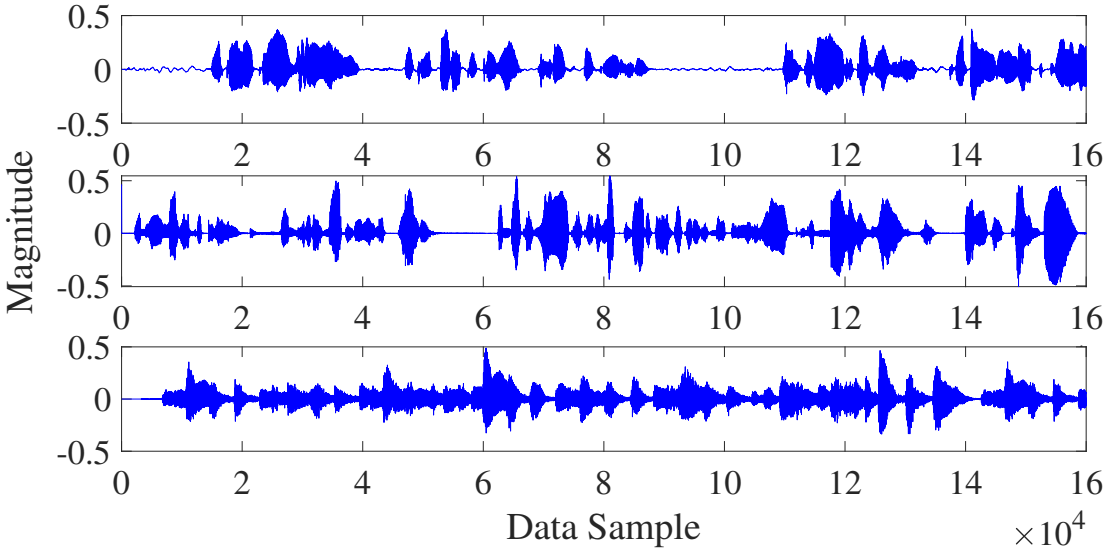
6. Conclusions

In this paper, we addressed the problem of blind source separation (BSS) through the lens of tensor decomposition. We established two fundamental connections between BSS and tensor models, which served as the basis for introducing two novel tensor-based methods: TenSOFO and TBCSS. The former is specifically designed for joint INDSCAL decomposition, addressing instantaneous BSS tasks, while the latter is an efficient constrained type-2 block term decomposition with two factors constrained to be identical, aligning its design with convolution BSS. Our experimental results indicated the effectiveness of both TenSOFO and TCBSS, showcasing their remarkable performance in both tensor decomposition and BSS tasks, particularly when compared to state-of-the-art algorithms. Notably, the proposed algorithm, TCBSS, demonstrated its capability in reconstructing EMG sources and exhibited improved performance as compared to other tensor-based methods.

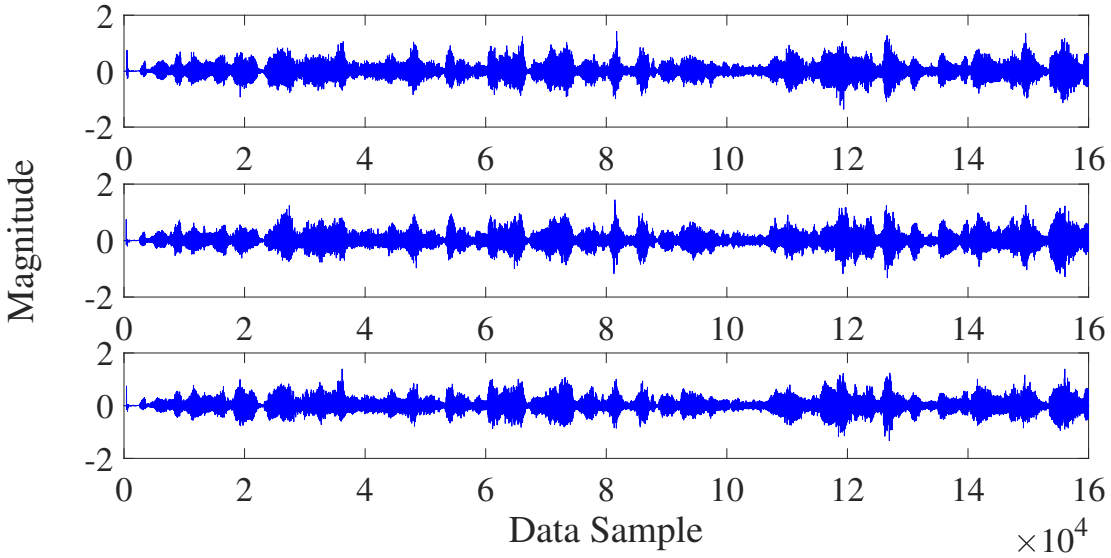
Appendix A. Proof of Theorem 1

Our convergence analysis of TenSOFO consists of three main stages: (i) demonstrating that the Lagrangian $\mathcal{L}_0\big(\mathbf{B}^{(l)}, \mathbf{K}^{(l)}, \mathbf{A}^{(l)}, \mathbf{\Sigma}^{(l)}, \mathbf{U}^{(l)}\big)$ strictly decreases with each itera-

[7] The solver is available at: http://dimitri.nion.free.fr/Codes/matlab/tools/pack_solve_scale.zip



(a) Three sound sources: female speech (above), male speech (milde), and piano (below)



(b) Data observations at three microphones in 10 seconds

Figure 13: Speech signals used in this study.

tion of ADMM, while providing an upper bound on the difference between Lagrangians computed at consecutive iterations; (ii) establishing the boundedness of both the sequence $\{\mathbf{B}^{(l)}, \mathbf{K}^{(l)}, \mathbf{A}^{(l)}, \mathbf{\Sigma}^{(l)}, \mathbf{U}^{(l)}\}$ and its corresponding Lagrangian $\mathcal{L}_0\big(\mathbf{B}^{(l)}, \mathbf{K}^{(l)}, \mathbf{A}^{(l)}, \mathbf{\Sigma}^{(l)}, \mathbf{U}^{(l)}\big)$; and (iii) indicating that the limit point of the sequence generated by TenSOFO serves as a

