Fast and Structured Block-Term Tensor Decomposition For Hyperspectral Unmixing

Meng Ding, Xiao Fu, and Xi-Le Zhao

***Abstract*—The block-term tensor decomposition model with multilinear rank-$(L_r, L_r, 1)$ terms (or, the "LL1 tensor decomposition" in short) offers a valuable alternative formulation for *hyperspectral unmixing* (HU), which ensures identifiability of the endmembers/abudnaces in cases where classic matrix factorization (MF) approaches cannot provide such guarantees. However, existing LL1 tensor decomposition-based HU algorithms use a three-factor parameterization of the tensor (i.e., the hyperspectral image cube), which causes difficulties in incorporating structural prior information arising in HU. Consequently, their algorithms often exhibit high per-iteration complexity and slow convergence. This work focuses on LL1 tensor decomposition under structural constraints and regularization terms in HU. Our algorithm uses a two-factor re-parameterization of the tensor model. Like in the MF-based approaches, the factors correspond to the endmembers and abundances in the context of HU. Thus, the proposed framework is natural to incorporate physics-motivated priors in HU. To tackle the formulated optimization problem, a two-block alternating *gradient projection* (GP)-based algorithm is proposed. Carefully designed projection solvers are proposed to implement the GP algorithm with a relatively low per-iteration complexity. An extrapolation-based acceleration strategy is proposed to expedite the GP algorithm. Such extrapolated multi-block algorithm only had asymptotic convergence assurances in the literature. Our analysis shows that the algorithm converges to the vicinity of a stationary point within finite iterations, under reasonable conditions. Empirical study shows that the proposed algorithm often attains orders-of-magnitude speedup and substantial HU performance gains compared to the existing LL1 decomposition-based HU algorithms.**

***Index Terms*—Hyperspectral unmixing, structured block-term tensor decomposition, alternating gradient projection.**

I. Introduction

Remotely deployed hyperspectral sensors capture the reflected sunlight on the ground. The sensors then measure the spectra of received sunlight over a large number of wavelengths. The obtained pixels carry rich information about the ground materials. Hence, *hyperspectral images* (HSIs) are widely used in applications such as environment surveillance, agriculture, wild fire analysis, and mineral detection. However, hyperspectral sensors often have a limited spatial resolution [1]. Therefore, the spectral pixels in HSIs are usually mixtures of the spectral signatures of several different materials (i.e., endmembers). *Hyperspectral unmixing* (HU) aims at estimating the endmembers and their corresponding proportions (abundances) in the pixels [1].

HU is in essence a *blind source separation* (BSS) task. The *linear mixture model* (LMM) [1], [2] is the most commonly used BSS model for HU. Under the LMM, a spectral pixel of the HSI data is expressed as the convex combination of the endmember signatures, in which the combination coefficients represent the endmembers' abundances. The LMM is succinct and has proven effective for HU. In the past two decades, a large number of HU algorithms were developed under the LMM; see, e.g., [3]–[14]. In fact, numerical evidence shows that the LMM can oftentimes explain the vast majority of pixels with high accuracy; see, e.g., [15]. It is worth noting that many nonlinear mixture models are also proposed for HU; see [16] and recent developments using neural networks, e.g., [17]. Nonlinear models are used to capture the complex dynamics and data generating mechanisms that could not be interpreted by the LMM, which reduces modeling errors and enhances the HU performance. However, using nonlinear models are not without price—the computational problems under nonlinear models are in general much harder to tackle (especially when deep neural networks are involved); see, e.g., [17]. In fact, the LMM often allows us to design algorithms that strike a good balance between modeling accuracy and computational convenience. Hence, we focus on the more classic and more widely used LMM in this work.

One of the most important considerations in HU is the *identifiability* of the endmembers and the abundances. Under the LMM, the endmembers and abundances can be considered as the two "latent factors" of a matrix factorization (MF) model—which are non-identifiable in general. A remedy is to exploit structural prior information of the latent factors. For example, to establish model identifiability, an important line of work in HU uses the *convex geometry* (CG) of the abundances, e.g., the existence of the so-called "pure pixels" [3], [4], [12], [13], [18] or the "sufficiently scattered condition" [7], [19], [20]. CG-based identfibiaility analysis has also been used in many machine learning tasks; see [21].

The CG and MF based HU algorithms have enjoyed many successes. However, some challenges remain. In particular, the CG-based identfiability conditions require the existence of some special pixels, which may not always hold. Recently, a line of work in [22]–[28] proposed an alternative HU approach under the LMM. The work in [22] linked the so-called *block-term tensor decomposition model with multilinear rank* $(L_r, L_r, 1)$*-block terms* (or, in short, the LL1 tensor de-

The work of M. Ding is supported in part by the National Natural Science Foundation of China (12201522). The work of X. Fu is supported in part by the National Science Foundation (NSF) under Project ECCS 2024058. The work of X.-L. Zhao is supported in part by the National Natural Science Foundation of China (61876203). *(Corresponding author: Xiao Fu).*

M. Ding is with the School of Mathematics, Southwest Jiaotong University, Chengdu, China. e-mail: dingmeng56@163.com. X. Fu is with the School of Electrical Engineering and Computer Science, Oregon State University, Corvallis, OR 97331, United States. email: xiao.fu@oregonstate.edu. X.-L. Zhao is with the School of Mathematical Sciences, University of Electronic Science and Technology of China, Chengdu, China. e-mail: xlzhao122003@163.com.

composition model) [29]–[31] with LMM-based HU. Instead of treating the HU problem as an MF problem, the LL1-based approach takes a tensor decomposition perspective via exploiting the spatial dependence of the abundances. The LL1 model-based HU framework is refreshing. The approach offers complementary identifiability guarantees for HU in cases where CG-based methods cannot provide such assurances.

A. Challenges of Structured LL1 Model-Based HU

In the context of HU, the LL1 tensor decomposition formulations often have structural constraints and regularization on the latent factors, which usually come from physical meaning and prior information of hyperspectral images. However, designing *structured* LL1 tensor decomposition algorithms that are tailored for the HU problem is a nontrivial task.

The first challenge lies in efficiency. The speed of the existing algorithms are often unsatisfactory. One reason is that all the existing LL1-based HU algorithms (see [22]–[28]) adopted a three-factor parameterization of the LL1 tensor. Consequently, the classic *alternating least squares* (ALS) framework (see [30]) is employed as the backbone of their tensor decomposition algorithms. The nonnegativity constraints on the endmembers and abundances are often handled by the classic *multiplicative update* (MU) algorithm [32]. This ALS-MU combination often leads to a considerably high per-iteration complexity under typical HU settings. The MU algorithm is also prone to numerical issues in some cases, e.g., when there are iterates that contain zero elements [33]. In addition, MU typically needs a relatively large number of iterations to converge to a reasonable solution [34].

The second notable challenge is that the existing LL1-based HU algorithms have difficulties in enforcing many physically meaningful constraints on the abundances of the endmembers. The reason is that the abundance map of any endmember is represented by the product of two latent matrices under the three-block parameterization in existing LL1 based HU algorithms. This makes imposing prior information of the abundance maps (e.g., sparsity and spatial smoothness) fairly inconvenient and oftentimes leads to cumbersome reformulations, multiple slack variables and over-loaded tuning parameters (see, e.g., [23], [24], [26])—which may further complicate and slow down the algorithms.

B. Contributions

In [22]–[28], the lack of efficient LL1 decomposition algorithms under structural constraints/regularization stands as the computational bottleneck. In this work, we propose a *structured* LL1 tensor decomposition algorithm that is tailored for LMM-based HU. Our detailed contributions are as follows:

• **A Two-block Optimization Framework for Structured LL1 Tensor Decomposition-based HU.** We propose to employ the idea in [35] to recast the three-factor tensor decomposition problem into a rank-constrained two-factor MF formulation. Under the context of HU, the two factors explicitly represent the endmembers and their abundances—as in the classic MF approaches [2]. Hence, it is flexible and convenient to impose constraints and regularization based on their structural prior information (e.g., spatial smoothness of the abundances). Based on the reformulation, we develop an inexact and accelerated alternating *gradient projection* (GP) algorithm that admits substantially lower per-iteration complexity compared to the ALS-MU based structured LL1 approaches as in [22]–[28].

• **Fast Solvers for Structural Constraints.** A notable challenge of our two-block re-parameterization is that a number of complex constraints are imposed on the latent factors. This makes the *subproblems* in our two-block GP framework nontrivial to solve. In particular, the nonnegativity, sum-to-one, and low-rank constraints are all imposed on the abundance factor to reflect its physical properties. In addition, the low matrix rank constraint is added to the abundance maps under the tensor model. Simultaneously enforcing these constraints under our GP framework requires efficient and effective nonconvex set projection solvers. In this work, we propose two *alternating projection* (AP)-based algorithms to handle the projection problem of interest in a fast and accurate manner. This serves as a critical integrating component to flesh out the efficiency and effectiveness of the overall alternating GP-based LL1 decomposition algorithm.

• **Characterization and Validation.** Unlike the existing LL1-based HU algorithms (e.g., those in [22]–[28]) that often lack convergence understanding, we characterize the convergence behavior of the proposed two-block GP algorithmic framework. Note that *asymptotic* convergence of alternating two-block GP with acceleration was studied in [36]. Our analysis takes a step further and offers a finite iteration characterization. We test the proposed algorithm using a number of synthetic, semi-real, and real datasets under a variety of performance metrics. We compare our algorithms with a suite of existing ALS-MU over various datasets for numerical validation, and observe substantial efficiency and accuracy improvements attained by the proposed approach.

Part of the work appeared as a conference paper in [37]. In this journal version, we additionally include 1) a faster AP algorithm based on a convex approximation for the low-rank constraint, 2) the consideration of incorporating the total variation regularization on the abundance maps (to showcase the flexibility of the proposed framework), 3) convergence characterizations of the algorithm, and 4) extensive experiments on semi-real and real datasets.

Notation. The symbols x (or X), $\boldsymbol{x}$, $\boldsymbol{X}$, and $\underline{\boldsymbol{X}}$ denote the scalar, the vector, the matrix, and the tensor, respectively. The i-th, (i,j)-th, and (i,j,k)-th element of $\boldsymbol{x} \in \mathbb{R}^I$, $\boldsymbol{X} \in \mathbb{R}^{I\times J}$, and $\underline{\boldsymbol{X}} \in \mathbb{R}^{I\times J\times K}$ are represented by $[\boldsymbol{x}]_i$, $[\boldsymbol{X}]_{i,j}$, and $[\underline{\boldsymbol{X}}]_{i,j,k}$, respectively. The symbols $\boldsymbol{X}(i,:)$ and $\boldsymbol{X}(:,j)$ (or $\boldsymbol{x}_j$) denote the i-th row and j-th column of a matrix $\boldsymbol{X} \in \mathbb{R}^{I\times J}$, respectively. $\underline{\boldsymbol{X}}(i,j,:)$ denotes the (i,j)-th tube of $\underline{\boldsymbol{X}}$, and $\underline{\boldsymbol{X}}(:,:,k)$ denotes its k-th slab. $\|\boldsymbol{X}\|_F = \sqrt{\sum_{i,j}[\boldsymbol{X}]_{i,j}^2}$ and $\|\underline{\boldsymbol{X}}\|_F = \sqrt{\sum_{i,j,k}[\underline{\boldsymbol{X}}]_{i,j,k}^2}$ represent the Frobenius norms of $\boldsymbol{X}$ and $\underline{\boldsymbol{X}}$, respectively. Given a matrix $\boldsymbol{X} \in \mathbb{R}^{I\times J}$ and a vector $\boldsymbol{y} \in \mathbb{R}^K$, the outer product $\boldsymbol{X} \circ \boldsymbol{y}$ yields an $I \times J \times K$ tensor such that $[\boldsymbol{X} \circ \boldsymbol{y}]_{i,j,k} = \boldsymbol{X}(i,j)\boldsymbol{y}(k)$. The nuclear norm of $\boldsymbol{X}$ is denoted as the sum of singular values $\sigma_i(\boldsymbol{X})$, i.e., $\|\boldsymbol{X}\|_* = \sum_i \sigma_i(\boldsymbol{X})$. $\sigma_{\max}(\boldsymbol{X})$ denotes the largest singular

value of $\boldsymbol{X}$.

II. Preliminary

We briefly introduce the pertinent background of the LMM and the LL1 tensor decomposition model.

A. LMM-Based HU

Denote an HSI as $\underline{\boldsymbol{Y}} \in \mathbb{R}^{I\times J\times K}$, where I and J are the dimensions of the vertical and horizontal spatial modes, respectively, and K is the number of wavelengths. Here, $\underline{\boldsymbol{Y}}(:,:,k) \in \mathbb{R}^{I\times J}$ represents the $I\times J$ spatial image captured at the k-th wavelength. A pixel $\boldsymbol{y}_\ell := \underline{\boldsymbol{Y}}(i,j,:) \in \mathbb{R}^K$ is a K-dimensional vector with $\ell = i+(j-1)I$. Consider a noise-free case, under the LMM, a spectral pixel $\boldsymbol{y}_\ell$ is modeled as a convex combination of several endmembers contained in the pixel [2]. To be specific, we have

$$\boldsymbol{y}_\ell = \boldsymbol{C}\boldsymbol{s}_\ell, \tag{1}$$

where $\boldsymbol{c}_r \in \mathbb{R}^K$ for $r = 1,\ldots,R$ in $\boldsymbol{C} = [\boldsymbol{c}_1,\ldots,\boldsymbol{c}_R] \in \mathbb{R}^{K\times R}$ denote the R materials' spectral signatures (i.e., endmembers), and $\boldsymbol{s}_\ell \in \mathbb{R}^R$ is the corresponding abundance vector satisfying the following simplex constraint [1], [2]:

$$\mathbf{1}^\top \boldsymbol{s}_\ell = 1,\ \boldsymbol{s}_\ell \geq \mathbf{0},\ \ell = 1,\ldots,IJ. \tag{2}$$

The constraints stem from the physical interpretation of the LMM, where $s_{r,\ell}$ is the proportion of endmember r in the pixel ℓ.

Putting the pixels together to form a matrix, we have

$$\boldsymbol{Y} = \boldsymbol{C}\boldsymbol{S}, \tag{3}$$

where $\boldsymbol{Y} = [\boldsymbol{y}_1,\ldots,\boldsymbol{y}_{IJ}]$ obtained by setting $\boldsymbol{y}_\ell = \underline{\boldsymbol{Y}}(i,j,:)$, and $\boldsymbol{S} = [\boldsymbol{s}_1,\ldots,\boldsymbol{s}_{IJ}]$. The LMM in (3) can also be expressed using the following tensor notations:

$$\underline{\boldsymbol{Y}} = \sum_{r=1}^R \boldsymbol{S}_r \circ \boldsymbol{C}(:,r), \tag{4a}$$

$$\sum_{r=1}^R \boldsymbol{S}_r = \mathbf{1}\mathbf{1}^\top,\ \boldsymbol{S}_r \geq \mathbf{0}, \tag{4b}$$

where $\boldsymbol{C}(:,r) = \boldsymbol{c}_r \in \mathbb{R}^K$, $\mathbf{1}$ is an all-1 vector with a proper length, and $\circ$ denotes the outer product. The matrix $\boldsymbol{S}_r \in \mathbb{R}^{I\times J}$is obtained by reshaping the row vector $\boldsymbol{S}(r,:) \in \mathbb{R}^{IJ}$; specifically, we have

$$\boldsymbol{S}(r,:) = \mathrm{vec}(\boldsymbol{S}_r)^\top.$$

The matrix $\boldsymbol{S}_r$ can be interpreted as the *abundance map* of the r-th endmember in the context of HU; see Fig. 1. LMM-based HU aims at finding $\boldsymbol{S}_r$ for $r = 1,\ldots,R$ (or, equivalently, the matrix $\boldsymbol{S}$) and $\boldsymbol{C}$ simultaneously.

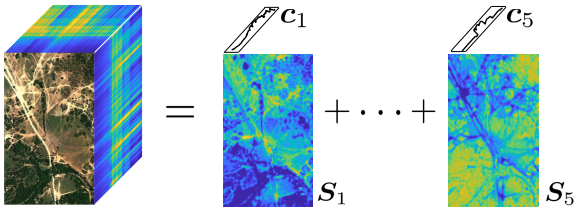


Fig. 1. Illustration of the LMM with $R = 5$ endmembers.

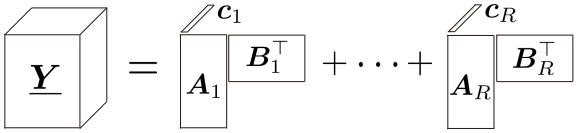


Fig. 2. Illustration of the LL1 model.

B. CG-based MF, Identifiability

Under the matrix factorization model in (3), a large number of MF-based methods have been proposed for HU; see the overviews in [1], [2]. As a BSS problem, the effectiveness of these MF-based HU methods heavily depends on the *identifiability* of $\boldsymbol{C}$ and $\boldsymbol{S}$ from $\boldsymbol{Y}$. Generally speaking, without any constraints on the factors $\boldsymbol{C}$ and $\boldsymbol{S}$, the MF model in (3) is not identifiable, even without noise, as one can easily find an invertible $\boldsymbol{Q}$ such that

$$\widehat{\boldsymbol{C}} = \boldsymbol{C}\boldsymbol{Q} \geq \mathbf{0},\ \widehat{\boldsymbol{S}} = \boldsymbol{Q}^{-1}\boldsymbol{S} \geq \mathbf{0},$$

but $\boldsymbol{Y} = \widehat{\boldsymbol{C}}\widehat{\boldsymbol{S}}$ still holds; see more discussions on the identifiability issues in [2], [20], [21].