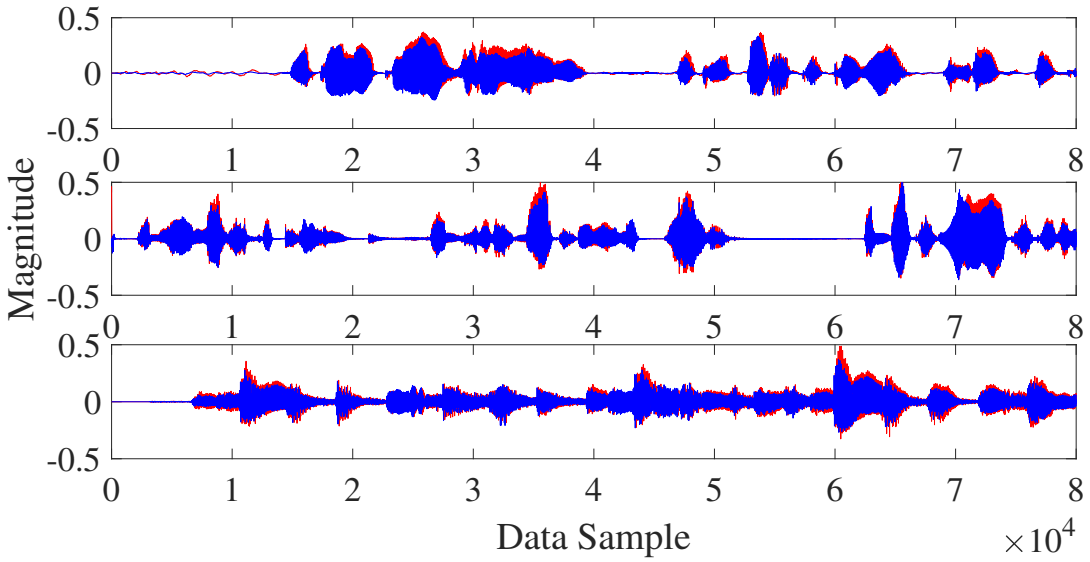


Figure 14: Recovered sources within the first 5 seconds: ground truth (—) and estimated source (—). Due the inherent ambiguity of BTD, the recovered source signals are not perfectly matched to the original ones.

stationary point of (37).

We begin with stage (i) by establishing Lemma 1.

Lemma 1 (Sufficient Descent)**.** *At each iteration l, let $\{\mathbf{B}^{(l)}, \mathbf{K}^{(l)}, \mathbf{A}^{(l)}, \mathbf{\Sigma}^{(l)}, \mathbf{U}^{(l)}\}$ denote the sequence of variables generated by TenSOFO and $\mathcal{L}_O^{(l)} = \mathcal{L}_O(\mathbf{B}^{(l)}, \mathbf{K}^{(l)}, \mathbf{A}^{(l)}, \mathbf{\Sigma}^{(l)}, \mathbf{U}^{(l)})$. If $\rho_O > 1$ and $\rho_A, \rho_B > 0$, there always exists a set of positive numbers $\{c_1, c_2, c_3, c_4\}$ such that:*

$$\begin{aligned}\mathcal{L}_O^{(l+1)} - \mathcal{L}_O^{(l)} \leq & -c_1\left\|\mathbf{B}^{(l+1)} - \mathbf{B}^{(l)}\right\|_F^2 - c_2\left\|\mathbf{K}^{(l+1)} - \mathbf{K}^{(l)}\right\|_F^2 \\ & - c_3\left\|\mathbf{A}^{(l+1)} - \mathbf{A}^{(l)}\right\|_F^2 - c_4\left\|\mathbf{\Sigma}^{(l+1)} - \mathbf{\Sigma}^{(l)}\right\|_F^2. \end{aligned} \tag{A.1}$$

Proof. We begin with proof of Lemma 1 by examining the difference in the Lagrangian $\mathcal{L}_O(\cdot)$ when $\mathbf{A}, \mathbf{\Sigma}$ and $\mathbf{U}$ are constant. Recall that, $\mathbf{B}^{(l)}$ and $\mathbf{K}^{(l)}$ are derived from the iterative procedure (42). Therefore, we first establish Proposition 4 to demonstrate the convergence of (42).

Proposition 4. *Let $\{\mathbf{B}_L^{(l,i)}, \mathbf{B}_R^{(l,i)}, \mathbf{K}^{(l,i)}\}_{i=1}^{\infty}$ denote the sequence generated by the iterative*

procedure (42) in the main text. If we denote $\mathbf{M}^{(l,i)}$ *as one of these variables, the minimizer* $\mathbf{M}^{(l,i+1)}$ *of the Lagrangian* $\mathcal{L}_{\mathcal{B}}(\cdot)$ *at the* $(i+1)$*-th iteration satisfies:*

$$\mathcal{L}_{\mathcal{B}}\big(\mathbf{M}^{(l,i+1)},\cdot\big) - \mathcal{L}_{\mathcal{B}}\big(\mathbf{M}^{(l,i)},\cdot\big) \leq -d_m\big\|\mathbf{M}^{(l,i+1)} - \mathbf{M}^{(l,i)}\big\|_F^2, \tag{A.2}$$

where d_m *is a positive number.*

Proof. Recall that $\mathbf{K}^{(l,i+1)}$ defined in (42a) is indeed the minimizer of the Lagrangian $\mathcal{L}_{\mathtt{B}}(\cdot)$ when $\mathbf{B}_{\mathtt{L}}, \mathbf{B}_{\mathtt{R}}$, and $\mathbf{D}$ are fixed, i.e.,

$$\mathbf{K}^{(l,i+1)} = \underset{\mathbf{K}}{\operatorname{argmin}}\left[\mathcal{L}_{\mathtt{B}}(\mathbf{K},\cdot) = \Big\|\mathcal{C} - [\![\mathbf{B}_{\mathtt{L}}^{(l,i)}, \mathbf{B}_{\mathtt{R}}^{(l,i)}, \mathbf{K}]\!]\Big\|_F^2 + \text{const}\right]. \tag{A.3}$$

Minimizing (A.3) is equivalent to solving the following least-squares problem

$$\mathbf{K}^{(l,i+1)} = \underset{\mathbf{K}}{\operatorname{argmin}}\left[h(\mathbf{K}) = \Big\|\big[\mathcal{C}\big]_{(3)} - \mathbf{K}\big(\mathbf{B}_{\mathtt{R}}^{(l,i)} \odot \mathbf{B}_{\mathtt{L}}^{(l,i)}\big)^{\top}\Big\|_F^2\right]. \tag{A.4}$$

In particular, the least-squares function $h(\mathbf{K})$ in (A.4) is strongly convex w.r.t. the matrix $\mathbf{K}$ as its Hessian matrix $\big(\mathbf{B}_{\mathtt{R}}^{(l,i)} \odot \mathbf{B}_{\mathtt{L}}^{(l,i)}\big)^{\top}\big(\mathbf{B}_{\mathtt{R}}^{(l,i)} \odot \mathbf{B}_{\mathtt{L}}^{(l,i)}\big)$ is positive definite, and thus,

$$h\big(\mathbf{K}^{(\mathtt{l},i+1)}\big) \leq h\big(\mathbf{K}^{(l,i)}\big) - \frac{1}{2}\Big\|\mathbf{K}^{(l,i+1)} - \mathbf{K}^{(l,i)}\Big\|_F^2, \tag{A.5}$$

In other words, we have

$$\mathcal{L}_{\mathtt{B}}\big(\mathbf{K}^{(l,i+1)},\cdot\big) - \mathcal{L}_{\mathtt{B}}\big(\mathbf{K}^{(l,i)},\cdot\big) \leq -\frac{1}{2}\Big\|\mathbf{K}^{(l,i+1)} - \mathbf{K}^{(l,i)}\Big\|_F^2. \tag{A.6}$$

Next, we examine the matrix $\mathbf{B}_{\mathtt{L}}^{(l,i+1)}$ which is particularly derived from

$$\begin{aligned}
\mathbf{B}_{\mathtt{L}}^{(l,i+1)} = \underset{\mathbf{B}_{\mathtt{L}}}{\operatorname{argmin}}\Bigg[&\mathcal{L}_{\mathtt{B}}(\mathbf{B}_{\mathtt{L}},\cdot) = \Big\|\mathcal{C} - [\![\mathbf{B}_{\mathtt{L}}, \mathbf{B}_{\mathtt{R}}^{(l,i)}, \mathbf{K}^{(l,i+1)}]\!]\Big\|_F^2 \\
&+ \frac{\rho_{\mathtt{0}}}{2}\Big\|\mathbf{B}_{\mathtt{L}} - \mathbf{A}_{\odot 2}^{(l-1)} + \mathbf{U}^{(l-1)}\Big\|_F^2 + \frac{\rho_{\mathtt{B}}}{2}\Big\|\mathbf{B}_{\mathtt{L}} - \mathbf{B}_{\mathtt{R}}^{(l,i)} + \mathbf{D}^{(l,i)}\Big\|_F^2 + \text{const}\Bigg] \\
= \underset{\mathbf{B}_{\mathtt{L}}}{\operatorname{argmin}}\Bigg[&\mathcal{L}_{\mathtt{B}}(\mathbf{B}_{\mathtt{L}},\cdot) = \Big\|\big[\mathcal{C}\big]_{(1)} - \mathbf{B}_{\mathtt{L}}\big(\mathbf{K}^{(l,i+1)} \odot \mathbf{B}_{\mathtt{R}}^{(l,i)}\big)^{\top}\Big\|_F^2 \\
&+ \frac{\rho_{\mathtt{0}}}{2}\Big\|\mathbf{B}_{\mathtt{L}} - \mathbf{A}_{\odot 2}^{(l-1)} + \mathbf{U}^{(l-1)}\Big\|_F^2 + \frac{\rho_{\mathtt{B}}}{2}\Big\|\mathbf{B}_{\mathtt{L}} - \mathbf{B}_{\mathtt{R}}^{(l,i)} + \mathbf{D}^{(l,i)}\Big\|_F^2 + \text{const}\Bigg].
\end{aligned} \tag{A.7}$$

All three terms of the right hand side (RHS) of (A.7) are in the least-squares form. Thus, the objective function (A.7) are strongly convex, characterized by the following positive definite

Hessian matrix

$$\nabla^2 \mathcal{L}_{\mathtt{B}}\big(\mathbf{B}_{\mathtt{L}}, \cdot\big) = (\rho_{\mathtt{0}} + \rho_{\mathtt{B}})\mathbf{I} + \big(\mathbf{K}^{(l,i+1)} \odot \mathbf{B}_{\mathtt{R}}^{(l,i)}\big)^{\top}\big(\mathbf{K}^{(l,i+1)} \odot \mathbf{B}_{\mathtt{R}}^{(l,i)}\big) \succ \mathbf{0}. \tag{A.8}$$

Accordingly, we derive

$$\mathcal{L}_{\mathtt{B}}\big(\mathbf{B}_{\mathtt{L}}^{(l,i+1)}, \cdot\big) - \mathcal{L}_{\mathtt{B}}\big(\mathbf{B}_{\mathtt{L}}^{(l,i)}, \cdot\big) \leq -\frac{d_{1i}}{2}\Big\|\mathbf{B}_{\mathtt{L}}^{(l,i+1)} - \mathbf{B}_{\mathtt{L}}^{(l,i)}\Big\|_F^2, \tag{A.9}$$

with $d_{1i} = \rho_{\mathtt{0}} + \rho_{\mathtt{B}} + \lambda_{\min}(\mathbf{K}^{(l,i+1)} \odot \mathbf{B}_{\mathtt{R}}^{(l,i)})$ where $\lambda_{\min}(\mathbf{N})$ denotes the smallest singular value of matrix $\mathbf{N}$.

The matrix $\mathbf{B}_{\mathtt{R}}^{(l,i+1)}$ is the minimizer of the following optimization

$$\begin{aligned}
\mathbf{B}_{\mathtt{R}}^{(l,i+1)} = \underset{\mathbf{B}_{\mathtt{R}}}{\operatorname{argmin}} \Bigg[& \mathcal{L}_{\mathtt{B}}(\mathbf{B}_{\mathtt{R}}, \cdot) = \Big\|\mathcal{C} - [\![\mathbf{B}_{\mathtt{L}}^{(l,i+1)}, \mathbf{B}_{\mathtt{R}}, \mathbf{K}^{(l,i+1)}]\!]\Big\|_F^2 \\
& + \frac{\rho_{\mathtt{B}}}{2}\Big\|\mathbf{B}_{\mathtt{L}}^{(l,i+1)} - \mathbf{B}_{\mathtt{R}} + \mathbf{D}^{(l,i)}\Big\|_F^2 + \text{const} \Bigg] \\
= \underset{\mathbf{B}_{\mathtt{R}}}{\operatorname{argmin}} \Bigg[& \mathcal{L}_{\mathtt{B}}(\mathbf{B}_{\mathtt{R}}, \cdot) = \Big\|\big[\mathcal{C}\big]_{(2)} - \mathbf{B}_{\mathtt{R}}\big(\mathbf{K}^{(l,i+1)} \odot \mathbf{B}_{\mathtt{L}}^{(l,i+1)}\big)^{\top}\Big\|_F^2 \\
& + \frac{\rho_{\mathtt{B}}}{2}\Big\|\mathbf{B}_{\mathtt{L}}^{(l,i+1)} - \mathbf{B}_{\mathtt{R}} + \mathbf{D}^{(l,i)}\Big\|_F^2 + \text{const} \Bigg].
\end{aligned} \tag{A.10}$$