The identifiability problem has been studied extensively, primarily from a CG-based simplex-structured MF (SSMF) viewpoint [21]. In a nutshell, it has been established that if the abundance matrix $\boldsymbol{S}$ satisfies certain geometric conditions, namely, the *pure pixel condition* [3], [4], [12], [13] and the *sufficiently scattered condition* [7], [20], [21], then $\boldsymbol{C}$ and $\boldsymbol{S}$ can be identified up to column and row permutations, respectively. These important results reflect the long postulations in the HU community, i.e., the Winter's and Craig's beliefs [9], [11]. Nonetheless, despite of the elegance of SSMF's identifiability research, these geometric conditions can still be stringent in some cases, as they both assume the existence of some special pixels. Hence, these conditions may not always hold in real data; see, e.g., the "highly mixed cases" in [38].

C. LL1 Tensor Decomposition-Based HU

The work in [22] proposed a tensor decomposition method for HU under the LMM. The employed LL1 model is similar to the tensor expression of LMM in (4a)—but has an extra low-rank assumption on the abundance maps. To be specific, assume that each abundance map $\boldsymbol{S}_r$ is a low-rank matrix such that $\mathrm{rank}(\boldsymbol{S}_r) = L_r \leq \min\{I,J\}$, then the expression in (4a) can be re-written as follows:

$$\underline{\boldsymbol{Y}} = \sum_{r=1}^R \left(\boldsymbol{A}_r\boldsymbol{B}_r^\top\right) \circ \boldsymbol{C}(:,r), \tag{5}$$

where $\boldsymbol{A}_r \in \mathbb{R}^{I\times L_r}$, $\boldsymbol{B}_r \in \mathbb{R}^{J\times L_r}$, and $\boldsymbol{S}_r = \boldsymbol{A}_r\boldsymbol{B}_r^\top$. The model in (5) is the LL1 tensor model [29], which is illustrated

in Fig. 2. It admits an important identifiability property [29] as follows:

Theorem 1 (Identifiability of LL1)**.** *Assume that the latent factors $(\boldsymbol{A}_r, \boldsymbol{B}_r, \boldsymbol{C})$ in* (5) *are drawn from any joint absolutely continuous distributions. Assume $L_r = L$, $IJ \geq L^2 R$, and*

$$\min\left(\left\lfloor \frac{I}{L} \right\rfloor, R\right) + \min\left(\left\lfloor \frac{J}{L} \right\rfloor, R\right) + \min(K, R) \geq 2R + 2.$$

Then, the LL1 *decomposition of $\underline{\boldsymbol{Y}}$ is essentially unique almost surely.*

The "essential uniqueness" means that if there exists $(\bar{\boldsymbol{A}}_r, \bar{\boldsymbol{B}}_r, \bar{\boldsymbol{C}})$ satisfying $\underline{\boldsymbol{Y}} = \sum_{r=1}^{R} (\bar{\boldsymbol{A}}_r (\bar{\boldsymbol{B}}_r)^\top) \circ \bar{\boldsymbol{C}}(:, r)$, we must have $\bar{\boldsymbol{S}} = \boldsymbol{S}\boldsymbol{\Pi}\boldsymbol{\Lambda}$, $\bar{\boldsymbol{C}} = \boldsymbol{C}\boldsymbol{\Pi}\boldsymbol{\Lambda}^{-1}$, where $\boldsymbol{\Pi}$ is a permutation matrix, $\boldsymbol{\Lambda}$ is a nonsingular diagonal matrix, $\bar{\boldsymbol{S}} = [\text{vec}(\bar{\boldsymbol{S}}_1), \ldots, \text{vec}(\bar{\boldsymbol{S}}_R)]^\top$, $\bar{\boldsymbol{S}}_r = \bar{\boldsymbol{A}}_r \left(\bar{\boldsymbol{B}}_r\right)^\top$, $\boldsymbol{S} = [\text{vec}(\boldsymbol{S}_1), \ldots, \text{vec}(\boldsymbol{S}_R)]^\top$, $\boldsymbol{S}_r = \boldsymbol{A}_r \boldsymbol{B}_r^\top$, and the vec$(\cdot)$ denotes the "vectorization" operator. Note that if $\boldsymbol{S}$ has known column norms as in hyperspectral imaging [cf. Eq. (2)], then the scaling ambiguity $\boldsymbol{\Lambda}$ is automatically removed.

Theorem 1 asserts that the abundance maps ($\boldsymbol{S}_r$'s) and the endmembers ($\boldsymbol{C}$) are identifiable up to a permutation ambiguity if the abundance maps have a relatively low rank. In the context of HU, because of the smoothness and continuity of the materials' spread over the spatial domain, the abundance maps are often approximately low rank matrices; see an example in Sec. V-B. The identifiability conditions in Theorem 1 are different from those geometric conditions (e.g., the pure pixel condition and the sufficiently scattered condition) used in CG-based SSMF [21], and thus the LL1-based approach is a valuable complement to existing SSMF-based HU methods. In addition, the identifiability under the LL1 decomposition model can even hold when $\boldsymbol{C}$ does not have linearly independent columns, but $\text{rank}(\boldsymbol{C}) = R$ is often needed in SSMF [2], [21].

D. Challenges of Existing LL1*-based HU Algorithms*

Directly applying the vanilla LL1 tensor model to LMM-based HU without considering important physical constraints in HU (e.g., nonnegativity of the endmembers) may be undesirable—as using such priors are often vital when fending against noise. The work in [22] proposed the following criterion:

$$\begin{aligned} \min_{\{\boldsymbol{A}_r, \boldsymbol{B}_r\}, \boldsymbol{C}} \; & \frac{1}{2} \left\| \underline{\boldsymbol{Y}} - \sum_{r=1}^{R} \left(\boldsymbol{A}_r \boldsymbol{B}_r^\top\right) \circ \boldsymbol{C}(:, r) \right\|_F^2 \\ & + \frac{\delta}{2} \left\| \sum_{r=1}^{R} \boldsymbol{A}_r \boldsymbol{B}_r^\top - \mathbf{1}\mathbf{1}^\top \right\|_F^2, \\ \text{s.t. } & \boldsymbol{A}_r \geq \mathbf{0}, \; \boldsymbol{B}_r \geq \mathbf{0}, \; \boldsymbol{C} \geq \mathbf{0}. \end{aligned} \tag{6}$$

The nonnegativity constraints on the latent factors are added per the endmembers' and abundances' physical meaning. The second penalty term in the objective function is for approximating the abundance sum-to-one constraint, i.e., $\sum_{r=1}^{R} \boldsymbol{S}_r = \mathbf{1}\mathbf{1}^\top$ in (4). Similar formulations are also used in a number of follow-up works; see, e.g., [23]–[28]. This line of work encounters a number of challenges:

• **High Per-iteration Complexity.** The work in [22] and its variants in [23]–[28] adopt the ALS-MU algorithms, i.e., alternately update $\boldsymbol{A} = [\boldsymbol{A}_1, \ldots, \boldsymbol{A}_R]$, $\boldsymbol{B} = [\boldsymbol{B}_1, \ldots, \boldsymbol{B}_R]$ and $\boldsymbol{C}$ using matrix unfoldings of $\underline{\boldsymbol{Y}}$, and use MU to enforce the non-negative constraints. The first challenge of ALS-MU lies in computational complexity. Because of using $\boldsymbol{A} \in \mathbb{R}^{I \times LR}$, $\boldsymbol{B} \in \mathbb{R}^{J \times LR}$ and $\boldsymbol{C} \in \mathbb{R}^{K \times R}$ as factors of the parameterization and the ALS framework, the ALS-MU algorithm costs $\mathcal{O}(IJKLR + IKL^2R^2 + JKL^2R^2)$ flops at each iteration—which is fairly expensive since LR can easily reach the level of $10^2 \sim 10^3$ in many cases of HU. This scheme essentially treats the LL1 decomposition problem as a *canonical polyadic decomposition* problem [39], [40] with a *tensor rank* of LR, which is very hard when LR is large.

• **Slow Convergence and Numerical Issues of MU.** All the algorithms in [22]–[28] employed the MU algorithm for handling nonnegativity constraints on $\boldsymbol{A}, \boldsymbol{B}$ and $\boldsymbol{C}$. Essentially, MU updates one factor using the majorization minimization method but with a very conservative step-size such that the nonnegativity is satisfied; see [32]. Thus, MU often takes a large number of iterations to attain a sensible result and perhaps worsens the efficiency [34]. In addition, as shown in [33], the MU algorithm is prone to numerical issue if there is a zero element in any iterates of $\boldsymbol{A}, \boldsymbol{B}$ or $\boldsymbol{C}$. This is problematic in the context of HU, as $\boldsymbol{S}_r = \boldsymbol{A}_r \boldsymbol{B}_r^\top$ may contain many zeros due to the spatial sparsity of the abundance maps

• **Difficulty in Incorporating More Priors.** The existing LL1-based HU methods adopted the three-factor parameterization using $\boldsymbol{A} = [\boldsymbol{A}_1, \ldots, \boldsymbol{A}_R]$, $\boldsymbol{B} = [\boldsymbol{B}_1, \ldots, \boldsymbol{B}_R]$ and $\boldsymbol{C}$ as in (5). However, the parameters $\boldsymbol{A}_r$ and $\boldsymbol{B}_r$ do not have physical meaning, and the abundance map of material r is represented as $\boldsymbol{A}_r \boldsymbol{B}_r^\top$, i.e., the product of two matrices. This introduces extra difficulties in incorporating prior information of the abundance maps—but using prior information is often critical for performance enhancement. The works in [22]–[28] used the following optimization criterion (and some variants):

$$\begin{aligned} \min_{\{\boldsymbol{A}_r, \boldsymbol{B}_r\}, \boldsymbol{C}} \; & \frac{1}{2} \left\| \underline{\boldsymbol{Y}} - \sum_{r=1}^{R} \left(\boldsymbol{A}_r \boldsymbol{B}_r^\top\right) \circ \boldsymbol{C}(:, r) \right\|_F^2 \\ & + \frac{\delta}{2} \left\| \sum_{r=1}^{R} \boldsymbol{A}_r \boldsymbol{B}_r^\top - \mathbf{1}\mathbf{1}^\top \right\|_F^2 + \lambda \sum_{r=1}^{R} \text{reg}\left(\boldsymbol{A}_r \boldsymbol{B}_r^\top\right) \\ \text{s.t. } & \boldsymbol{A}_r \geq \mathbf{0}, \; \boldsymbol{B}_r \geq \mathbf{0}, \; \boldsymbol{C} \geq \mathbf{0}, \end{aligned} \tag{7}$$

where $\text{reg}(\boldsymbol{A}_r \boldsymbol{B}_r)$ represents different regularization terms, e.g., sparsity $\|\boldsymbol{A}_r \boldsymbol{B}_r^\top\|_1$ and low-rank $\|\boldsymbol{A}_r \boldsymbol{B}_r^\top\|_*$ in [27], total variation $(\boldsymbol{A}_r \boldsymbol{B}_r^\top)_{\text{TV}}$ in [23], and the weighted sparsity regularization terms in [24], [25]. From an optimization perspective, handling the term $\text{reg}\left(\boldsymbol{A}_r \boldsymbol{B}_r^\top\right)$ is nontrivial. Hence,

the work in [23] further recasts the problem in (7) as follows:

$$\begin{aligned}\min_{\substack{\boldsymbol{A},\boldsymbol{B},\boldsymbol{C}\\ \{\boldsymbol{E}_r\},\boldsymbol{U},\boldsymbol{V}}}\ & \frac{1}{2}\left\|\underline{\boldsymbol{Y}}-\sum_{r=1}^{R}\left(\boldsymbol{A}_r\boldsymbol{B}_r^\top\right)\circ\boldsymbol{C}(:,r)\right\|_F^2 \\ &+\frac{\delta}{2}\left\|\sum_{r=1}^{R}\boldsymbol{A}_r\boldsymbol{B}_r^\top-\boldsymbol{1}\boldsymbol{1}^\top\right\|_F^2+\lambda\sum_{r=1}^{R}\mathrm{reg}\left(\boldsymbol{E}_r^\top\right)\\ &+\frac{\omega}{2}\left(\left\|\boldsymbol{E}_r-\boldsymbol{U}_r\boldsymbol{V}_r^\top\right\|_F^2+\|\boldsymbol{U}-\boldsymbol{A}\|_F^2+\|\boldsymbol{V}-\boldsymbol{B}\|_F^2\right)\\ &\text{s.t. } \boldsymbol{A}_r\geq\boldsymbol{0},\ \boldsymbol{B}_r\geq\boldsymbol{0},\ \boldsymbol{C}\geq\boldsymbol{0},\end{aligned}\tag{8}$$

where $\boldsymbol{E}_r$ is introduced to replace the product $\boldsymbol{A}_r\boldsymbol{B}_r^\top$, i.e., $\boldsymbol{E}_r\approx\boldsymbol{A}_r\boldsymbol{B}_r^\top$ is desired, so that the regularization on the abundances could be handled. In addition, the $\boldsymbol{U}$ and $\boldsymbol{V}$ variables are introduced to avoid constraints when optimizing $\boldsymbol{E}_r$. A number of works, e.g., [26]–[28], used similar ideas to reformulate their respective problems. Such reformulations make sense, but the many added extra regularization terms, new auxiliary variables and hyperparameters make the optimization procedure and parameter tuning even more complicated. Furthermore, the backbone of the algorithm is still ALS-MU, which means that these algorithms share the high per-iteration complexity and slow convergence challenges as the plain-vanilla ALS-MU algorithm in [22].

III. Structured LL1 Decomposition for HU

To tackle the aforementioned challenges in Sec. II-D, this work aims at providing an alternative LL1-based HU algorithmic framework. To be specific, we propose to employ a constrained two-block parameterization of the LL1 tensor model. Under this parameterization, a two-block alternating optimization strategy is proposed. The new algorithm empirically exhibits a much faster convergence speed, perhaps because two-block alternating optimization often converges under milder conditions relative to the multi-block counterparts [41]. More importantly, the two-block structure circumvents the large-scale subproblems under the three-block based ALS frameworks in prior works. The new formulation also admits the flexibility to add regularization and structural constraints on the abundance maps. Moreover, the new formulation allows us to propose efficient alternating gradient projection algorithms to deal with the block optimization subproblems—which further improves effieciency and accuracy.

A. LL1 *via Constrained Matrix Factorization*

Our idea starts from a two-block parameterization of the LL1 model in (5). To be specific, the following equivalence is readily seen:

$$\begin{aligned}&\{\boldsymbol{S}_r\in\mathbb{R}^{I\times J}|\boldsymbol{S}_r=\boldsymbol{A}_r\boldsymbol{B}_r^\top,\ \boldsymbol{A}_r\in\mathbb{R}^{I\times L},\boldsymbol{B}_r\in\mathbb{R}^{J\times L}\}\\ &=\{\boldsymbol{S}_r\in\mathbb{R}^{I\times J}|\mathrm{rank}(\boldsymbol{S}_r)\leq L\}.\end{aligned}\tag{9}$$

The above equivalence allows us to re-express the LL1 model in (5) as follows [35]:

$$\underline{\boldsymbol{Y}}=\sum_{r=1}^{R}\boldsymbol{S}_r\circ\boldsymbol{c}_r,\ \mathrm{rank}(\boldsymbol{S}_r)\leq L.\tag{10}$$

By (9), the two-block tensor representation in (10) is equivalent to the three-block representation in (5). Using (10), we propose the following criterion for LL1-based HU:

$$\min_{\boldsymbol{S},\boldsymbol{C}}\ \frac{1}{2}\|\boldsymbol{Y}-\boldsymbol{C}\boldsymbol{S}\|_F^2+\sum_{r=1}^{R}\theta_r\varphi(\boldsymbol{S}_r)\tag{11a}$$

$$\text{s.t. } \boldsymbol{S}_r\in\mathcal{A}_{\mathrm{LR}},\ r=1,\ldots,R,\tag{11b}$$

$$\boldsymbol{S}\geq\boldsymbol{0},\ \boldsymbol{1}^\top\boldsymbol{S}=\boldsymbol{1}^\top,\ \boldsymbol{C}\geq\boldsymbol{0},\tag{11c}$$

where $\boldsymbol{S}_r=\mathrm{mat}(\boldsymbol{S}(r,:))$ is the abundance map of endmember r, $\mathrm{mat}(\cdot)$ denotes the "matricization" operator that reshapes the row vector $\boldsymbol{S}(r,:)$ to an $I\times J$ matrix, $\theta_r\geq 0$ is a regularization parameter, $\varphi(\cdot)$ is a regularization term added on the abundance map (e.g., sparsity, weighted sparsity, and total variation as used in [22]–[28]), and the set $\mathcal{A}_{\mathrm{LR}}$ is for adding a low-rank (or approximate low-rank) constraint onto $\boldsymbol{S}_r$, which will be specified later.