Similar to (A.7), the objective function (A.10) is strongly convex with respect to $\mathbf{B}_{\mathtt{R}}$. Consequently, we have:

$$\mathcal{L}_{\mathtt{B}}\big(\mathbf{B}_{\mathtt{R}}^{(l,i+1)}, \cdot\big) - \mathcal{L}_{\mathtt{B}}\big(\mathbf{B}_{\mathtt{R}}^{(l,i)}, \cdot\big) \leq -\frac{d_{2i}}{2}\Big\|\mathbf{B}_{\mathtt{L}}^{(l,i+1)} - \mathbf{B}_{\mathtt{L}}^{(l,i)}\Big\|_F^2, \tag{A.11}$$

with $d_{2i} = \rho_{\mathtt{B}} + \lambda_{\min}(\mathbf{K}^{(l,i+1)} \odot \mathbf{B}_{\mathtt{L}}^{(l,i+1)})$. It ends the proof. □

Thanks to Proposition 4, we observe that the set of the objective values of $\mathcal{L}_{\mathtt{B}}(\cdot)$ at solutions is monotonically decreasing and bounded. Thanks to monotone convergence theorem, we can conclude that the cluster $\big\{\mathbf{B}_{\mathtt{L}}^{(l,i)}, \mathbf{B}_{\mathtt{R}}^{(l,i)}, \mathbf{K}^{(l,i)}\big\}$ converges to a limit point as i tends to infinity. This convergence implies that the solution $\big\{\mathbf{B}^{(l)}, \mathbf{K}^{(l)}\big\}$ of problem (P1.1) generated by the iterative procedure (42) converge to a stationary point of $\mathcal{L}_{\mathtt{B}}(\cdot)$ as its gradients with respect to both matrices are zero, according to our optimization framework. In addition, the Lagrangian $\mathcal{L}_{\mathtt{0}}(\mathbf{B}, \mathbf{K}, \cdot)$ in (38) comprises three convex terms with respect to $\mathbf{B}$ and $\mathbf{K}$, including a function involving the Frobenius inner product $\big\langle \mathbf{Z}, \mathbf{B} - \mathbf{A}_{\odot 2}\big\rangle$ and two least-squares

functions $f(\mathbf{B},\mathbf{K})$ and $\frac{\rho_0}{2}\|\mathbf{B}-\mathbf{A}_{\odot 2}\|_F^2$. Therefore, $\mathcal{L}_0(\mathbf{B},\mathbf{K},\cdot)$ is a multi-convex function and using the same arguments above (e.g., as for (A.7)) results in

$$\begin{aligned}\mathcal{L}_0\big(\mathbf{B}^{(l+1)},\mathbf{K}^{(l+1)},\cdot\big)-\mathcal{L}_0\big(\mathbf{B}^{(l)},\mathbf{K}^{(l)},\cdot\big)\\ \leq -\frac{c_1}{2}\Big\|\mathbf{B}^{(l+1)}-\mathbf{B}^{(l)}\Big\|_F^2-\frac{c_2}{2}\Big\|\mathbf{K}^{(l+1)}-\mathbf{K}^{(l)}\Big\|_F^2,\end{aligned} \tag{A.12}$$

where c_1 and c_2 are any positive numbers satisfying $\nabla^2\mathcal{L}_0(\mathbf{B},\cdot)\succeq c_1\mathbf{I}$ and $\nabla^2\mathcal{L}_0(\mathbf{K},\cdot)\succeq c_2\mathbf{I}$.

Similarly, employing the same approach as in the proof of Proposition 4 and arguments above, we also obtain

$$\begin{aligned}\mathcal{L}_0\big(\mathbf{A}^{(l+1)},\mathbf{\Sigma}^{(l+1)},\cdot\big)-\mathcal{L}_0\big(\mathbf{A}^{(l)},\mathbf{\Sigma}^{(l)},\cdot\big)\\ \leq -\frac{c_3}{2}\Big\|\mathbf{A}^{(l+1)}-\mathbf{A}^{(l)}\Big\|_F^2-\frac{c_4}{2}\Big\|\mathbf{\Sigma}^{(l+1)}-\mathbf{\Sigma}^{(l)}\Big\|_F^2,\end{aligned} \tag{A.13}$$

where c_3 and c_4 are positive numbers. The result in Lemma 1 follows the combination of two inequalities (A.12) and (A.13). It ends the proof. □

Proposition 5 (Boundedness)**.** *The sequence* $\{\mathbf{B}^{(l)},\mathbf{K}^{(l)},\mathbf{A}^{(l)},\mathbf{\Sigma}^{(l)},\mathbf{U}^{(l)}\}$ *generated by TenSOFO and the Lagrangian* $\mathcal{L}_0\big(\mathbf{B}^{(l)},\mathbf{K}^{(l)},\mathbf{A}^{(l)},\mathbf{\Sigma}^{(l)},\mathbf{U}^{(l)}\big)$ *are bounded.*

Proof. Thanks to Lemma 1, we obtain $\mathcal{L}_0^{(l)}\leq\mathcal{L}_0^{(0)}\ \forall l$, i.e., the Lagrangian $\mathcal{L}_0(\cdot)$ has an upper bound $\mathcal{L}_0$. Next, we consider the the lower bound for $\mathcal{L}_0^{(l)}$.

Employing the dual variable update $\mathbf{U}^{(l)}$ in (P1.3), we can recast the formulation of $\mathcal{L}_0(\cdot)$ in (39) into the following form

$$\begin{aligned}\mathcal{L}_0^{(l)}=\big\|\mathcal{C}-[\![\mathbf{B}^{(l)},\mathbf{B}^{(l)},\mathbf{K}^{(l)}]\!]\big\|_F^2+\big\|\mathcal{R}-[\![\mathbf{A}^{(l)},\mathbf{A}^{(l)},\mathbf{\Sigma}^{(l)}]\!]\big\|_F^2\\ +\frac{\rho_0-1}{2}\big\|\mathbf{B}^{(l)}-\mathbf{A}_{\odot 2}^{(l)}\big\|_F^2.\end{aligned} \tag{A.14}$$

When $\rho_0>1$, the RHS of (A.14) is guaranteed to be positive, indicating that the Lagrangian $\mathcal{L}_0(\cdot)$ is bounded below by zero. The boundedness of Lagrangian suggests that all (positive) terms within (A.14) are also bounded. Consequently, we can infer that the sequence $\{\mathbf{B}^{(l)},\mathbf{A}^{(l)}\}$ is bounded due to the boundedness of $\|\mathbf{B}^{(l)}-\mathbf{A}_{\odot 2}^{(l)}\|_F^2$. Expressing the first term of (A.14) as $\big\|[\![\mathcal{C}]\!]_{(3)}-\mathbf{K}^{(l)}(\mathbf{B}^{(l)}\odot\mathbf{B}^{(l)})^\top\big\|_F^2$, combined with the boundedness of

$\mathcal{C}$, ensures that $\{\mathbf{K}^{(l)}\}$ is bounded. Following the same arguments above, we also conclude $\{\mathbf{\Sigma}^{(l)}\}$ is bounded. The boundedness of the scaled dual variable $\{\mathbf{U}^{(l)}\}$ stems from the boundedness of the third term, $\left\langle \rho_0 \mathbf{U}^{(l)}, \mathbf{B}^{(l)} - \mathbf{A}_{\odot 2}^{(l)} \right\rangle$, in the original version (38) of $\mathcal{L}_0(\cdot)$ (note that $\mathbf{Z} = \rho_0 \mathbf{U}$). □

Now, iterating the inequality (A.1) in Lemma 1 for $l = 0, 1, \ldots, L$ results in

$$\begin{aligned}\mathcal{L}_0^{(0)} - \mathcal{L}_0^{(L+1)} \geq c_1 \sum_{l=0}^{L} \left\|\mathbf{B}^{(l+1)} - \mathbf{B}^{(l)}\right\|_F^2 + c_2 \sum_{l=0}^{L} \left\|\mathbf{K}^{(l+1)} - \mathbf{K}^{(l)}\right\|_F^2 \\ + c_3 \sum_{l=0}^{L} \left\|\mathbf{A}^{(l+1)} - \mathbf{A}^{(l)}\right\|_F^2 + c_4 \sum_{l=0}^{L} \left\|\mathbf{\Sigma}^{(l+1)} - \mathbf{\Sigma}^{(l)}\right\|_F^2. \end{aligned} \tag{A.15}$$

Due to the Lagrangian $\mathcal{L}_0(\cdot)$ is decreasing and bounded, we get

$$\sum_{l=0}^{\infty} \left\|\mathbf{B}^{(l+1)} - \mathbf{B}^{(l)}\right\|_F^2 < \infty, \qquad \sum_{l=0}^{\infty} \left\|\mathbf{K}^{(l+1)} - \mathbf{K}^{(l)}\right\|_F^2 < \infty, \tag{A.16}$$

$$\sum_{l=0}^{\infty} \left\|\mathbf{A}^{(l+1)} - \mathbf{A}^{(l)}\right\|_F^2 < \infty, \qquad \sum_{l=0}^{\infty} \left\|\mathbf{\Sigma}^{(l+1)} - \mathbf{\Sigma}^{(l)}\right\|_F^2 < \infty, \tag{A.17}$$

as L goes to infinity. In other words, we conclude

$$\left\|\mathbf{B}^{(l+1)} - \mathbf{B}^{(l)}\right\|_F^2 \to 0, \qquad \left\|\mathbf{K}^{(l+1)} - \mathbf{K}^{(l)}\right\|_F^2 \to 0, \tag{A.18}$$

$$\left\|\mathbf{A}^{(l+1)} - \mathbf{A}^{(l)}\right\|_F^2 \to 0, \qquad \left\|\mathbf{\Sigma}^{(l+1)} - \mathbf{\Sigma}^{(l)}\right\|_F^2 \to 0. \tag{A.19}$$

Proposition 6. *Let* $(\mathbf{B}^\star, \mathbf{K}^\star, \mathbf{A}^\star, \mathbf{\Sigma}^\star, \mathbf{U}^\star)$ *be a saddle point of* $\mathcal{L}_0(\cdot)$ *and define*

$$V^{(l)} = \rho_0 \left\|\mathbf{U}^{(l)} - \mathbf{U}^\star\right\|_F^2 + \rho_0 \left\|\mathbf{A}_{\odot 2}^{(l)} - \mathbf{A}_{\odot 2}^\star\right\|_F^2. \tag{A.20}$$

Then $V^{(l)}$ *is a Lyapunov function for TenSOFO, i.e.,*

$$V^{(l+1)} - V^{(l)} \leq -\rho_0 \left\|\mathbf{U}^{(l+1)} - \mathbf{U}^{(l)}\right\|_F^2 - \rho_0 \|\mathbf{A}_{\odot 2}^{(l+1)} - \mathbf{A}_{\odot 2}^{(l)}\|_F^2. \tag{A.21}$$

Proof. Its proof follows the same mathematical framework used in the convergence analysis of the standard ADMM method, as detailed in [39, Appendix A]. □

Proposition 6 also demonstrates the boundedness of the the sequence of the (scaled) dual variable $\{\mathbf{U}^{(l)}\}$. Subsequently, we utilize the same arguments as in (A.15)-(A.19) to

establish

$$\sum_{l=0}^{\infty} \rho_0 \left\| \mathbf{U}^{(l+1)} - \mathbf{U}^{(l)} \right\|_F^2 < V^{(0)} < \infty \tag{A.22}$$

$$\left\| \mathbf{U}^{(l+1)} - \mathbf{U}^{(l)} \right\|_F^2 \to 0 \text{ as } l \to \infty. \tag{A.23}$$

To sum up, the sequence $\left\{\mathbf{B}^{(l)}, \mathbf{K}^{(l)}, \mathbf{A}^{(l)}, \mathbf{\Sigma}^{(l)}, \mathbf{U}^{(l)}\right\}$ generated by TenSOFO converges to a limit point $(\mathbf{B}^{\infty}, \mathbf{K}^{\infty}, \mathbf{A}^{\infty}, \mathbf{\Sigma}^{\infty}, \mathbf{U}^{\infty})$ as l goes to infinity, and

$$\mathcal{L}_0\big(\mathbf{B}^{\infty}, \mathbf{K}^{\infty}, \mathbf{A}^{\infty}, \mathbf{\Sigma}^{\infty}, \mathbf{U}^{\infty}\big) = \lim_{l \to \infty} \mathcal{L}_0\big(\mathbf{B}^{(l)}, \mathbf{K}^{(l)}, \mathbf{A}^{(\ell)}, \mathbf{\Sigma}^{(l)}, \mathbf{U}^{(l)}\big). \tag{A.24}$$