The motivation for using this reformulation is as follows. First, using the two-factor representation of LL1 model can effectively avoid large-size subproblems as in the ALS framework (in particular, the subproblems for updating $\boldsymbol{A}$ and $\boldsymbol{B}$), and thus could substantially reduce the complexity of each iteration. Second, we have replaced $\boldsymbol{A}_r\boldsymbol{B}_r^\top$ by $\boldsymbol{S}_r$ under the low-rank constraint, that makes it conveniently to add the prior information on $\boldsymbol{S}_r$ (the abundance map). Third, the works in [22]–[28] used a quadratic approximation to enforce sum-to-one constraint, i.e., $\delta/2\|\sum_{r=1}^{R}\boldsymbol{A}_r\boldsymbol{B}_r^\top-\boldsymbol{1}\boldsymbol{1}^\top\|_F^2$, which does not necessarily output abundance maps that satisfy the sum-to-one constraint. We keep the sum-to-one requirement as a hard constraint, which spares the tuning of δ and can always have the sum-to-one property satisfied. In addition, compared to reformulations like (8), our formulation has avoided using auxiliary variables and additional tuning parameters, which may be easier to implement by practitioners.

Remark 1. The formulation in (11) can be understood as a constrained matrix factorization-based *re-expression* of the LL1 tensor decomposition problem. We stress that the reformulation is the means for realizing the LL1 tensor decomposition objective (i.e., to find the factorization model in (5)), but the theoretical foundation for using this formulation still lies in the identifiability of LL1 tensor decomposition model (cf. Theorem 1). The key ingredient for linking (11) with (5) is the low rank constraint on $\boldsymbol{S}_r$.

B. Proposed Approach: Alternating Gradient Projection

In this section, we propose a first-order optimization algorithm to handle (11). To demonstrate its convenience of incorporating structural constraints, we consider a 2D spatial TV regularizer (see the definition in (16) and details in Sec. III-B1) to exploit the spatial similarity between the neighboring abundance pixels.

Our idea is to deal with $\boldsymbol{C}$ and $\boldsymbol{S}$ in an alternating manner. In iteration t, we use the following gradient projection (GP) step to update $\boldsymbol{C}$:

$$\boldsymbol{C}^{(t+1)}\leftarrow\max\left\{\boldsymbol{C}^{(t)}-\alpha^{(t)}\boldsymbol{G}_{\boldsymbol{C}}^{(t)},\ \boldsymbol{0}\right\},\tag{12}$$

where $\max\{\cdot,\ \mathbf{0}\}$ is the orthogonal projector onto the nonnegativity orthant, $\alpha^{(t)}$ and $\boldsymbol{G}_{\boldsymbol{C}}^{(t)}$ are the pre-designed step size and gradient w.r.t. $\boldsymbol{C}$, respectively, where we have

$$\boldsymbol{G}_{\boldsymbol{C}}^{(t)} = \boldsymbol{C}^{(t)}\boldsymbol{S}^{(t)}(\boldsymbol{S}^{(t)})^\top - \boldsymbol{Y}(\boldsymbol{S}^{(t)})^\top. \tag{13}$$

For the $\boldsymbol{S}$-subproblem, assume that $\varphi(\cdot)$ is differentiable. Then, the GP step w.r.t., $\boldsymbol{S}$ can be expressed as follows:

$$\boldsymbol{S}^{(t+1)} \leftarrow \mathsf{Proj}_{\mathcal{S}}\left(\boldsymbol{S}^{(t)} - \beta^{(t)}\boldsymbol{G}_{\boldsymbol{S}}^{(t)}\right), \tag{14}$$

where the notation $\mathsf{Proj}_{\mathcal{S}}(\boldsymbol{Q})$ means finding the projection of $\boldsymbol{Q}$ on the set $\mathcal{S}$, $\beta^{(t)}$ and $\boldsymbol{G}_{\boldsymbol{S}}^{(t)}$ (shown in Appendix A) are the step size and the gradient w.r.t. $\boldsymbol{S}$, respectively, and the set $\mathcal{S} \subseteq \mathbb{R}^{R\times IJ}$ is defined as

$$\mathcal{S} = \{\boldsymbol{S} | \boldsymbol{S} \geq \mathbf{0}, \mathbf{1}^\top\boldsymbol{S} = \mathbf{1}^\top, \boldsymbol{S}_r \in \mathcal{A}_{\rm LR},\ r = 1,\ldots,R\}. \tag{15}$$

The GP step in (14) is conceptually simple. However, to implement (14), there are two challenges that need to be carefully addressed. First, the TV regularization $\varphi(\cdot)$ should be designed to have a gradient. Second, the projection onto $\mathcal{A}_{\rm LR}$ should be easy to compute. Our designs are detailed in the following:

1) ℓ_q Function-Based TV Regularization: To address the first challenge, we employ the ℓ_q function-based smoothed TV regularization in [42], which is defined as follows:

$$\varphi(\boldsymbol{S}_r) = \varphi_{q,\varepsilon}(\boldsymbol{H}_x\boldsymbol{q}_r) + \varphi_{q,\varepsilon}(\boldsymbol{H}_y\boldsymbol{q}_r), \tag{16}$$

where $\boldsymbol{q}_r = \boldsymbol{S}(r,:)^\top$ and $\varphi_{q,\varepsilon}(\boldsymbol{x}) = \sum([\boldsymbol{x}]_i^2 + \varepsilon)^{\frac{q}{2}}$ with $0 < q \leq 1$ and $\varepsilon > 0$. The matrices $\boldsymbol{H}_x = \boldsymbol{H}\otimes\boldsymbol{I}$ and $\boldsymbol{H}_y = \boldsymbol{I}\otimes\boldsymbol{H}$ are the horizontal and vertical gradient matrices, where $\boldsymbol{I} \in \mathbb{R}^{J\times J}$ is an identity matrix and

$$\boldsymbol{H} = \begin{bmatrix} 1 & -1 & 0 & \cdots & 0 & 0 \\ 0 & 1 & -1 & \cdots & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & \cdots & 0 & 1 & -1 \\ -1 & 0 & \cdots & 0 & 0 & 1 \end{bmatrix} \in \mathbb{R}^{I\times I}.$$

In the design of $\varphi(\boldsymbol{S}_r)$ function, the ℓ_q function is an effective tool to promote the sparsity when $q < 2$ [7], [8]. Meanwhile, we set the parameter $\varepsilon > 0$ to make the function smooth and the objective function in (11) continuously differentiable.

2) Low-rank Constraints: Some options of (11b) include

$$\mathcal{A}_{\rm LR} = \{\boldsymbol{S}_r \in \mathbb{R}^{I\times J} | \mathrm{rank}(\boldsymbol{S}_r) \leq L\} \tag{17}$$

and the nuclear norm-based approximation

$$\mathcal{A}_{\rm LR} = \{\boldsymbol{S}_r \in \mathbb{R}^{I\times J} | \|\boldsymbol{S}_r\|_* \leq \widetilde{L}\}, \tag{18}$$

where $\widetilde{L} \in \mathbb{R}_{++}$ is a tunable parameter related to the rank of $\boldsymbol{S}_r$, as the value of the nuclear norm is not exactly the rank. The expression in (17) is an exact low-rank constraint as in the LL1 model, but it presents a nonconvex combinitorial constraint in the criterion in (11). The latter is often used in data analytics (e.g., recommender systems) to promote low rank. It serves as a convex approximation for the low-rank constraint and often helps design convergence guaranteed algorithms. In this work, we will design an algorithmic framework that can effectively work with both (17) and (18).

3) Projection Algorithm for (14)*:* To solve (14), one needs a solver to project a matrix onto the set

$$\mathcal{S} = \mathcal{A}_{\rm splx} \cap \mathcal{A}_{\rm LR}.$$

where $\mathcal{A}_{\rm splx} := \{\boldsymbol{S} \in \mathbb{R}^{R\times IJ} \mid \mathbf{1}^\top\boldsymbol{S} = \mathbf{1}^\top,\ \boldsymbol{S} \geq \mathbf{0}\}$. To this end, we propose an *alternating projection* (AP)-based method. More specifically, the projection uses the following iterations:

$$\boldsymbol{F}^{(k+1)} \leftarrow \mathsf{Proj}_{\mathcal{A}_{\rm LR}}\left(\boldsymbol{W}^{(k)}\right), \tag{19a}$$

$$\boldsymbol{W}^{(k+1)} \leftarrow \mathsf{Proj}_{\mathcal{A}_{\rm splx}}\left(\boldsymbol{F}^{(k+1)}\right), \tag{19b}$$

where $\boldsymbol{W}^{(0)} = \boldsymbol{S}^{(t)} - \beta^{(t)}\boldsymbol{G}_{\boldsymbol{S}}^{(t)}$ and k is used as the iteration index of the AP updates. The second subproblem, i.e., (19b), admits an efficient solver. That is, projecting a column of $\boldsymbol{F}^{(k+1)}$ onto the probability simplex can be solved in $\mathcal{O}(R\log R)$ flops in the worst case by a water-filling type algorithm; see [7] and the references therein. The subproblem in (19a) also admits simple solutions. To be specific,

1) if (17) is used, then projecting $\boldsymbol{W}^{(k)}$ onto the exact low-rank constraint is computed via
$$\boldsymbol{F}^{(k+1)} = \boldsymbol{U}_{\boldsymbol{W}}^{(k)}(:,1:L)\boldsymbol{\Sigma}_{\boldsymbol{W}}^{(k)}(1:L,1:L)\boldsymbol{V}_{\boldsymbol{W}}^{(k)}(:,1:L)^\top, \tag{20}$$
where $(\boldsymbol{U}_{\boldsymbol{W}}^{(k)}, \boldsymbol{\Sigma}_{\boldsymbol{W}}^{(k)}, \boldsymbol{V}_{\boldsymbol{W}}^{(k)}) \leftarrow \mathrm{svd}(\boldsymbol{W}^{(k)})$, which is based on the Eckart–Young–Mirsky theorem [43]; and
2) if (18) is used, then the projection is computed by
$$\boldsymbol{F}^{(k+1)} = \boldsymbol{U}_{\boldsymbol{W}}^{(k)}\widetilde{\boldsymbol{\Sigma}}_{\boldsymbol{W}}^{(k)}(\boldsymbol{V}_{\boldsymbol{W}}^{(k)})^\top, \tag{21}$$
where $(\boldsymbol{U}_{\boldsymbol{W}}^{(k)}, \Sigma_{\boldsymbol{W}}^{(k)}, \boldsymbol{V}_{\boldsymbol{W}}^{(k)}) \leftarrow \mathrm{svd}(\boldsymbol{W}^{(k)})$ and
$$\widetilde{\boldsymbol{\Sigma}}_{\boldsymbol{W}}^{(k)} \leftarrow \underset{\widetilde{\boldsymbol{\Sigma}}\geq\mathbf{0},\mathbf{1}^\top\mathrm{diag}(\widetilde{\boldsymbol{\Sigma}})=\widetilde{L}}{\mathrm{argmin}} \|\widetilde{\boldsymbol{\Sigma}} - \boldsymbol{\Sigma}_{\boldsymbol{W}}^{(k)}\|_F^2, \tag{22}$$
which is again a projection onto simplex problem that costs at most $\mathcal{O}(\min\{I,J\}\log\min\{I,J\})$ flops (if $I \leq J$)—see more discussions in [44].

Note that only relatively simple matrix operations are involved in the above procedure. Hence, when using both (17) and (18), the AP algorithm in (19) can be carried out efficiently.

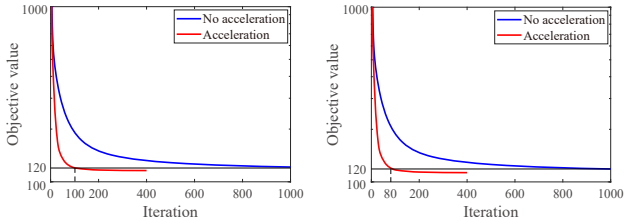
C. Extrapolation-Based Acceleration

As a first-order optimization algorithm, the proposed alternating gradient projection algorithm in (12)-(14) may take many iterations to get a reasonably "good" result in practice. Hence, in our implementation, the extrapolation technique is employed to accelerate the proposed algorithm without increasing the complexity of each iteration. To be specific, in each iteration, we compute the partial gradients w.r.t. some extrapolated points $\check{\boldsymbol{C}}^{(t+1)}$ and $\check{\boldsymbol{S}}^{(t+1)}$, instead of the gradients of $\boldsymbol{C}^{(t+1)}$ and $\boldsymbol{S}^{(t+1)}$. For example, the extrapolated point of $\boldsymbol{C}^{(t)}$ is defined as follows:

$$\check{\boldsymbol{C}}^{(t+1)} = \boldsymbol{C}^{(t+1)} + \mu_1^{(t)}(\boldsymbol{C}^{(t+1)} - \boldsymbol{C}^{(t)}), \tag{23}$$

where $\mu_1^{(t)}$ is a parameter that combines the the current iterate and the previous iterate to form an "extrapolation". Using the extrapolated point, the update rule in (12) is replaced by

$$\boldsymbol{C}^{(t+1)} \leftarrow \max\left\{\check{\boldsymbol{C}}^{(t)} - \alpha^{(t)}\boldsymbol{G}_{\check{\boldsymbol{C}}}^{(t)},\ \mathbf{0}\right\}. \tag{24}$$



(a) low-rank constraint (17) (b) low-rank constraint (18)

Fig. 3. The acceleration attained by extrapolation of GradPAPA.

Similarly, the $\boldsymbol{S}$-update is replaced by

$$\boldsymbol{S}^{(t+1)} \leftarrow \mathsf{Proj}_{\mathcal{S}}\left(\check{\boldsymbol{S}}^{(t)} - \beta^{(t)}\boldsymbol{G}_{\check{\boldsymbol{S}}}^{(t)}\right), \tag{25}$$

where $\check{\boldsymbol{S}}^{(t)}$ is defined in the same way as in (23) with its own sequence $\mu_2^{(t)}$. The extrapolation technique has been proven powerful in first-order convex optimization. Algorithms like gradient projection and proximal gradient typically need t iterations to reach an $\mathcal{O}(1/t)$-optimal solution (i.e., a solution that is $\mathcal{O}(1/t)$ away from the optimal solution by some distance metric) in the absence of strong convexity. The extrapolation can provably reach $\mathcal{O}(1/t^2)$-optimal solutions with the same number of iterations—yet the additional flops are nearly negligible; see [45]. For nonconvex and multiblock problems, extrapolation was also shown useful [46].

Fig. 3 compares the objective value curves of the original alternating GP algorithm and the accelerated one. The results are obtained by averaging from 10 random trials, and the experiment aims at unmixing an HSI (size $500 \times 307 \times 166$) with 40 dB additive Gaussian noise; please find more details in Section IV. The noise at each trial is generated randomly, and the corresponding initialization is obtained by applying successive projection algorithm [47] on the observed HSI. One can see that the accelerated algorithm takes about 100 iterations to get a fairly low objective value (i.e., where the objective value $= 120$), while the unaccelerated version uses more than 800 iterations to reach the same level. Therefore, throughout the experiment section, we adopt the accelerated version.

The proposed algorithm for (11) is summarized in Algorithm 1, which is referred to as the *gradient projection alternating projection algorithm* (GradPAPA). The two versions of algorithm for handling low-rank (LR) and nuclear norm (NN) constraints in (17) and (18) are termed as GradPAPA-LR and GradPAPA-NN, respectively.

D. Convergence Properties

Unlike the ALS-MU algorithms in [22]–[28] that may have convergence issues, the proposed GradPAPA algorithm's convergence properties are better understood. Indeed, the GradPAPA algorithm falls under the umbrella of *inexact and extrapolated block coordinate descent* (BCD) [46], [48]. The works in [46], [48] showed such algorithms *asymptotically* converge to a stationary point, under some conditions—but the finite iteration complexity was not shown. In this work, we show that, with properly pre-defined parameters $\alpha^{(t)}$ and $\beta^{(t)}$, GradPAPA is guaranteed to find a stationary point in a *sublinear* rate, if the projections in (17) and (18) are solved.

Algorithm 1: GradPAPA for solving (11).

Input: HSI $\underline{\boldsymbol{Y}}$; starting points $\boldsymbol{C}^{(0)}$ and $\boldsymbol{S}^{(0)}$; pre-defined sequences of $\mu_1^{(t)}$, $\mu_2^{(t)}$, $\alpha^{(t)}$, and $\beta^{(t)}$.

Parameters: $\gamma_1^{(0)} = \gamma_2^{(0)} = 1$, θ_r, q, ε, L_r, and $\widetilde{L}$.
$\check{\boldsymbol{C}}^{(0)} = \boldsymbol{C}^{(0)}$, $\check{\boldsymbol{S}}^{(0)} = \boldsymbol{S}^{(0)}$, $t = 0$.
repeat
%% **update** $\boldsymbol{C}$ %%
$\boldsymbol{C}^{(t+1)} \leftarrow \max\left\{\check{\boldsymbol{C}}^{(t)} - \alpha^{(t)}\boldsymbol{G}_{\check{\boldsymbol{C}}}^{(t)},\ \boldsymbol{0}\right\}$;
$\check{\boldsymbol{C}}^{(t+1)} = \boldsymbol{C}^{(t+1)} + \mu_1^{(t)}(\boldsymbol{C}^{(t+1)} - \boldsymbol{C}^{(t)})$;
%% **update** $\boldsymbol{S}$ %%
$\check{\boldsymbol{W}}^{(0)} = \check{\boldsymbol{S}}^{(t)} - \beta^{(t)}\boldsymbol{G}_{\check{\boldsymbol{S}}}^{(t)}$;
  repeat
    if choosing cons. (17) **do**
      update $\check{\boldsymbol{F}}^{(t+1)}$ by (20);
    if choosing cons. (18) **do**
      update $\check{\boldsymbol{F}}^{(t+1)}$ by (21);
    $\check{\boldsymbol{W}}^{(k+1)} \leftarrow \mathsf{Proj}_{\mathcal{A}_{\mathrm{splx}}}\left(\check{\boldsymbol{F}}^{(k+1)}\right)$;
  until satisfying the stopping rule;
$\boldsymbol{S}^{(t+1)} = \check{\boldsymbol{W}}^{(k+1)}$;
$\check{\boldsymbol{S}}^{(t+1)} = \boldsymbol{S}^{(t+1)} + \mu_2^{(t)}(\boldsymbol{S}^{(t+1)} - \boldsymbol{S}^{(t)})$;
$t = t + 1$;
until satisfying the stopping criterion.

Output: $\widehat{\boldsymbol{C}} = \boldsymbol{C}^{(t)}$ and $\widehat{\boldsymbol{S}} = \boldsymbol{S}^{(t)}$.

To see our result, we define $\boldsymbol{Z} = (\boldsymbol{C}, \boldsymbol{S})$. Let $\mathcal{J}(\boldsymbol{Z})$ and $\mathcal{C}(\boldsymbol{Z})$ be the objective function of (11) and the indicator function of its constraints, respectively. This way, the optimization problem is written as

$$\min_{\boldsymbol{Z}}\ \mathcal{J}(\boldsymbol{Z}) + \mathcal{C}(\boldsymbol{Z}). \tag{26}$$

According to [49, Lemma 2.1], the gradient of the objective function in (26) is:

$$\begin{aligned}&\partial(\mathcal{J}(\boldsymbol{Z}) + \mathcal{C}(\boldsymbol{Z}))\\ =&[(\partial_{\boldsymbol{C}}\mathcal{J}(\boldsymbol{C},\boldsymbol{S}) + \partial\mathcal{C}_{\boldsymbol{C}}(\boldsymbol{C}))^\top, (\partial_{\boldsymbol{S}}\mathcal{J}(\boldsymbol{C},\boldsymbol{S}) + \partial\mathcal{C}_{\boldsymbol{S}}(\boldsymbol{S}))^\top]^\top,\end{aligned}$$

where $\mathcal{C}_{\boldsymbol{C}}(\boldsymbol{C})$ and $\mathcal{C}_{\boldsymbol{S}}(\boldsymbol{S})$ are indicator functions of the constraints on $\boldsymbol{C}$ and $\boldsymbol{S}$, respectively.

We adopt the definition of ϵ-stationary point in [50]:

Definition 1. A point $\boldsymbol{Z}$ is an ϵ-stationary point of the optimization problem in (26) if

$$\mathrm{dist}\left(\boldsymbol{0}, \partial_{\boldsymbol{Z}}\mathcal{J}(\boldsymbol{Z}) + \partial_{\boldsymbol{Z}}\mathcal{C}(\boldsymbol{Z})\right) \le \epsilon,$$

where $\partial_{\boldsymbol{Z}}$ denotes the subgradient with respect to $\boldsymbol{Z}$.

It is readily seen that when ϵ is small, the definition covers a vicinity of any stationary point of Problem (11). We also

define

$$L_{\boldsymbol{C}}^{(t)} = \sigma_{\max}^2\left(\boldsymbol{S}^{(t)}\right)$$
$$L_{\boldsymbol{S}}^{(t)} = \sigma_{\max}^2\left(\boldsymbol{C}^{(t+1)}\right) + q\max_r \theta_r \sigma_{\max}\left(\boldsymbol{H}_x^\top \boldsymbol{U}_r^{(t)} \boldsymbol{H}_x\right) + q\max_r \theta_r \sigma_{\max}\left(\boldsymbol{H}_y^\top \boldsymbol{V}_r^{(t)} \boldsymbol{H}_y\right),$$

where $\boldsymbol{U}_r^{(t)}$ and $\boldsymbol{V}_r^{(t)}$ are diagonal matrices with $[\boldsymbol{U}_r^{(t)}]_{i,i} = ([\boldsymbol{H}_x \boldsymbol{q}_r^{(t)}]_i^2 + \varepsilon)^{\frac{q-2}{2}}$, and $[\boldsymbol{V}_r^{(t)}]_{i,i} = ([\boldsymbol{H}_y \boldsymbol{q}_r^{(t)}]_i^2 + \varepsilon)^{\frac{q-2}{2}}$, $r = 1, \ldots, R$. Using these notations, we present the following convergence guarantee:

Proposition 1. Assume that $\mathcal{J}^\star = \min$ (26) is finite, that $0 < \inf_t \alpha^{(t)} \le \sup_t \alpha^{(t)} < \infty$ and $0 < \inf_t \beta^{(t)} \le \sup_t \beta^{(t)} < \infty$ for all t, that there exist $c_1, \ldots, c_4$ such that $c_2 L_{\boldsymbol{C}}^{(t)} \le 1/\alpha^{(t)} \le c_1 L_{\boldsymbol{C}}^{(t)}$ and $c_4 L_{\boldsymbol{S}}^{(t)} \le 1/\beta^{(t)} \le c_3 L_{\boldsymbol{S}}^{(t)}$ in all iterations, and that the projection in (25) is solved exactly. Then, it holds that

$$\min_{t'=0,1,\ldots,t} \operatorname{dist}\left(\boldsymbol{0}, \partial_{\boldsymbol{Z}} \mathcal{J}\left(\boldsymbol{Z}^{(t'+1)}\right) + \partial \mathcal{C}_{\boldsymbol{Z}}\left(\boldsymbol{Z}^{(t'+1)}\right)\right) \le \frac{C}{\sqrt{t}},$$

where

$$C = C_1\sqrt{4\left(\mathcal{J}\left(\boldsymbol{Z}^{(0)}\right) - \mathcal{J}^\star\right)/C_2},$$
$$C_1 = \max\{\bar{\mu}_1, \bar{\mu}_2, 1\}\max\left\{(c_1+1)\sup_t \alpha^{(t)}, (c_3+1)\sup_t \beta^{(t)}\right\},$$
$$C_2 = \min\left\{(1-\tau_1^2)/\sup_t \alpha^{(t)}, (1-\tau_2^2)/\sup_t \beta^{(t)}\right\},$$

in which the constants satisfy

$$\mu_1^{(t)} \le \tau_1\sqrt{\left(c_1 L_{\boldsymbol{C}}^{(t-1)}\right)/\left(c_2 L_{\boldsymbol{C}}^{(t)}\right)} \le \bar{\mu}_1, \tag{27a}$$
$$\mu_2^{(t)} \le \tau_2\sqrt{\left(c_3 L_{\boldsymbol{S}}^{(t-1)}\right)/\left(c_4 L_{\boldsymbol{S}}^{(t)}\right)} \le \bar{\mu}_2, \tag{27b}$$

and $\tau_1 < 1$, $\tau_2 < 1$ for all t.

The proposition asserts that the solution sequence produced by Algorithm 1 converges to an ϵ-stationary point in $\mathcal{O}(1/\epsilon^2)$ iterations. The proof of Proposition 1 is relegated to Appendix B. Our convergence analysis is reminiscent of the technique in [50]. However, the work in [50] is concerned with single block optimization with convex constraints. Our proof generalizes the results to cover multiple block cases with nonconvex constraints.

Remark 2. We hope to remark that the result in Proposition 1 is built upon the premise that (14) can be solved to optimality. When the nuclear norm based constrained is employed (i.e., in the GradPAPA-NN version), this assumption is not hard to be met, since $\mathcal{S} = \mathcal{A}_{\text{splx}} \cap \mathcal{A}_{\text{LR}}$ is a convex set if $\mathcal{A}_{\text{LR}}$ is from (18). If $\mathcal{A}_{\text{LR}}$ is from (17), then $\mathcal{S} = \mathcal{A}_{\text{splx}} \cap \mathcal{A}_{\text{LR}}$ is nonconvex. Projection onto this set is not guaranteed in theory using the proposed AP algorithm. Interestingly, our empirical study shows that AP almost always finds a feasible solution in $\mathcal{S} = \mathcal{A}_{\text{splx}} \cap \mathcal{A}_{\text{LR}}$. We leave theoretical underpinning of this nonconvex projection step to a future work.

Remark 3. It is important to note that the sequences $\{\mu_1^{(t)}\}$ and $\{\mu_2^{(t)}\}$ need to be specified for GradPAPA. By Proposition 1, the sequences should be selected so that (27a) and (27b) are satisfied. This is nontrivial, since four constants $c_1, \ldots, c_4$ are involved. Nonetheless, in practice, we find that using Nesterov's extrapolation sequence [45] as a heuristic to select $\{\mu_1^{(t)}\}$ and $\{\mu_2^{(t)}\}$ works fairly well—and spares us the computations to determine the two sequences. Hence, in this work, we simply set

$$\mu_i^{(t)} = \frac{\gamma_1^{(t)} - 1}{\gamma_i^{(t+1)}}, \quad \gamma_i^{(t+1)} = \frac{1 + \sqrt{1 + 4\left(\gamma_1^{(t)}\right)^2}}{2},$$

with $\gamma_i^{(0)} = 1$ for $i = 1, 2$. In addition, we choose the step sizes $\alpha^{(t)}$ and $\beta^{(t)}$ as $\alpha^{(t)} = 1/L_{\boldsymbol{C}}^{(t)}$, $\beta^{(t)} = 1/L_{\boldsymbol{S}}^{(t)}$ as often done in the unextrapolated alternating gradient descent-based algorithms, which also works well in practice.

TABLE I
COMPLEXITY OF EACH TERM OF GRADPAPA.

Terms	Complexity
$\boldsymbol{C}^{(t)}\boldsymbol{S}^{(t)}(\boldsymbol{S}^{(t)})^\top - \boldsymbol{Y}(\boldsymbol{S}^{(t)})^\top$	$\mathcal{O}(IJKR))$
$((\boldsymbol{C}^{(t+1)})^\top \boldsymbol{C}^{(t+1)}\boldsymbol{S}^{(t)}) - (\boldsymbol{C}^{(t+1)})^\top \boldsymbol{Y}$	$\mathcal{O}(IJKR)$
$\boldsymbol{H}_x^\top \boldsymbol{U}_r^{(t)} \boldsymbol{H}_x \boldsymbol{q}_r^{(t)}$, $\boldsymbol{H}_y^\top \boldsymbol{V}_r^{(t)} \boldsymbol{H}_y \boldsymbol{q}_r^{(t)}$	$\mathcal{O}(I^2JR)$
Computation of $\alpha^{(t)}$ and $\beta^{(t)}$	$\mathcal{O}(R^3)$
Projection of (19b)	$\mathcal{O}(IJR\log R)$
Projection of (20)	$\mathcal{O}(IJLR)$
Projection of (21)	$\mathcal{O}(IJ\min\{I,J\}R + \min\{I,J\}\log\min\{I,J\})$

TABLE II
COMPARISON OF THE COMPLEXITY BETWEEN ALS-MU ALGORITHMS AND GRADPAPA.

Methods	Complexity
ALS-MU algorithms	$\mathcal{O}(IJKLR + IKL^2R^2 + JKL^2R^2)$
GradPAPA-LR (no TV)	$\mathcal{O}(IJKR + mIJR(\log R + L))$
GradPAPA-NN (no TV)	$\mathcal{O}(IJKR + m(IJR(\min\{I,J\} + \log R) + \min\{I,J\}\log\min\{I,J\}))$
GradPAPA-LR (with TV)	$\mathcal{O}(IJR(I+K) + mIJR(\log R + L))$
GradPAPA-NN (with TV)	$\mathcal{O}(IJR(I+K) + m(IJR(\min\{I,J\} + \log R) + \min\{I,J\}\log\min\{I,J\}))$

E. Computational Complexity

The detailed complexity analysis of the proposed Algorithm 1 is listed in Table I. The computation of the gradients $\boldsymbol{G}_{\boldsymbol{C}}^{(t)}$ and $\boldsymbol{G}_{\boldsymbol{S}}^{(t)}$ takes $\mathcal{O}(IJKR)$ and $\mathcal{O}(IJR(I+K))$ flops, respectively. In the computation of the step size $\beta^{(t)}$, computing $\sigma_{\max}(\boldsymbol{H}_x^\top \boldsymbol{U}_r^{(t)} \boldsymbol{H}_x)$ and $\sigma_{\max}(\boldsymbol{H}_y^\top \boldsymbol{V}_r^{(t)} \boldsymbol{H}_y)$ may increase the computational flops at each iteration. In this work, instead of computing the exact values, we just compute their upper bound. For example, we have

$$\sigma_{\max}(\boldsymbol{H}_x^\top \boldsymbol{U}_r^{(t)} \boldsymbol{H}_x) \le \sigma_{\max}(\boldsymbol{H}_x^\top)\sigma_{\max}(\boldsymbol{U}_r^{(t)})\sigma_{\max}(\boldsymbol{H}_x),$$

where $\sigma_{\max}(\boldsymbol{U}_r^{(t)})$ is simply chosen to be the largest value of the diagonal matrix $\boldsymbol{U}_r^{(t)}$ and the other two terms are precomputed. Therefore, computing $\alpha^{(t)}$ and $\beta^{(t)}$ takes $\mathcal{O}(R^3)$ flops, but R is normally small. In the AP solver, (19b) costs $\mathcal{O}(IJR\log R)$ flops via the water-filling type algorithm, the SVD in (20) takes $\mathcal{O}(IJLR)$, and the projection in (21) takes $\mathcal{O}(IJ\min\{I,J\}R + \min\{I,J\}\log\min\{I,J\})$.

The complexity of each iteration of the proposed algorithms are summarized in Table II. To be specific, the proposed algorithms take at most $\mathcal{O}(IJR(I+K) + m(IJR(\min\{I,J\} + \log R) + \min\{I,J\}\log\min\{I,J\}))$ flops at each iteration, where m is the number of AP iterations—usually only 3 to 6 (see Table IV). In addition, $\widetilde{L}$ is a tunable parameter related to the nuclear norm of $\boldsymbol{S}_r$ and admits the same magnitude as the size of HSI data in the numerical experiments.

It is worth noting that the proposed algorithm can be much more lightweight relative to the ALS-MU algorithms in [22]–[28]. Even the vanilla ALS-MU algorithm in [22] takes $\mathcal{O}(IJKLR + IKL^2R^2 + JKL^2R^2)$ flops in each iteration. To see how much our algorithm could save in terms of per-iteration complexity, consider an example where $\widetilde{L}R \approx LR \approx I \approx J \approx K$. In this example, one can see that ALS-MU takes $\mathcal{O}(I^4)$ flops per-iteration, but our algorithms only cost $\mathcal{O}(I^3R)$ flops in each iteration—no matter with or without the TV regularization. Note that I is often not small, thus saving an order-of-magnitude complexity w.r.t. I can be quite significant, as one will see in the experiments.

IV. Experiments

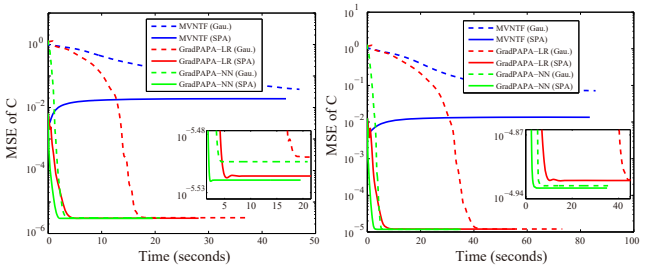
In this section, we showcase the effectiveness and efficiency of the proposed GradPAPA methods using experiments on synthetic data, semi-real data, and real data.

A. Experiment Settings

1) Baselines: We use a number of relevant baselines. These include SISAL [5], MVCNMF [6], MVNTF [22], MVNTFTV [23], SSWNTF [26], and SPLRTF [27]. Note that the first two methods are classic low-rank matrix factorization-based HU algorithms—considering the minimum volume constraint on the spectral signatures; the remaining four methods are ALS-MU based LL1 algorithms that MVNTF [22] handled the formulation shown in (6), MVNTFTV [23], SSWNTF [26], and SPLRTF [27] worked with different regularization terms, e.g., total variation in [23], weighted sparsity in [26], and sparsity and low rank in [27].

2) Algorithm Settings: The proposed GradPAPA involves a set of parameters, i.e., the endmember number R, the parameters L and $\widetilde{L}$ related to $\boldsymbol{S}_r$, and the regularization parameter $\{\theta_r\}$. The number of endmember R can be selected by the existing estimation algorithms, e.g., [12], [51]. The rank L is selected as the maximal number which satisfies the condition shown in Theorem 1. The parameter $\widetilde{L}$ is chosen by a heuristic, namely, $\widetilde{L} = 1.5 \times \max\{I, J, K\}$. The parameter θ is selected from one of values in $\{z \times 10^{-4}\}$ where $z = 1, 3, 5, 7, 9$ in the synthetic and semi-real experiment. We present the best result in terms of the estimation accuracy over the z's, but one will see that there is a wide range of θ that gives similar results (cf. Fig. 14); that is, the algorithm seems not to be sensitive to this hyperparameter. In the real experiments, we set θ as 10^{-4} in the real-data experiment. For the parameters in the TV regularization (16), we fix $q = 0.5$ and $\varepsilon = 10^{-3}$. In addition, when the relative change of the iterates of the latent factors is smaller than 10^{-3}, we stop the AP solver in the proposed algorithms.