In the final step, we examine the gradient of $\mathcal{L}_0(\cdot)$ at the limit point $(\mathbf{B}^{\infty}, \mathbf{K}^{\infty}, \mathbf{A}^{\infty}, \mathbf{\Sigma}^{\infty}, \mathbf{U}^{\infty})$. In short, we denote $\boldsymbol{\mathcal{Z}} \coloneqq (\mathbf{B}, \mathbf{K}, \mathbf{A}, \mathbf{\Sigma}, \mathbf{U})$ and $\boldsymbol{\mathcal{Z}}^{\infty} \coloneqq (\mathbf{B}^{\infty}, \mathbf{K}^{\infty}, \mathbf{A}^{\infty}, \mathbf{\Sigma}^{\infty}, \mathbf{U}^{\infty})$. First, we observe that partial derivatives of $\mathcal{L}_0(\cdot)$ w.r.t. each variable is Lipschitz continuous (as multi-block convex functions are locally Lipschitz). Accordingly, we obtain

$$\left| \mathcal{L}_0(\boldsymbol{\mathcal{Z}}^{(l)}) - \mathcal{L}_0(\boldsymbol{\mathcal{Z}}^{(l+1)}) - \operatorname{tr}\Big[\big(\boldsymbol{\mathcal{Z}}^{(l)} - \boldsymbol{\mathcal{Z}}^{(l+1)}\big)^{\top} \nabla \mathcal{L}_0(\boldsymbol{\mathcal{Z}}^{(l+1)}) \Big] \right| \leq \tilde{L} \left\| \boldsymbol{\mathcal{Z}}^{(l)} - \boldsymbol{\mathcal{Z}}^{(l+1)} \right\|_F^2, \tag{A.25}$$

where $\tilde{L}$ represents the maximum Lipschitz constant of all partial derivatives of $\mathcal{L}_0(\cdot)$. We then have

$$\operatorname{tr}\Big[\big(\boldsymbol{\mathcal{Z}}^{(l)} - \boldsymbol{\mathcal{Z}}^{(l+1)}\big)^{\top} \nabla \mathcal{L}_0(\boldsymbol{\mathcal{Z}}^{(l+1)}) \Big] \leq \tilde{L} \left\| \boldsymbol{\mathcal{Z}}^{(l)} - \boldsymbol{\mathcal{Z}}^{(l+1)} \right\|_F^2 + \mathcal{L}_0^{(l)} - \mathcal{L}_0^{(l+1)}, \tag{A.26}$$

thanks to the triangle inequality. Hence, summing (A.26) for $l = 0, 1, \ldots, \infty$ results in

$$\sum_{l=0}^{\infty} \operatorname{tr}\Big[\big(\boldsymbol{\mathcal{Z}}^{(l)} - \boldsymbol{\mathcal{Z}}^{(l+1)}\big)^{\top} \nabla \mathcal{L}_0(\boldsymbol{\mathcal{Z}}^{(l+1)}) \Big] < \infty, \tag{A.27}$$

It implies that

$$\lim_{l \to \infty} \operatorname{tr}\Big[\big(\boldsymbol{\mathcal{Z}}^{(l)} - \boldsymbol{\mathcal{Z}}^{(l+1)}\big)^{\top} \nabla \mathcal{L}_0(\boldsymbol{\mathcal{Z}}^{(l+1)}) \Big] = 0. \tag{A.28}$$

Inspired by our companion work on CP decomposition in [48], we can apply the proof by contradiction to indicate that $\boldsymbol{\mathcal{Z}}^{(\infty)}$ is a stationary point of $\mathcal{L}_0^{(\infty)} \triangleq \lim_{l \to \infty} \mathcal{L}_0(\cdot)$. Specifically, assume that $\boldsymbol{\mathcal{Z}}^{(\infty)}$ is not a stationary point of $\mathcal{L}_0^{(l)}$ when $l \to \infty$. In such a case, there exist $\boldsymbol{\mathcal{Z}}$ and $\varepsilon > 0$ satisfying

$$\operatorname{tr}\Big[\big(\boldsymbol{\mathcal{Z}} - \boldsymbol{\mathcal{Z}}^{(\infty)}\big)^{\top} \nabla \mathcal{L}_0^{(\infty)}(\boldsymbol{\mathcal{Z}}^{(\infty)}) \Big] < -\varepsilon < 0. \tag{A.29}$$

Thanks to the triangle inequality, we have

$$\begin{aligned}\Big\|\big(\mathcal{Z}-\mathcal{Z}^{(l+1)}\big)^{\top}\nabla\mathcal{L}_0(\mathcal{Z}^{(l+1)})-\big(\mathcal{Z}-\mathcal{Z}^{(\infty)}\big)^{\top}\nabla\mathcal{L}_0^{(\infty)}(\mathcal{Z}^{(\infty)})\Big\|_F^2 \\ \leq \Big\|\nabla\mathcal{L}_0(\mathcal{Z}^{(l+1)})-\nabla\mathcal{L}_0^{(\infty)}(\mathcal{Z}^{(\infty)})\Big\|_F^2\Big\|\mathcal{Z}-\mathcal{Z}^{(l+1)}\Big\|_F^2 \\ +\Big\|\nabla\mathcal{L}_0^{(\infty)}(\mathcal{Z}^{(\infty)})\Big\|_F^2\Big\|\mathcal{Z}^{(\infty)}-\mathcal{Z}^{(l+1)}\Big\|_F^2 \qquad (A.30)\end{aligned}$$

We observe that when $l\to\infty$, the RHS of (A.30) tends to zero due to $\mathcal{Z}^{(l+1)}\to\mathcal{Z}^{(\infty)}$ and $\nabla\mathcal{L}_0(\mathcal{Z}^{(l+1)})\to\nabla\mathcal{L}_0^{(\infty)}(\mathcal{Z}^{(\infty)})$. From (A.29), we derive

$$\mathrm{tr}\Big[\big(\mathcal{Z}-\mathcal{Z}^{(l+1)}\big)^{\top}\nabla\mathcal{L}_0(\mathcal{Z}^{(l+1)})\Big]\leq-\varepsilon<0. \qquad (A.31)$$