For the parameter settings of baselines, we mainly follow the respective papers' suggestions and make proper adjustments to enhance their performance under our settings. The baselines and proposed algorithms are terminated when the relative change of the objective value is smaller than 10^{-5}. Since when handling large-scale problems, the ALS-MU based algorithms typically run with extra lengthy time but do not reach this stopping criterion, we also set the maximal number of iterations to be 1,200 (resp. 2,500) for the synthetic data experiments (resp. semi-real and real data experiments).



(a) $R = 5$ (b) $R = 10$

Fig. 4. The average MSE values against time of MVNTF and GradPAPA.

TABLE III
Feasibility percentage of estimated $\widehat{\boldsymbol{S}}$.

Initialization		Gaussian Init.		SPA Init.	
Constraints	Methods	$R = 5$	$R = 10$	$R = 5$	$R = 10$
STO	MVNTF ($p = 10^{-2}$)	10.38%	12.69%	10.09%	11.72%
	MVNTF ($p = 10^{-5}$)	0.010%	0.014%	0.011%	0.006%
	GradPAPA-LR ($p = 10^{-5}$)	100%	100%	100%	100%
	GradPAPA-NN ($p = 10^{-5}$)	100%	100%	100%	100%
LR	MVNTF	100%	100%	100%	100%
	GradPAPA-LR	99.88%	99.90%	99.88%	99.90%
	GradPAPA-NN	97.94%	97.02%	97.94%	97.22%

3) Metrics: In the synthetic and semi-real experiments, we mainly use the *spectral angle distance* (SAD) [22] of the estimated $\widehat{\boldsymbol{C}}$ and the *mean squared error* (MSE) [7] of the estimated $\widehat{\boldsymbol{C}}$ and $\widehat{\boldsymbol{S}}$ as the performance metrics. The SAD of the estimated $\widehat{\boldsymbol{C}}$ is defined as follows:

$$\text{SAD} = \min_{\pi \in \Pi} \frac{1}{R} \sum_{r=1}^{R} \arccos\left(\frac{\boldsymbol{c}_r \widehat{\boldsymbol{c}}_{\pi_r}^\top}{\|\boldsymbol{c}_r\|_2 \|\widehat{\boldsymbol{c}}_{\pi_r}\|_2} \right),$$

and the MSE of the estimated $\widehat{\boldsymbol{C}}$ is defined as follows:

$$\text{MSE} = \min_{\pi \in \Pi} \frac{1}{R} \sum_{r=1}^{R} \left\| \frac{\boldsymbol{c}_r}{\|\boldsymbol{c}_r\|_2} - \frac{\widehat{\boldsymbol{c}}_{\pi_r}}{\|\widehat{\boldsymbol{c}}_{\pi_r}\|_2} \right\|_2^2,$$

where Π is the set of all permutation of $\{1, \ldots, R\}$, $\boldsymbol{c}_r$ and $\widehat{\boldsymbol{c}}_{\pi_r}$ are the ground truth of the r-th column of $\boldsymbol{C}$ and the corresponding estimate, respectively. The MSE of $\widehat{\boldsymbol{S}}$ is defined in an identical way using its transpose.

For the real data experiment, it is hard to measure the performance quantaitively due to the absence of ground-truth. Therefore, we qualitatively comment on the performance of the estimated factors using visual inspection. In addition, we use the pure pixels manually extracted from HSI data to measure the quality of the estimated endmembers.

B. Synthetic Data Experiments

We first use a set of experiments to test the basic properties of GradPAPA, e.g., accuracy, sensitivity to initialization, convergence speed, and feasibility enforcing. In these experiments, we set $\theta = 0$ and compare the GradPAPA algorithm with the plain-vanilla ALS-MU algorithm, namely, MVNTF in [22], that also does not have any structural regularization on the abundances except for the nonnegativity and sum-to-one constraints.

TABLE IV
THE AVERAGE NUMBER OF AP ITERATIONS (m) UNDER DIFFERENT R'S AND INITIALIZATION SCHEMES.

Initialization	Gaussian Init.		SPA Init.	
	$R=5$	$R=10$	$R=5$	$R=10$
Ave. AP iterations (LR (17))	6	6	3	4
Ave. AP iterations (LR (18))	2	2	2	2

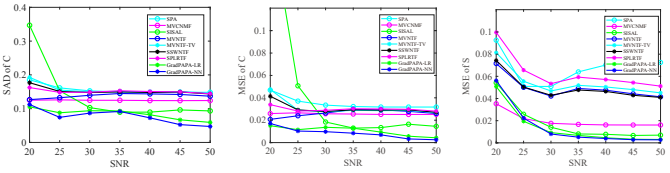
1) Synthetic Data Generation: The procedures of generating $\boldsymbol{C}$ and $\boldsymbol{S}$ are detailed as follows: 1) we generate two matrices $\boldsymbol{E}_1 \in \mathbb{R}^{K\times R}$ and $\boldsymbol{E}_2 \in \mathbb{R}^{R\times IJ}$ following the i.i.d. Gaussian distribution with unit variance and zero mean; 2) we use the AP algorithm in (19) on the matrix $\boldsymbol{E}_2$ to produce $\boldsymbol{S}$ satisfying the low-rank structure (i.e., $\boldsymbol{S} \in \mathcal{A}_{\rm LR}$ in (17)); similarly, we threshold the negative values of $\boldsymbol{E}_1$ to obtain the nonnegative matrix $\boldsymbol{C}$; 3) finally, we synthesize the LL1 tensor $\boldsymbol{Y} \leftarrow \boldsymbol{C}\boldsymbol{S}$. In addition, the i.i.d. zero-mean Gaussian noise is added to the synthetic tensors. The size of the synthetic tensor is set to be $I = J = K = 100$, $L = 30$, and $R = 5$ or 10. We test two initialization strategies, including i.i.d. Gaussian initialization and successive projection algorithm (SPA)-based [47] initialization. The details of SPA-based initialization are as follows: 1) we use the spectral pixel extraction method SPA to generate $\tilde{\boldsymbol{C}}$ and threshold its negative values to get the initial endmembers $\boldsymbol{C}^{(0)}$; 2) we generate the initial abundance maps by $\tilde{\boldsymbol{S}} = \left((\boldsymbol{C}^{(0)})^\dagger \boldsymbol{C}^{(0)}\right)^\dagger (\boldsymbol{C}^{(0)})^\dagger \boldsymbol{Y}$, where $(\boldsymbol{C}^{(0)})^\dagger$ is the pseudo-inverse of $\boldsymbol{C}^{(0)}$; 3) we use the nonnegative matrix factorization method by hierarchical alternating least squares method [52] to get the latent factors $\{\tilde{\boldsymbol{A}}_r, \tilde{\boldsymbol{B}}_r\}_{r=1}^R$; 4) finally, we obtain the initial abundance maps $\boldsymbol{S}_r^{(0)} = \tilde{\boldsymbol{A}}_r \tilde{\boldsymbol{B}}_r^\top$.

2) Results: Fig. 4 shows the MSE curves of the estimated $\widehat{\boldsymbol{C}}$ against time by the algorithms. The results are averaged from 20 independent trials under SNR=25dB. A number of observations are in order. First, for both $R = 5$ and $R = 10$, the proposed GradPAPA algorithm performs much better than the ALS-MU based MVNTF algorithm in terms of accuracy and speed. Second, using the same Gaussian initialization, GradPAPA-NN converges faster than GradPAPA-LR to reach the same MSE level. In particular, when $R = 10$, GradPAPA-NN converges to an MSE level close to 10^{-5} using less than 5 seconds, but GradPAPA-LR needs 40 seconds to reach a similar level. Third, using SPA can help both GradPAPA-NN and GradPAPA-NN to converge even faster. In particular, the SPA initialization further speeds up GradPAPA-LR by about 75%. Although MVNTF works to a certain extent, its MSE is more than three orders of magnitude higher than those of GradPAPA-NN and GradPAPA-LR in all cases.

Table III shows how often the solutions obtained by the algorithms satisfying the structural constraints on latent factors in the context HU—i.e., the nonnegativity of $\boldsymbol{S}$ and $\boldsymbol{C}$, the sum-to-one constraint of the abundances, and the low-rank constraint on $\boldsymbol{S}_r$. Note that nonnegativity is relatively easy to enforce. Hence, we look into the satisfaction of two harder to enforce constraints, namely, the sum-to-one (STO) constraint on the columns of $\boldsymbol{S}$ and the low-rank (LR) constraint on $\boldsymbol{S}_r$. To measure the STO feasibility, we calculate the percentage of the estimated abundance vectors (i.e., $\boldsymbol{s}_\ell$) satisfying $|\mathbf{1}^\top \boldsymbol{s}_\ell - 1| \le p$, where p is specified in Table III.

TABLE V
SAD, MSE, AND TIME PERFORMANCE ON THE TERRAIN DATA (SNR=40DB) BY THE ALGORITHMS.

Methods	SAD of $\boldsymbol{C}$	MSE of $\boldsymbol{C}$	MSE of $\boldsymbol{S}$	Time (min.)
SPA	0.1468 ± 0.0008	0.0318 ± 0.0005	0.0701 ± 0.0025	—
MVCNMF	0.1240 ± 0.0003	0.0251 ± 0.0001	0.0162 ± 0.0001	—
SISAL	0.0897 ± 0.0041	0.0134 ± 0.0016	0.0077 ± 0.0016	—
MVNTF	0.1437 ± 0.0107	0.0292 ± 0.0044	0.0475 ± 0.0143	79.85 ± 0.92
MVNTFTV	0.1495 ± 0.0094	0.0285 ± 0.0036	0.0504 ± 0.0130	101.80 ± 4.98
SSWNTF	0.1479 ± 0.0088	0.0290 ± 0.0035	0.0463 ± 0.0122	82.00 ± 1.04
SPLRTF	0.1507 ± 0.0145	0.0300 ± 0.0040	0.0571 ± 0.0163	142.68 ± 1.23
GradPAPA-LR	0.0819 ± 0.0016	0.0094 ± 0.0004	0.0046 ± 0.0002	6.32 ± 0.38
GradPAPA-NN	**0.0727 ± 0.0005**	**0.0069 ± 0.0001**	**0.0039 ± 0.0001**	**5.09 ± 0.05**



(a) SAD of $\boldsymbol{C}$ (b) MSE of $\boldsymbol{C}$ (b) MSE of $\boldsymbol{S}$

Fig. 5. The estimated SAD of $\boldsymbol{C}$ and MSE of $\boldsymbol{C}$ and $\boldsymbol{S}$ for Terrain under different noises.

The LR constraint satisfaction is measured by averaging $(\sum_{i=1}^{L} \sigma_i^r / \sum_{i=1}^{\min\{I,J\}} \sigma_i^r) \times 100\%$ over $r = 1\ldots, R$, where σ_i^r is the i-th singular value of the estimated $\boldsymbol{S}_r$. One can see that MVNTF has difficulties in satisfying the STO constraint, maybe because it uses a "soft" reglularization to enforce this requirement [cf. Eq. (6)]. However, GradPAPA-NN and GradPAPA-LR almost achieve 100% feasibility for STO and LR. More notably, such feasibility can be performed at a relatively small cost: Table IV shows that for the $\boldsymbol{S}$ projection problem, GradPAPA-LR only needs about 6 AP iterations and GradPAPA-NN needs about 2 AP iterations.

C. Semi-Real Data Experiments

1) Semi-Real Data: These datasets are "semi-real" because the pixels are synthesized exactly following the LMM model using abundance maps and endmembers extracted from real datasets. This type of semi-real data helps validate the effectiveness of our algorithms under controlled (yet realistic) data generating processes and noise levels.

The first experiment uses the Terrain data. This dataset is acquired by the HYDICE sensor. After removing water absorption-corrupted bands, we obtain an HSI data that has a size of $500 \times 307 \times 166$. This data contains 5 prominent materials, namely, `Soil1`, `Soil2`, `Tree`, `Shadow`, and `Grass`, so the number of endmembers is set to be $R = 5$.

TABLE VI
SAD, MSE, AND TIME PERFORMANCE ON THE TERRAIN DATA (SNR=10DB) BY THE ALGORITHMS.

Methods	MSE of $\boldsymbol{C}$	MSE of $\boldsymbol{C}$	MSE of $\boldsymbol{S}$	Time (min.)
SPA	0.4184 ± 0.0087	0.2169 ± 0.0738	0.2300 ± 0.0396	—
MVCNMF	0.3813 ± 0.0481	0.1782 ± 0.0497	0.2053 ± 0.0558	—
SISAL	0.5461 ± 0.0449	0.3298 ± 0.0900	0.1500 ± 0.0236	—
MVNTF	0.2879 ± 0.0337	0.1104 ± 0.0619	0.1769 ± 0.0960	59.02 ± 21.20
MVNTFTV	0.3180 ± 0.0437	0.1312 ± 0.0626	0.1815 ± 0.0807	135.93 ± 3.18
SSWNTF	0.3100 ± 0.0292	0.1469 ± 0.0496	0.2061 ± 0.0459	109.40 ± 2.47
SPLRTF	0.2866 ± 0.0638	0.1066 ± 0.0425	0.1862 ± 0.0663	176.88 ± 4.05
GradPAPA-LR	**0.2631 ± 0.0367**	**0.0965 ± 0.0270**	**0.1318 ± 0.0263**	12.85 ± 0.43
GradPAPA-NN	0.3287 ± 0.0797	0.1352 ± 0.0634	0.1375 ± 0.0349	**7.82 ± 1.01**

The second dataset we used is a subscene of the Urban data with a size of $307 \times 307 \times 162$. This dataset is obtained by

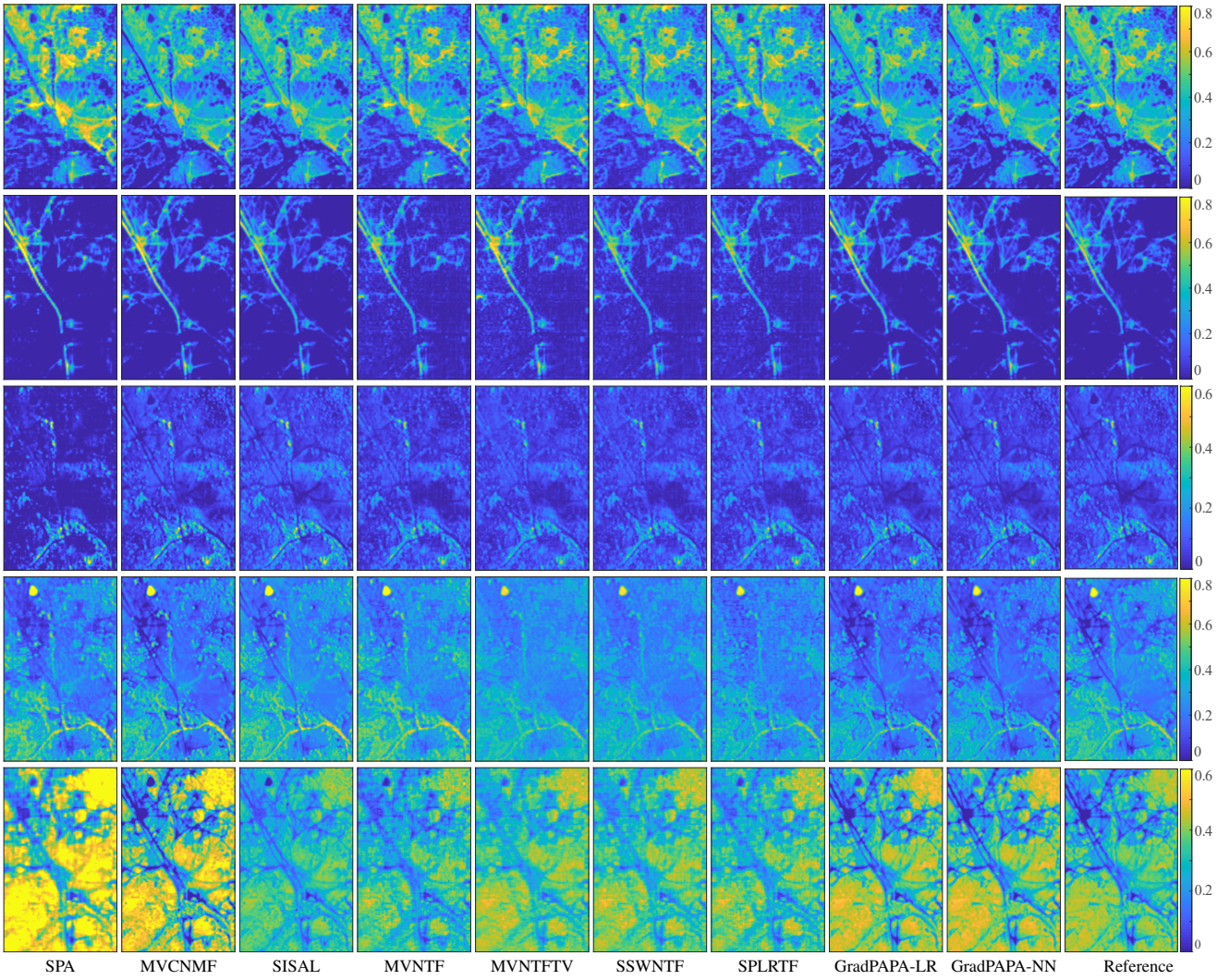


Fig. 6. The estimated abundance maps of Terrain data (SNR=40dB) by different methods. From top to bottom: `Soil1`, `Soil2`, `Tree`, `Shadow`, and `Grass`.

TABLE VII
SAD, MSE, AND TIME PERFORMANCE ON THE URBAN DATA (SNR=30DB) BY THE ALGORITHMS.

Methods	MSE of $\boldsymbol{C}$	MSE of $\boldsymbol{C}$	MSE of $\boldsymbol{S}$	Time (min.)
SPA	0.1120 ± 0.0006	0.0166 ± 0.0002	0.0497 ± 0.0043	—
MVCNMF	0.0963 ± 0.0020	0.0106 ± 0.0006	0.0355 ± 0.0010	—
SISAL	0.0911 ± 0.0019	0.0095 ± 0.0005	0.0358 ± 0.0011	—
MVNTF	0.0815 ± 0.0230	0.0085 ± 0.0016	0.0356 ± 0.0093	42.21 ± 1.55
MVNTFTV	0.0915 ± 0.0263	0.0105 ± 0.0042	0.0372 ± 0.0093	46.66 ± 0.57
SSWNTF	0.0661 ± 0.0199	0.0051 ± 0.0010	0.0316 ± 0.0075	43.18 ± 1.17
SPLRTF	0.1009 ± 0.0237	0.0127 ± 0.0039	0.0418 ± 0.0088	72.60 ± 0.86
GradPAPA-LR	**0.0431 ± 0.0003**	**0.0020 ± 0.0001**	**0.0246 ± 0.0004**	5.38 ± 0.10
GradPAPA-NN	0.0537 ± 0.0012	0.0033 ± 0.0002	0.0272 ± 0.0009	**0.90 ± 0.04**

the HYDICE sensor. The number of endmembers is set as 4, including `Asphalt`, `Grass`, `Tree`, and `Roof`. The ground-truth of abundance maps and spectral signatures are available online (https://rslab.ut.ac.ir/data). The details of generating the semi-real dataset can be found in [53].

2) Baselines: In addition to MVNTF [22] that does not have structural regularization on latent factors, we compare GradPAPA with another five baselines, i.e., MVCNMF [6], SISAL [5], MVNTFTV [23], SSWNTF [26], and SPLRTF [27].

3) Results: Table V shows the SAD and MSE performance of the algorithms on the Terrain data under SNR=40dB. One can see that the two versions of GradPAPA achieve an order of magnitude improvement over SPA in terms of SAD and MSE. The table also includes the runtime performance of all the LL1 algorithms. In terms of running time, the two versions of GradPAPA use about 6 minutes for this task, while the four ALS-MU based LL1 baselines (i.e., MVNTF, MVNTFTV, SSWNTF, and SPLRTF) use more than 1 hour. We should mention that SISAL and SPA are in general faster than the other baselines because they do not work with the LL1 model but a computationally more convenient MF model. But as we mentioned, their identifiability has a nontrivial unoverlapped "regime" with that of the LL1 model.

Fig. 5 shows the SADs and MSEs of the algorithms on the Terrain data under different SNRs. One can see that the proposed algorithms outperform the baselines in almost all cases. The abundance maps and spectral signatures produced by GradPAPA-LR and GradPAPA-NN are also visually much closer to the ground truth; see Figs. 6 and 7. One can see that our algorithms perform well in keeping the edges of abundance maps, better than baselines.

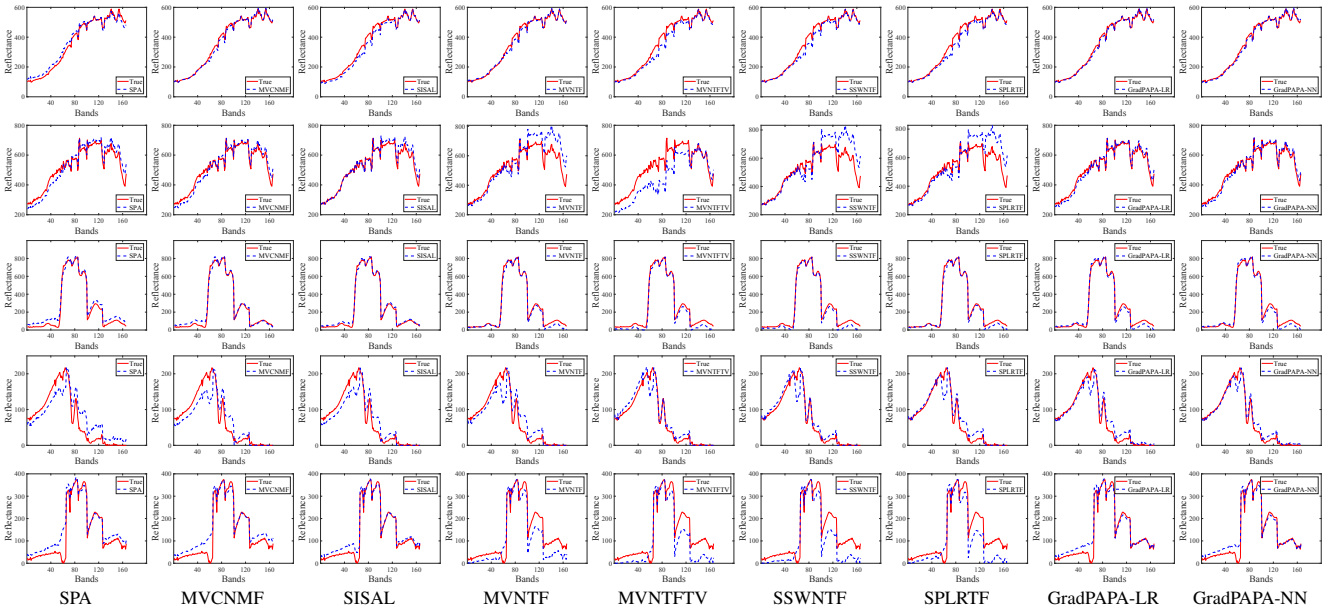


Fig. 7. The estimated spectral signatures of Terrain data (SNR=40dB) by different methods. From top to bottom: `Soil1`, `Soil2`, `Tree`, `Shadow`, and `Grass`.

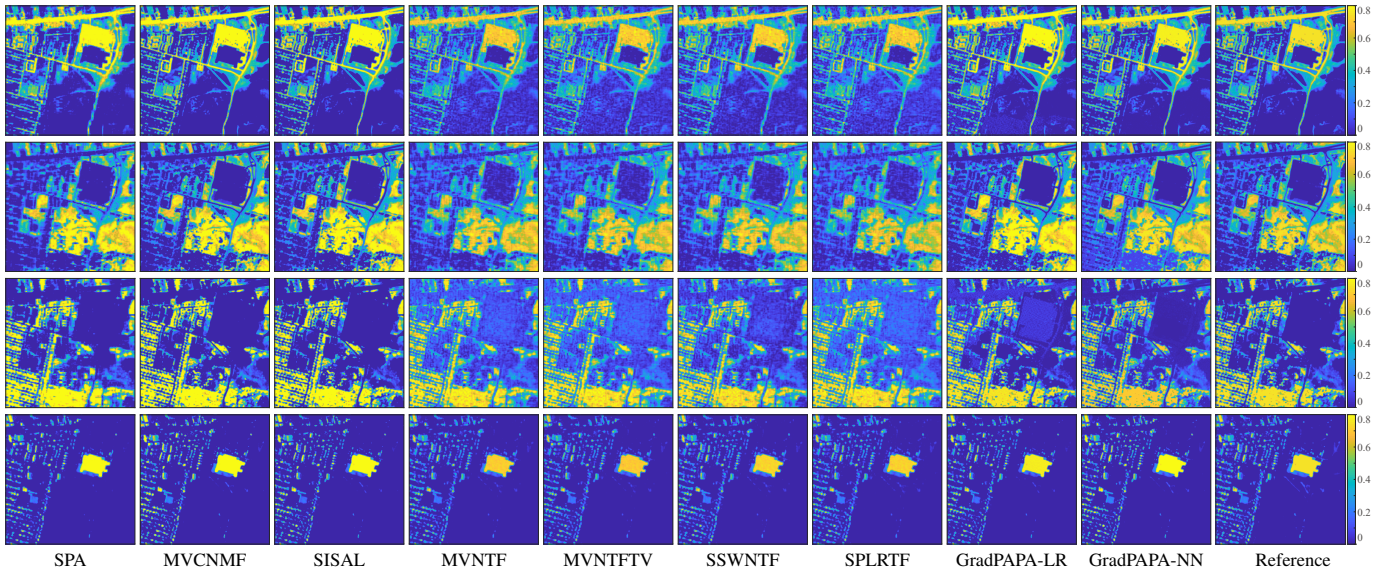


Fig. 8. The estimated abundance maps of Urban data (SNR=30dB) by different methods. From top to bottom: `Asphalt`, `Grass`, `Tree`, and `Roof`.

We also validate the effectiveness and convergence of the proposed algorithms under a relatively low SNR in the context of HU, i.e., SNR = 10dB. Table VI shows the MSE performance of the algorithms. One can see that GradPAPA-LR achieves the lowest MSE of estimated $\boldsymbol{C}$ and $\boldsymbol{S}$. In terms of running time, the proposed methods are approximately 5 times faster than the four ALS-MU based LL1 baselines (i.e., MVNTF, MVNTFTV, SSWNTF, and SPLRTF).

Table VII shows the SADs and MSEs of all methods on the Urban dataset under SNR=30dB. One can see that our GradPAPA methods again achieve the best performance of estimating $\boldsymbol{C}$ and $\boldsymbol{S}$. In terms of running time, the proposed methods are approximately 10 times faster than the ALS-MU based LL1 baselines. In addition, one can see that GradPAPA-LR achieves lower estimated MSE values than GradPAPA-NN at the expense of more running time.

Figs. 8-9 show the estimated abundance maps and endmembers, respectively. Again, the results by our algorithms are visually closer the ground truth. In particular, all the algorithms—except for the two GradPAPA algorithms—seem to have difficulties in correctly recovering the endmember `Asphalt`. Both GradPAPA algorithms offer visually accurate estimations for this signature.

D. Real Data Experiment

1) Data: In this experiment, we test the algorithms on real data. A subimage of the AVIRIS HSI data with 50×50 pixels and 181 bands (after removing the water absorption bands), covering the Moffett Field, is used. The subimage has been wildly studied in HU research, and is known to contain

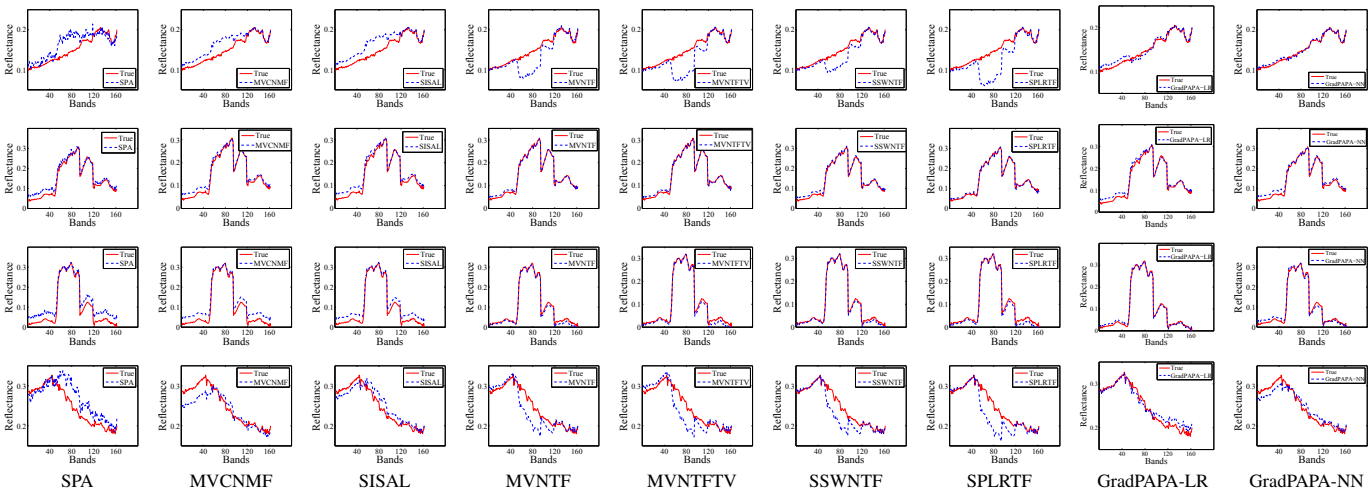


Fig. 9. The estimated spectral signatures of Urban data (SNR=30dB) by different methods. From top to bottom: `Asphalt`, `Grass`, `Tree`, and `Roof`.

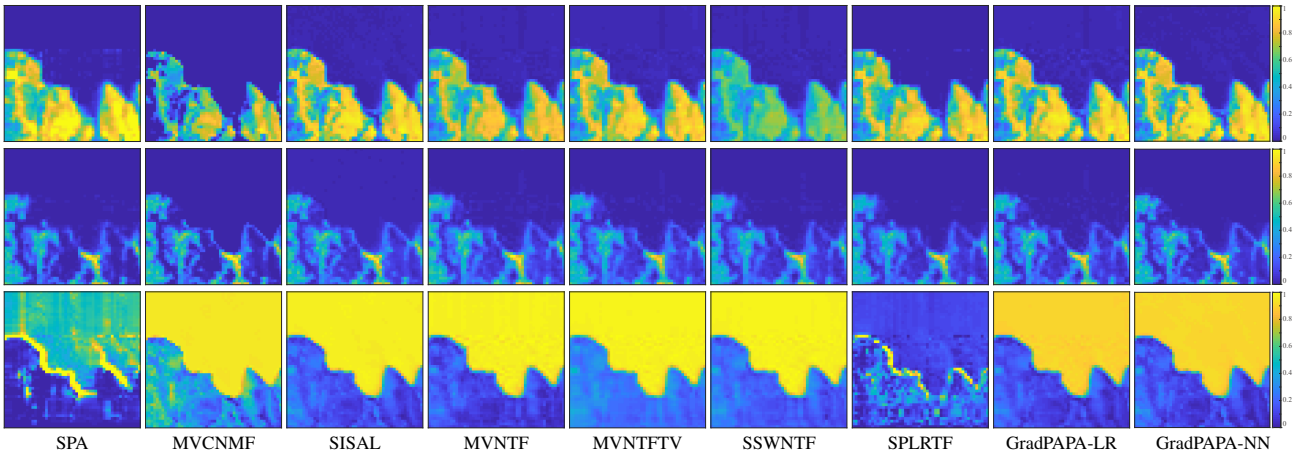


Fig. 10. The estimated abundance maps of Moffett data by different methods. From top to bottom: `Soil`, `Vegetation`, and `Water`.

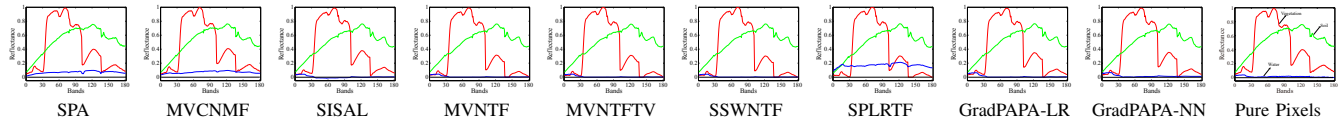


Fig. 11. The spectral signatures of manually selected pure pixels and estimated spectral signatures of Moffett data by different methods.

three types of materials—namely, `Soil`, `Vegetation`, and `Water`; see, e.g., [7].

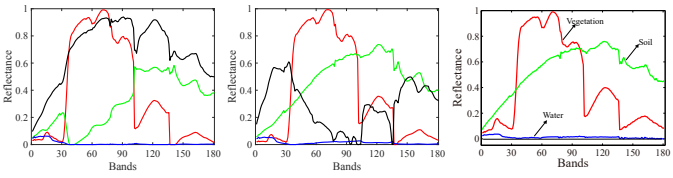
2) Baselines: We use the same baselines as those in the semi-real data experiments.

3) Results: Fig. 10 shows the estimated abundance maps. One can see that all methods produce similar maps. However, the proposed methods obtain slightly clearer boundaries among different materials (e.g. the map of `Vegetation`), and keep the smooth region of the map for `Soil` better than the ones obtained using other baselines.

Fig. 11 shows the estimated spectral signatures. For comparison, we also present the spectra of some manually selected pure pixels, which contain only one material. These pure pixels can approximately serve as the "ground truth". One can see that all baselines cannot provide accurate estimates of spectral signatures. To be specific, the spectral signatures of `Water` obtained by SPA, MVCNMF, and SPLRTF are far away from the spectra of the pure pixel. There are many negative values of the spectra of `Water` estimated by SISAL. The spectral signatures of `Vegetation` given by MVNTF, MVNTFTV, and SSWNTF, are hightly corrupted around the 30th band. Compared to the baselines, the proposed algorithms output spectra of the three materials that are clearly more similar to those of the manually picked pure pixels. The running time of all methods is listed in Table VIII. Again, the proposed GradPAPA-LR and GradPAPA-NN are at least

TABLE VIII
THE RUNNING TIME (IN SECONDS) FOR MOFFETT DATA BY ALL ALGORITHMS.