As mentioned above, partial derivatives of $\mathcal{L}_0(\cdot)$ are Lipschitz and we know that $\mathcal{Z}^{(l+1)}=\mathrm{argmin}_{\mathcal{Z}}\,\mathcal{L}_0(\mathcal{Z})$ at the $(l+1)$-th iteration. Due to Lipschitz continuity, we can further obtain

$$\begin{aligned}\mathrm{tr}\left[\frac{\big(\mathcal{Z}^{(l)}-\mathcal{Z}^{(l+1)}\big)^{\top}}{\|\mathcal{Z}-\mathcal{Z}^{(l+1)}\|_F}\nabla\mathcal{L}_0(\mathcal{Z}^{(l+1)})\right]\leq\inf\mathrm{tr}\left[\frac{\big(\mathcal{Z}-\mathcal{Z}^{(l+1)}\big)^{\top}}{\|\mathcal{Z}-\mathcal{Z}^{(l+1)}\|_F}\nabla\mathcal{L}_0(\mathcal{Z}^{(l+1)})\right] \\ +c\|\mathcal{Z}^{(l)}-\mathcal{Z}^{(l+1)}\|_F^2, \qquad (A.32)\end{aligned}$$

where c is a positive number, thanks to [49, Proposition 9]. Let l go to infinity, (A.32) becomes

$$\lim_{l\to\infty}\mathrm{tr}\Big[\big(\mathcal{Z}^{(l)}-\mathcal{Z}^{(l+1)}\big)^{\top}\nabla\mathcal{L}_0(\mathcal{Z}^{(l+1)})\Big]\leq-\varepsilon<0, \qquad (A.33)$$

because of $\|\mathcal{Z}^{(l)}-\mathcal{Z}^{(l+1)}\|_F^2\to0$ as $l\to\infty$. Here, (A.33) is a contradiction in (A.28). Therefore, $\mathcal{Z}^{(\infty)}$ must be a stationary point of $\mathcal{L}_0^{(\infty)}$. It ends the proof.

References

[1] P. Comon and C. Jutten, *Handbook of Blind Source Separation: Independent Component Analysis and Applications*, 2010.

[2] G. Chabriel, M. Kleinsteuber, E. Moreau *et al.*, "Joint matrices decompositions and blind source separation: A survey of methods, identification, and applications," *IEEE Signal Process. Mag.*, vol. 31, no. 3, pp. 34–43, 2014.

[3] M. S. Pedersen, J. Larsen, U. Kjems, and L. C. Parra, "A survey of convolutive blind source separation methods," *Multichannel Speech Processing Handbook*, pp. 114–126, 2007.

[4] T. G. Kolda and B. W. Bader, "Tensor decompositions and applications," *SIAM Rev.*, vol. 51, no. 3, pp. 455–500, 2009.

[5] A. Cichocki *et al.*, "Tensor decompositions for signal processing applications: From two-way to multiway component analysis," *IEEE Signal Process. Mag.*, vol. 32, no. 2, pp. 145–163, 2015.

[6] L. T. Thanh, K. Abed-Meraim, N. L. Trung, and A. Hafiane, "A contemporary and comprehensive survey on streaming tensor decomposition," *IEEE Trans. Knowl. Data Eng.*, vol. 35, no. 11, pp. 10 897–10 921, 2023.

[7] S. Xie, L. Yang, J.-M. Yang, G. Zhou, and Y. Xiang, "Time-frequency approach to underdetermined blind source separation," *IEEE Trans. Neural Netw. Learning Syst.*, vol. 23, no. 2, pp. 306–316, 2012.

[8] L. De Lathauwer, "Blind separation of exponential polynomials and the decomposition of a tensor in rank-(L_r,L_r,1) terms," *SIAM J. Matrix Anal. Appl.*, vol. 32, no. 4, pp. 1451–1474, 2011.

[9] O. Debals, M. Van Barel, and L. De Lathauwer, "Löwner-based blind signal separation of rational functions with applications," *IEEE Trans. Signal Process.*, vol. 64, no. 8, pp. 1909–1918, 2016.

[10] M. Boussé, O. Debals, and L. De Lathauwer, "A tensor-based method for large-scale blind source separation using segmentation," *IEEE Trans. Signal Process.*, vol. 65, no. 2, pp. 346–358, 2017.

[11] L. De Lathauwer, "A link between the canonical decomposition in multilinear algebra and simultaneous matrix diagonalization," *SIAM J. Matrix Anal. Appl.*, vol. 28, no. 3, pp. 642–666, 2006.

[12] L. De Lathauwer, J. Castaing, and J.-F. Cardoso, "Fourth-order cumulant-based blind identification of underdetermined mixtures," *IEEE Trans. Signal Process.*, vol. 55, no. 6, pp. 2965–2973, 2007.

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

[14] D. Nion, K. N. Mokios, N. D. Sidiropoulos, and A. Potamianos, "Batch and adaptive PARAFAC-based blind separation of convolutive speech mixtures," *IEEE Trans. Audio Speech Language Process.*, vol. 18, no. 6, pp. 1193–1207, 2010.

[15] X.-F. Gong, Q.-H. Lin, F.-Y. Cong, and L. De Lathauwer, "Double coupled canonical polyadic decomposition for joint blind source separation," *IEEE Trans. Signal Process.*, vol. 66, no. 13, pp. 3475–3490, 2018.

[16] L. De Lathauwer *et al.*, "Blind source separation by higher-order singular value decomposition," in *Proc. Eur. Signal Process. Conf.*, 1994, pp. 175–178.

[17] V. Zarzoso, A. Nandi, and E. Bacharakis, "Maternal and foetal ECG separation using blind source separation methods," *Math. Med. Biol.*, vol. 14, no. 3, pp. 207–225, 1997.

[18] V. D. Vrabie *et al.*, "Multicomponent wave separation using HOSVD/unimodal-ICA subspace method," *Geophysics*, vol. 71, no. 5, pp. V133–V143, 2006.

[19] N. Govindarajan, E. N. Epperly, and L. D. Lathauwer, "(l_r,l_r,1)-decompositions, sparse component analysis, and the blind separation of sums of exponentials," *SIAM J. Matrix Anal. Appl.*, vol. 43, no. 2, pp. 912–938, 2022.

[20] M. Ding, X. Fu, and X.-L. Zhao, "Fast and structured block-term tensor decomposition for hyperspectral unmixing," *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.*, vol. 16, pp. 1691–1709, 2023.

[21] F. Xiong, Y. Qian, J. Zhou, and Y. Y. Tang, "Hyperspectral unmixing via total variation regularized nonnegative tensor factorization," *IEEE Trans. Geosci. Remote Sens.*, vol. 57, no. 4, pp. 2341–2357, 2018.