Methods	MVCNMF	SISAL	MVNTF	MVNTFTV
Time (sec.)	—	—	37.5 ± 0.6	49.2 ± 2.5
Methods	SSWNTF	SPLRTF	GradPAPA-LR	GradPAPA-NN
Time (sec.)	38.3 ± 1.4	82.0 ± 1.9	4.6 ± 0.3	1.7 ± 0.1

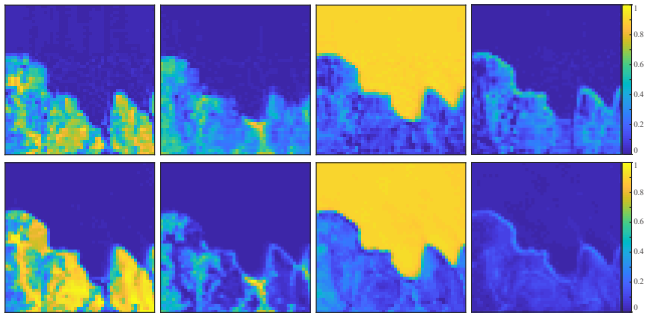


(a) GradPAPA-LR (b) GradPAPA-NN (c) Pure Pixels

Fig. 12. The spectral signatures of manually selected pure pixels and estimated spectral signatures of Moffett data by the proposed methods.

8 times and 22 times faster than the ALS-MU based LL1 baselines, respectively.

As mentioned, one can estimate the number of endmember R using existing algorithms model-order selection methods, e.g., those in [12], [51]. But sometimes the estimated R is inaccurate. Here, we demonstrate the effectiveness of the proposed algorithms for handling the case where the estimated R is slightly off. Figs. 12 and 13 show the estimated endmembers and abundance maps when $R = 4$ for Moffett Field in real data experiment, respectively. One can see that the proposed algorithms can well estimate the abundances of the three main materials. The spectra of the three materials estimated by GradPAPA-NN are similar to those of the manually picked pure pixels. The fourth material can be regarded as a *spurious endmember*, whose abundances have low intensity over the space.



(a) Soil (b) Vegetation (c) Water (d) Sparse component

Fig. 13. The estimated abundance maps of Moffett data by the proposed methods. First row: GradPAPA-LR; Second row: GradPAPA-NN.

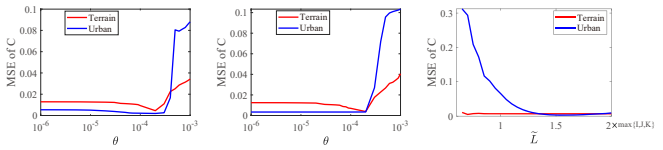
V. Discussions

In this section, we discuss the performance of the proposed algorithms under different parameters, the low-rankness of abundance maps, and the influence of selected pure pixels for the performance comparison.

A. Effect of The Parameters

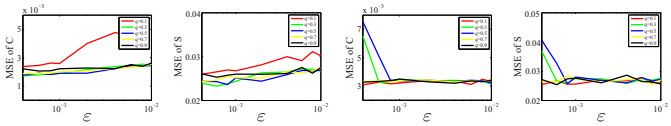
In Fig. 14, we study the sensitivity analysis of two key parameters θ and $\widetilde{L}$ on Terrain data with SNR=40dB and Urban data with SNR=30dB. One can see that the proposed approach can maintain similar low MSEs in a relatively wide range of these two parameters.

We also discuss the effect of parameters (q, ε) on Urban data with SNR=30dB. In Fig. 15, we show the impact of the different q and ε on the performance of the proposed GradPAPA. One can see that the performance change is relatively



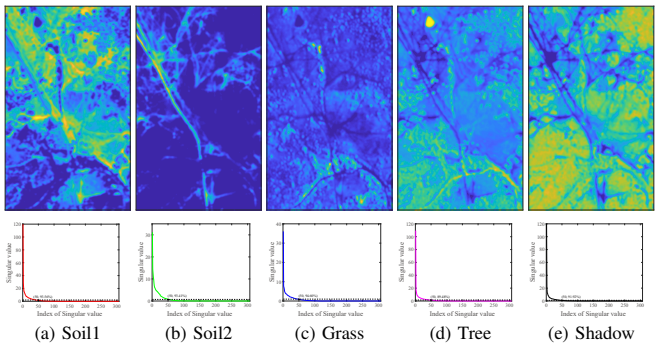
(a) θ in GradPAPA-LR (b) θ in GradPAPA-NN (c) $\widetilde{L}$ in GradPAPA-NN

Fig. 14. MSE of the estimated $\boldsymbol{C}$ under different algorithms, datasets, θ, and $\widetilde{L}$. When changing one parameter, the other one parameter is fixed to the "optimal value" as revealed in the figures.



(a) GradPAPA-LR (b) GradPAPA-LR (c) GradPAPA-NN (d) GradPAPA-NN

Fig. 15. MSEs of the estimated $\boldsymbol{C}$ and $\boldsymbol{S}$ under different q and ε.



(a) Soil1 (b) Soil2 (c) Grass (d) Tree (e) Shadow

Fig. 16. The abundance maps and the corresponding singular value curves of Terrain dataset.

insensitive to these two parameters. The other settings of the experiment here follow those in Fig. 8.

B. Low Rank of Abundance Maps

In Fig. 16, we present the singular values of the abundance maps of Terrain data. The ground-truth abundance maps are available online[1]. From Fig. 16, one can see that the first 50 principal components of $\boldsymbol{S}_r$ (first 50 left, right singular vectors and singular values) contain more than 90% of its energy. Note that the size of the abundance maps is 500×307, and the fact that most of the energy is concentrated in the first 50 principle components means that the maps are approximately low-rank. The reason why this model makes sense is that the abundance maps often exhibit correlations across the neighboring pixels (i.e., spatial correlations).

C. The Influence of Selected Pure Pixels

In our real-data experiment, the evaluation was based on manually selected pure pixels. This raises a question regarding the potential influence of using different pure pixels. Figs. 17 and 18 respectively show the RGB image of the real Moffett data and the pure pixels manually extracted from the real HSI data at different spatial locations. One can see that for one material, the corresponding manually extracted pure pixels at different spatial locations are very similar—which indicates

[1]https://github.com/LinaZhuang/NMF-QMV_demo

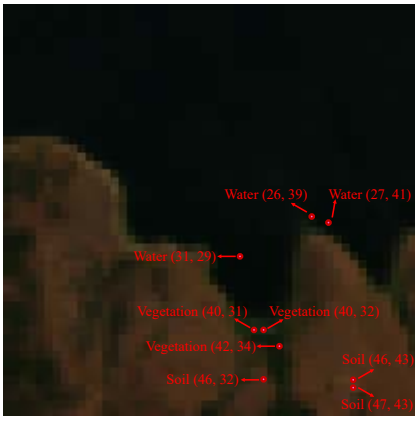
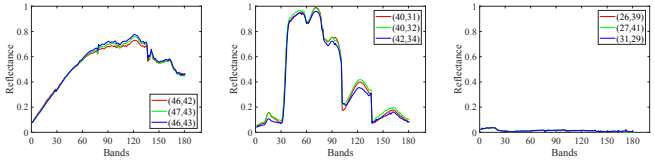


Fig. 17. The RGB image of the real Moffett data. The red dots are the visually picked pure pixels.



(a) Soil (b) Vegetation (c) Water

Fig. 18. The manually extracted pure pixels at different spatial locations.

that the selection of pixels may not have much influence on the performance comparisons.

VI. Conclusion

In this work, we proposed an algorithmic framework, namely, GradPAPA, for LL1 tensor decomposition with structural constraints and regularization arising in the context of HU. Different from existing LL1 tensor-based HU algorithms that use a three-factor parameterization and the ALS-MU type update strategies, our method utilizes a two-factor parameterization and a GP scheme. As a consequence, the proposed algorithm effectively avoids heavy computations in its iterations. To realize the GP framework, we proposed AP solvers for quickly enforcing a number of important constraints in the context of HU. We also provided custom analysis to understand the convergence properties of the proposed algorithm. Extensive experimental results on various synthetic, semi-real, and real datasets showed significant performance improvements (in terms of both accuracy and speed), compared to existing LL1 based HU algorithms. Future directions include extending the LL1 model to cover nonlinear/bilinear mixture models that are widely used in HU and to take outlying pixels into consideration.

We should mention that beyond HU, the LL1 model was also employed for many other tasks, e.g., hyperspectral super-resolution [42], [54], Electroencephalogram (EEG)/magnetic resonance imaging (MRI) analysis in medical imaging [55], fluorescence data analysis in chemometrics [31], and spectrum cartography in wireless communications [56]—and thus our algorithm design may be of broader interest.

Appendix A
The Gradient $\boldsymbol{G}_{\boldsymbol{S}}^{(t)}$ in (14)

For simplicity, we denote the objective function in (11) as

$$\mathcal{J}(\boldsymbol{C},\boldsymbol{S}) = \frac{1}{2}\|\boldsymbol{Y}-\boldsymbol{C}\boldsymbol{S}\|_F^2 + \sum_{r=1}^{R}\theta_r\varphi(\boldsymbol{S}_r).$$

Note that under the design of the smoothed 2D TV regularizer, the gradient $\boldsymbol{G}_{\boldsymbol{S}}^{(t)}$ of $\mathcal{J}(\boldsymbol{C}^{(t+1)},\boldsymbol{S})$ exists. The main idea of computing $\boldsymbol{G}_{\boldsymbol{S}}^{(t)}$ is that we first construct a tight upper bounded function $\mathcal{F}(\boldsymbol{C}^{(t+1)},\boldsymbol{S};\boldsymbol{S}^{(t)})$ such that

$$\mathcal{F}(\boldsymbol{C}^{(t+1)},\boldsymbol{S}^{(t)};\boldsymbol{S}^{(t)}) \geq \mathcal{J}(\boldsymbol{C}^{(t+1)},\boldsymbol{S}^{(t)}),$$
$$\nabla_{\boldsymbol{S}}\mathcal{F}(\boldsymbol{C}^{(t+1)},\boldsymbol{S}^{(t)};\boldsymbol{S}^{(t)}) = \nabla_{\boldsymbol{S}}\mathcal{J}(\boldsymbol{C}^{(t+1)},\boldsymbol{S}^{(t)}).$$

Then, we compute $\boldsymbol{G}_{\boldsymbol{S}}^{(t)} = \nabla_{\boldsymbol{S}}\mathcal{J}(\boldsymbol{C}^{(t+1)},\boldsymbol{S})$ through computing $\nabla_{\boldsymbol{S}}\mathcal{F}(\boldsymbol{C}^{(t+1)},\boldsymbol{S}^{(t)};\boldsymbol{S}^{(t)})$.

It is shown in [8] that $\varphi_{q,\varepsilon}(\boldsymbol{x})$ $(0<q\leq 1)$ admits a majorizer $\widetilde{\varphi}(\boldsymbol{x},\boldsymbol{x}^{(t)})$ as

$$\begin{aligned}\widetilde{\varphi}(\boldsymbol{x},\boldsymbol{x}^{(t)}) &= \sum_i [\boldsymbol{w}^{(t)}]_i[\boldsymbol{x}]_i^2 + \frac{2-q}{2}\Big(\frac{2}{q}[\boldsymbol{w}^{(t)}]_i\Big)^{\frac{q}{q-2}} + \varepsilon[\boldsymbol{w}^{(t)}]_i \\ &= \frac{q}{2}\boldsymbol{x}^\top\boldsymbol{U}^{(t)}\boldsymbol{x} + \text{const}, \end{aligned} \tag{28}$$

where $[\boldsymbol{w}^{(t)}]_i = \frac{q}{2}\big(([\boldsymbol{x}^{(t)}]_i)^2+\varepsilon\big)^{\frac{q-2}{2}}$, $\boldsymbol{U}^{(t)}$ is a diagonal matrix with $[\boldsymbol{U}^{(t)}]_{i,i} = [\boldsymbol{w}^{(t)}]_i$ and const is a constant. Therefore, we obtain the quadratic majorizer function $\mathcal{F}(\boldsymbol{C}^{(t+1)},\boldsymbol{S};\boldsymbol{S}^{(t)})$ as follows:

$$\begin{aligned}\mathcal{F}(\boldsymbol{C}^{(t+1)},\boldsymbol{S};\boldsymbol{S}^{(t)}) &= \frac{1}{2}\Big\|\boldsymbol{Y}-\boldsymbol{C}^{(t+1)}\boldsymbol{S}\Big\|_F^2 \\ &+\sum_{r=1}^{R}\big(\widetilde{\varphi}(\boldsymbol{H}_x\boldsymbol{q}_r,\boldsymbol{H}_x\boldsymbol{q}_r^{(t)}) + \widetilde{\varphi}(\boldsymbol{H}_y\boldsymbol{q}_r,\boldsymbol{H}_y\boldsymbol{q}_r^{(t)})\big). \end{aligned} \tag{29}$$

The gradient $\boldsymbol{G}_{\boldsymbol{S}}^{(t)}$ can be expressed as follows:

$$\begin{aligned}\boldsymbol{G}_{\boldsymbol{S}}^{(t)} =& (\boldsymbol{C}^{(t+1)})^\top\boldsymbol{C}^{(t+1)}\boldsymbol{S}^{(t)} - (\boldsymbol{C}^{(t+1)})^\top\boldsymbol{Y} \\ &+q[\theta_1\boldsymbol{H}_x^\top\boldsymbol{U}_1^{(t)}\boldsymbol{H}_x\boldsymbol{q}_1^{(t)},\ldots,\theta_R\boldsymbol{H}_x^\top\boldsymbol{U}_R^{(t)}\boldsymbol{H}_x\boldsymbol{q}_R^{(t)}] \\ &+q[\theta_1\boldsymbol{H}_y^\top\boldsymbol{V}_1^{(t)}\boldsymbol{H}_y\boldsymbol{q}_1^{(t)},\ldots,\theta_R\boldsymbol{H}_y^\top\boldsymbol{V}_R^{(t)}\boldsymbol{H}_y\boldsymbol{q}_R^{(t)}],\end{aligned}$$

where $[\boldsymbol{U}_r^{(t)}]_{i,i} = \big([\boldsymbol{H}_x\boldsymbol{q}_r^{(t)}]_i^2+\varepsilon\big)^{\frac{q-2}{2}}$, and $[\boldsymbol{V}_r^{(t)}]_{i,i} = \big([\boldsymbol{H}_y\boldsymbol{q}_r^{(t)}]_i^2+\varepsilon\big)^{\frac{q-2}{2}}$, $r=1,\ldots,R$.

Appendix B
Proof of Proposition 1

Before the proof, we give the following important lemma.

Lemma 1. [48, Lemma 1] Let

$$\boldsymbol{x}^{(t+1)} = \Pi_{\mathcal{X}}\left(\breve{\boldsymbol{x}} - \alpha^{(t)}\nabla\mathcal{H}(\breve{\boldsymbol{x}})\right),$$

where $\breve{\boldsymbol{x}} = \boldsymbol{x}^{(t)} + \mu^{(t)}(\boldsymbol{x}^{(t)}-\boldsymbol{x}^{(t-1)})$, $\boldsymbol{x}^{(t)}, \boldsymbol{x}^{(t-1)} \in \mathcal{X}$, $\mathcal{H}$ is proper and lower bounded on the set $\mathcal{X}$ and has Lipschitz continuous gradient $L^{(t)}$ at the current iterate $\boldsymbol{x}^{(t)}$, and $\alpha^{(t)}$ and $\mu^{(t)}$ are chosen to satisfy

$$0<\alpha^{(t)}<\infty, \quad \mu^{(t)} \leq \tau\sqrt{\big(c_1L^{(t-1)}\big)\,/\,\big(c_2L^{(t)}\big)}$$

for some $\tau < 1$, $c_1 > 0$, and $c_2 > 0$. Then, it holds that

$$\begin{aligned}&\mathcal{H}\left(\boldsymbol{x}^{(t)}\right) - \mathcal{H}\left(\boldsymbol{x}^{(t+1)}\right)\\ &\geq c_1 L^{(t)}\|\boldsymbol{x}^{(t+1)} - \boldsymbol{x}^{(t)}\|^2 - c_2\tau^2 L^{(t-1)}\|\boldsymbol{x}^{(t)} - \boldsymbol{x}^{(t-1)}\|^2.\end{aligned}$$

With Lemma 1, we now proceed to prove the theorem. The gradient of the objective function in (26) is:

$$\begin{aligned}&\partial(\mathcal{J}(\boldsymbol{Z}) + \mathcal{C}(\boldsymbol{Z}))\\ =&[(\partial_{\boldsymbol{C}}\mathcal{J}(\boldsymbol{C},\boldsymbol{S}) + \partial\mathcal{C}_{\boldsymbol{C}}(\boldsymbol{C}))^\top, (\partial_{\boldsymbol{S}}\mathcal{J}(\boldsymbol{C},\boldsymbol{S}) + \partial\mathcal{C}_{\boldsymbol{S}}(\boldsymbol{S}))^\top]^\top,\end{aligned}$$

where $\mathcal{C}_{\boldsymbol{C}}(\boldsymbol{C})$ and $\mathcal{C}_{\boldsymbol{S}}(\boldsymbol{S})$ are indicator functions of the constraints on $\boldsymbol{C}$ and $\boldsymbol{S}$, respectively.