[22] Y. Qian, F. Xiong, S. Zeng, J. Zhou, and Y. Y. Tang, "Matrix-vector nonnegative tensor factorization for blind unmixing of hyperspectral imagery," *IEEE Trans. Geosci. Remote Sensing*, vol. 55, no. 3, pp. 1776–1792, 2016.

[23] B. Hunyadi, D. Camps, L. Sorber, W. V. Paesschen, M. D. Vos, S. V. Huffel, and L. D. Lathauwer, "Block term decomposition for modelling epileptic seizures," *EURASIP J. Adv. Signal Process.*, vol. 2014, no. 1, pp. 1–19, 2014.

[24] P. M. R. de Oliveira, J. d. M. Goulart, C. A. R. Fernandes, and V. Zarzoso, "Blind source separation in persistent atrial fibrillation electrocardiograms using block-term tensor decomposition with lowner constraints," *IEEE J. Biomed. Health Inform.*, vol. 26, no. 4, pp. 1538–1548, 2022.

[25] J. H. d. M. Goulart, P. M. R. de Oliveira, R. C. Farias, V. Zarzoso, and P. Comon, "Alternating group lasso for block-term tensor decomposition and application to ECG source separation," *IEEE Trans. Signal Process.*, vol. 68, pp. 2682–2696, 2020.

[26] L. De Lathauwer and A. de Baynast, "Blind deconvolution of DS-CDMA signals by means of decomposition in rank-(1,L,L) terms," *IEEE Trans. Signal Process.*, vol. 56, no. 4, pp. 1562–1571, 2008.

[27] M. Sørensen, F. Van Eeghem, and L. De Lathauwer, "Blind multichannel deconvolution and convolutive extensions of canonical polyadic and block term decompositions," *IEEE Trans. Signal Process.*, vol. 65, no. 15, pp. 4132–4145, 2017.

[28] J.-F. Cardoso, "Blind signal separation: statistical principles," *Proc. IEEE*, vol. 86, no. 10, pp. 2009–2025, 1998.

[29] J.-F. Cardoso and A. Souloumiac, "Blind beamforming for non-Gaussian signals," *IEE Proc. F*, vol. 140, no. 6, pp. 362–370, 1993.

[30] 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, 1997.

[31] A. Ferreol, L. Albera, and P. Chevalier, "Fourth-order blind identification of underdetermined mixtures of sources (FOBIUM)," *IEEE Trans. Signal Process.*, vol. 53, no. 5, pp. 1640–1653, 2005.

[32] L. Albera, A. Ferréol, P. Chevalier, and P. Comon, "ICAR: a tool for blind source separation using fourth-order statistics only," *IEEE Trans. Signal Process.*, vol. 53, no. 10, pp. 3633–3643, 2005.

[33] L. T. Thanh, K. Abed-Meraim, P. Ravier, O. Buttelli, and A. Holobar, "Joint INDSCAL decomposition meets blind source separation," in *Proc. IEEE Int. Conf. Acoust. Speech Signal Process.*, 2024.

[34] ——, "Tensorial convolutive blind source separation," in *Proc. IEEE Int. Conf. Acoust. Speech Signal Process.*, 2024.

[35] J. D. Carroll and J.-J. Chang, "Analysis of individual differences in multidimensional scaling via an N-way generalization of "Eckart-Young" decomposition," *Psychometrika*, vol. 35, no. 3, pp. 283–319, 1970.

[36] I. Domanov and L. D. Lathauwer, "Generic uniqueness conditions for the canonical polyadic decomposition and INDSCAL," *SIAM J. Matrix Anal. Appl.*, vol. 36, no. 4, pp. 1567–1589, 2015.

[37] L. De Lathauwer, "Decompositions of a higher-order tensor in block terms – Part II: Definitions and uniqueness," *SIAM J. Matrix Anal. Appl.*, vol. 30, no. 3, pp. 1033–1066, 2008.

[38] L. De Lathauwer and D. Nion, "Decompositions of a higher-order tensor in block terms – Part III: Alternating least squares algorithms," *SIAM J. Matrix Anal. Appl.*, vol. 30, pp. 1067–1083, 2008.

[39] S. Boyd, N. Parikh, E. Chu, B. Peleato, and J. Eckstein, "Distributed optimization and statistical learning via the alternating direction method of multipliers," *Found. Trends Mach. Learn.*, vol. 3, no. 1, pp. 1–122, 2011.

[40] H. Bousbia-Salah, A. Belouchrani, and K. Abed-Meraim, "Jacobi-like algorithm for blind signal separation of convolutive mixtures," *Electr. Lett.*, vol. 37, no. 16, p. 1, 2001.

[41] K. Abed-Meraim, Y. Xiang, J. H. Manton, and Y. Hua, "Blind source-separation using second-order cyclostationary statistics," *IEEE Trans. Signal Process.*, vol. 49, no. 4, pp. 694–701, 2001.

[42] M. Kang, M. Kang, and M. Jung, "Inexact accelerated augmented Lagrangian methods," *Comput. Optim. Appl.*, vol. 62, pp. 373–404, 2015.

[43] R. Sameni and G. D. Clifford, "A review of fetal ECG signal processing; issues and promising directions," *Open Pacing Electrophysiol Ther J.*, vol. 3, p. 4, 2010.

[44] L. De Lathauwer, B. De Moor, and J. Vandewalle, "Fetal electrocardiogram extraction by blind source

subspace separation," *IEEE Trans. Biomed. Eng.*, vol. 47, no. 5, pp. 567–572, 2000.

[45] M. Rajih, P. Comon, and R. A. Harshman, "Enhanced line search: A novel method to accelerate PARAFAC," *SIAM J. Matrix Anal. Appl.*, vol. 30, no. 3, pp. 1128–1147, 2008.

[46] D. Nion and L. De Lathauwer, "An enhanced line search scheme for complex-valued tensor decompositions. application in DS-CDMA," *Signal Process.*, vol. 88, no. 3, pp. 749–755, 2008.

[47] D. Farina and A. Holobar, "Characterization of human motor units from surface EMG decomposition," *Proc. IEEE*, vol. 104, no. 2, pp. 353–373, 2016.

[48] L. T. Thanh, K. Abed-Meraim, N. L. Trung, and A. Hafiane, "Robust tensor tracking with missing data and outliers: Novel adaptive CP decomposition and convergence analysis," *IEEE Trans. Signal Process.*, vol. 70, pp. 4305 – 4320, 2022.

[49] H. Lyu, C. Strohmeier, and D. Needell, "Online nonnegative CP-dictionary learning for Markovian data," *J. Mach. Learn. Res.*, no. 23, pp. 1–50, 2022.