The projections of (24) and (25) can be written as

$$\begin{aligned}\boldsymbol{C}^{(t+1)} &= \underset{\boldsymbol{C}}{\operatorname{argmin}} \frac{1}{2\alpha^{(t)}}\left\|\boldsymbol{C} - \left(\check{\boldsymbol{C}}^{(t)} - \alpha^{(t)}\boldsymbol{G}_{\check{\boldsymbol{C}}}^{(t)}\right)\right\|_F^2 + \mathcal{C}_{\boldsymbol{C}}(\boldsymbol{C}),\\ \boldsymbol{S}^{(t+1)} &= \underset{\boldsymbol{S}}{\operatorname{argmin}} \frac{1}{2\beta^{(t)}}\left\|\boldsymbol{S} - \left(\check{\boldsymbol{S}}^{(t)} - \beta^{(t)}\boldsymbol{G}_{\check{\boldsymbol{S}}}^{(t)}\right)\right\|_F^2 + \mathcal{C}_{\boldsymbol{S}}(\boldsymbol{S}).\end{aligned}$$

According to the first-order optimality of $\boldsymbol{C}^{(t+1)}$-subproblem and $\boldsymbol{S}^{(t+1)}$-subproblem, we have

$$\begin{aligned}\mathbf{0} &\in \frac{1}{\alpha^{(t)}}\left(\boldsymbol{C}^{(t+1)} - \check{\boldsymbol{C}}^{(t)}\right) + \boldsymbol{G}_{\check{\boldsymbol{C}}}^{(t)} + \partial\mathcal{C}_{\boldsymbol{C}}(\boldsymbol{C}),\\ \mathbf{0} &\in \frac{1}{\beta^{(t)}}\left(\boldsymbol{S}^{(t+1)} - \check{\boldsymbol{S}}^{(t)}\right) + \boldsymbol{G}_{\check{\boldsymbol{S}}}^{(t)} + \partial\mathcal{C}_{\boldsymbol{S}}(\boldsymbol{S}).\end{aligned}$$

Let $\boldsymbol{V}_{\boldsymbol{C}}^{(t+1)} \in \partial\mathcal{C}_{\boldsymbol{C}}(\boldsymbol{C})$ and $\boldsymbol{V}_{\boldsymbol{S}}^{(t+1)} \in \partial\mathcal{C}_{\boldsymbol{S}}(\boldsymbol{S})$, we have

$$\begin{aligned}\mathbf{0} &= \frac{1}{\alpha^{(t)}}\left(\boldsymbol{C}^{(t+1)} - \check{\boldsymbol{C}}^{(t)}\right) + \boldsymbol{G}_{\check{\boldsymbol{C}}}^{(t)} + \boldsymbol{V}_{\boldsymbol{C}}^{(t+1)},\\ \mathbf{0} &= \frac{1}{\beta^{(t)}}\left(\boldsymbol{S}^{(t+1)} - \check{\boldsymbol{S}}^{(t)}\right) + \boldsymbol{G}_{\check{\boldsymbol{S}}}^{(t)} + \boldsymbol{V}_{\boldsymbol{S}}^{(t+1)},\end{aligned}$$

or equivalently,

$$\begin{aligned}&\partial_{\boldsymbol{C}}(\mathcal{J}(\boldsymbol{C}^{(t+1)},\boldsymbol{S}^{(t)})) + \boldsymbol{V}_{\boldsymbol{C}}^{(t+1)}\\ =&\partial_{\boldsymbol{C}}(\mathcal{J}(\boldsymbol{C}^{(t+1)},\boldsymbol{S}^{(t)})) - \frac{1}{\alpha^{(t)}}\left(\boldsymbol{C}^{(t+1)} - \check{\boldsymbol{C}}^{(t)}\right) - \boldsymbol{G}_{\check{\boldsymbol{C}}}^{(t)},\\ &\partial_{\boldsymbol{S}}(\mathcal{J}(\boldsymbol{C}^{(t+1)},\boldsymbol{S}^{(t+1)})) + \boldsymbol{V}_{\boldsymbol{S}}^{(t+1)}\\ =&\partial_{\boldsymbol{S}}(\mathcal{J}(\boldsymbol{C}^{(t+1)},\boldsymbol{S}^{(t+1)})) - \frac{1}{\beta^{(t)}}\left(\boldsymbol{S}^{(t+1)} - \check{\boldsymbol{S}}^{(t)}\right) + \boldsymbol{G}_{\check{\boldsymbol{S}}}^{(t)},\end{aligned} \tag{30}$$

Note that $\boldsymbol{Z} = (\boldsymbol{C},\boldsymbol{S})$, then

$$\begin{aligned}&\operatorname{dist}\left(\mathbf{0}, \partial\left(\mathcal{J}\left(\boldsymbol{Z}^{(t+1)}\right) + \mathcal{C}\left(\boldsymbol{Z}^{(t+1)}\right)\right)\right)\\ \leq&\operatorname{dist}\left(\mathbf{0}, \partial_{\boldsymbol{C}}\left(\mathcal{J}\left(\boldsymbol{C}^{(t+1)},\boldsymbol{S}^{(t)}\right) + \mathcal{C}\left(\boldsymbol{C}^{(t+1)}\right)\right)\right)\\ +&\operatorname{dist}\left(\mathbf{0}, \partial_{\boldsymbol{S}}\left(\mathcal{J}\left(\boldsymbol{C}^{(t+1)},\boldsymbol{S}^{(t+1)}\right) + \mathcal{C}\left(\boldsymbol{S}^{(t+1)}\right)\right)\right)\\ \leq&\left\|\partial_{\boldsymbol{C}}(\mathcal{J}(\boldsymbol{C}^{(t+1)},\boldsymbol{S}^{(t)})) + \boldsymbol{V}_{\boldsymbol{C}}^{(t+1)}\right\|_F\\ +&\left\|\partial_{\boldsymbol{S}}(\mathcal{J}(\boldsymbol{C}^{(t+1)},\boldsymbol{S}^{(t+1)})) + \boldsymbol{V}_{\boldsymbol{S}}^{(t+1)}\right\|_F\\ \overset{(a)}{=}&\left\|\frac{1}{\alpha^{(t)}}\left(\boldsymbol{C}^{(t+1)} - \check{\boldsymbol{C}}^{(t)}\right) + \boldsymbol{G}_{\check{\boldsymbol{C}}}^{(t)} - \partial_{\boldsymbol{C}}(\mathcal{J}(\boldsymbol{C}^{(t+1)},\boldsymbol{S}^{(t)}))\right\|_F\\ +&\left\|\frac{1}{\beta^{(t)}}\left(\boldsymbol{S}^{(t+1)} - \check{\boldsymbol{S}}^{(t)}\right) + \boldsymbol{G}_{\check{\boldsymbol{S}}}^{(t)} - \partial_{\boldsymbol{S}}(\mathcal{J}(\boldsymbol{C}^{(t+1)},\boldsymbol{S}^{(t+1)}))\right\|_F\\ \leq&\frac{1}{\alpha^{(t)}}\left\|\boldsymbol{C}^{(t+1)} - \check{\boldsymbol{C}}^{(t)}\right\|_F + \|\boldsymbol{G}_{\check{\boldsymbol{C}}}^{(t)} - \partial_{\boldsymbol{C}}(\mathcal{J}(\boldsymbol{C}^{(t+1)},\boldsymbol{S}^{(t)}))\|_F\\ +&\frac{1}{\beta^{(t)}}\left\|\boldsymbol{S}^{(t+1)} - \check{\boldsymbol{S}}^{(t)}\right\| + \left\|\boldsymbol{G}_{\check{\boldsymbol{S}}}^{(t)} - \partial_{\boldsymbol{S}}(\mathcal{J}(\boldsymbol{C}^{(t+1)},\boldsymbol{S}^{(t+1)}))\right\|_F,\end{aligned} \tag{31}$$

where we have applied (30) to get (a). Now, we analyze the last fours terms in (31). First,

$$\begin{aligned}&\left\|\boldsymbol{C}^{(t+1)} - \check{\boldsymbol{C}}^{(t)}\right\|_F\\ =&\left\|\boldsymbol{C}^{(t)} - \boldsymbol{C}^{(t+1)} + \mu_1^{(t)}(\boldsymbol{C}^{(t)} - \boldsymbol{C}^{(t-1)})\right\|_F\\ \leq&\ \mu_1^{(t)}\left\|\boldsymbol{C}^{(t)} - \boldsymbol{C}^{(t-1)}\right\|_F + \left\|\boldsymbol{C}^{(t)} - \boldsymbol{C}^{(t+1)}\right\|_F.\end{aligned} \tag{32}$$

Second,

$$\begin{aligned}&\left\|\boldsymbol{G}_{\check{\boldsymbol{C}}}^{(t)} - \partial_{\boldsymbol{C}}(\mathcal{J}(\boldsymbol{C}^{(t+1)},\boldsymbol{S}^{(t)}))\right\|_F\\ =&\left\|\left(\check{\boldsymbol{C}}^{(t)} - \boldsymbol{C}^{(t+1)}\right)\boldsymbol{S}^{(t)}\left(\boldsymbol{S}^{(t)}\right)^\top\right\|_F\\ \leq&\ L_{\boldsymbol{C}}^{(t)}\left\|\check{\boldsymbol{C}}^{(t)} - \boldsymbol{C}^{(t+1)}\right\|_F\\ \leq&\ L_{\boldsymbol{C}}^{(t)}\mu_1^{(t)}\left\|\boldsymbol{C}^{(t)} - \boldsymbol{C}^{(t-1)}\right\|_F + L_{\boldsymbol{C}}^{(t)}\left\|\boldsymbol{C}^{(t)} - \boldsymbol{C}^{(t+1)}\right\|_F,\end{aligned} \tag{33}$$

where the first inequality is due to $L_{\boldsymbol{C}}^{(t)} = \sigma_{\max}^2\left(\boldsymbol{S}^{(t)}\right)$; the second inequality uses $\check{\boldsymbol{C}}^{(t)} = \boldsymbol{C}^{(t)} + \mu_1^{(t)}(\boldsymbol{C}^{(t)} - \boldsymbol{C}^{(t-1)})$. Similarly, one can get

$$\begin{aligned}&\left\|\boldsymbol{S}^{(t+1)} - \check{\boldsymbol{S}}^{(t)}\right\|_F\\ \leq&\ \mu_2^{(t)}\left\|\boldsymbol{S}^{(t)} - \boldsymbol{S}^{(t-1)}\right\|_F + \left\|\boldsymbol{S}^{(t)} - \boldsymbol{S}^{(t+1)}\right\|_F,\end{aligned} \tag{34}$$

and

$$\begin{aligned}&\left\|\boldsymbol{G}_{\check{\boldsymbol{S}}}^{(t)} - \partial_{\boldsymbol{S}}(\mathcal{J}(\boldsymbol{C}^{(t+1)},\boldsymbol{S}^{(t+1)}))\right\|_F\\ \leq&\ L_{\boldsymbol{S}}^{(t)}\mu_2^{(t)}\left\|\boldsymbol{S}^{(t)} - \boldsymbol{S}^{(t-1)}\right\|_F + L_{\boldsymbol{S}}^{(t)}\left\|\boldsymbol{S}^{(t)} - \boldsymbol{S}^{(t+1)}\right\|_F.\end{aligned} \tag{35}$$

Combining the results in (31)-(34), we have

$$\begin{aligned}&\operatorname{dist}\left(\mathbf{0}, \partial\left(\mathcal{J}\left(\boldsymbol{Z}^{(t+1)}\right) + \mathcal{C}\left(\boldsymbol{Z}^{(t+1)}\right)\right)\right)\\ \leq&\ \mu_1^{(t)}\left(L_{\boldsymbol{C}}^{(t)} + \frac{1}{\alpha^{(t)}}\right)\left\|\boldsymbol{C}^{(t)} - \boldsymbol{C}^{(t-1)}\right\|_F\\ &+\ \left(L_{\boldsymbol{C}}^{(t)} + \frac{1}{\alpha^{(t)}}\right)\left\|\boldsymbol{C}^{(t)} - \boldsymbol{C}^{(t+1)}\right\|_F\\ &+\ \mu_2^{(t)}\left(L_{\boldsymbol{S}}^{(t)} + \frac{1}{\beta^{(t)}}\right)\left\|\boldsymbol{S}^{(t)} - \boldsymbol{S}^{(t-1)}\right\|_F\\ &+\ \left(L_{\boldsymbol{S}}^{(t)} + \frac{1}{\beta^{(t)}}\right)\left\|\boldsymbol{S}^{(t)} - \boldsymbol{S}^{(t+1)}\right\|_F\\ \leq&\ C_1\left(\left\|\boldsymbol{Z}^{(t)} - \boldsymbol{Z}^{(t-1)}\right\|_F + \left\|\boldsymbol{Z}^{(t)} - \boldsymbol{Z}^{(t+1)}\right\|_F\right),\end{aligned} \tag{36}$$

where

$$C_1 = \max\{\bar{\mu}_1, \bar{\mu}_2, 1\}\max\left\{(c_1{+}1)\sup_t \alpha^{(t)}, (c_3{+}1)\sup_t \beta^{(t)}\right\};$$

Note that the last inequality is due to $\mu_1^{(t)} \leq \bar{\mu}_1$, $\mu_2^{(t)} \leq \bar{\mu}_2$, $1/\alpha^{(t)} \leq c_1 L_{\boldsymbol{C}}^{(t)}$, and $1/\beta^{(t)} \leq c_3 L_{\boldsymbol{S}}^{(t)}$.

According to the update rules of $\boldsymbol{C}$ and $\boldsymbol{S}$, we have

$$\begin{aligned}&\mathcal{J}\left(\boldsymbol{C}^{(t)},\boldsymbol{S}^{(t)}\right) - \mathcal{J}\left(\boldsymbol{C}^{(t+1)},\boldsymbol{S}^{(t)}\right)\\ \geq&\ c_1 L_{\boldsymbol{C}}^{(t)}\left\|\boldsymbol{C}^{(t+1)} - \boldsymbol{C}^{(t)}\right\|_F^2 - c_2\tau_1^2 L_{\boldsymbol{C}}^{(t-1)}\left\|\boldsymbol{C}^{(t)} - \boldsymbol{C}^{(t-1)}\right\|_F^2\\ &\mathcal{J}\left(\boldsymbol{C}^{(t+1)},\boldsymbol{S}^{(t)}\right) - \mathcal{J}\left(\boldsymbol{C}^{(t+1)},\boldsymbol{S}^{(t+1)}\right)\\ \geq&\ c_3 L_{\boldsymbol{S}}^{(t)}\left\|\boldsymbol{S}^{(t+1)} - \boldsymbol{S}^{(t)}\right\|_F^2 - c_4\tau_2^2 L_{\boldsymbol{S}}^{(t-1)}\left\|\boldsymbol{S}^{(t)} - \boldsymbol{S}^{(t-1)}\right\|_F^2,\end{aligned}$$

where the equations are due to Lemma 1 with $\mathcal{H} = \mathcal{J}\left(\cdot, \boldsymbol{S}^{(t)}\right)$ (first inequality) and $\mathcal{H} = \mathcal{F}\left(\boldsymbol{C}^{(t+1)}, \cdot; \boldsymbol{S}^{(t)}\right)$ (second inequality).

Now, combining Lemma 1 and (36), we can use the proof technique in [50] to establish the final convergence rate:

$$
\begin{aligned}
&\mathcal{J}\left(\boldsymbol{Z}^{(0)}\right) - \mathcal{J}\left(\boldsymbol{Z}^{(t+1)}\right) \\
&= \sum_{t'=0}^{t} \mathcal{J}\left(\boldsymbol{Z}^{(t')}\right) - \mathcal{J}\left(\boldsymbol{Z}^{(t'+1)}\right) \\
&\geq \sum_{t'=0}^{t} c_1 L_{\boldsymbol{C}}^{(t')} \left\|\boldsymbol{C}^{(t'+1)} - \boldsymbol{C}^{(t')}\right\|_F^2 \\
&\quad - c_2 \tau_1^2 L_{\boldsymbol{C}}^{(t'-1)} \left\|\boldsymbol{C}^{(t')} - \boldsymbol{C}^{(t'-1)}\right\|_F^2 \\
&\quad + \sum_{t'=0}^{t} c_3 L_{\boldsymbol{S}}^{(t')} \left\|\boldsymbol{S}^{(t'+1)} - \boldsymbol{S}^{(t')}\right\|_F^2 \\
&\quad - c_4 \tau_2^2 L_{\boldsymbol{S}}^{(t'-1)} \left\|\boldsymbol{S}^{(t')} - \boldsymbol{S}^{(t'-1)}\right\|_F^2 \\
&= \sum_{t'=0}^{t-1} \left(c_1 - c_2\tau_1^2\right) L_{\boldsymbol{C}}^{(t')} \left\|\boldsymbol{C}^{(t'+1)} - \boldsymbol{C}^{(t')}\right\|_F^2 \\
&\quad + \sum_{t'=0}^{t-1} \left(c_3 - c_4\tau_2^2\right) L_{\boldsymbol{C}}^{(t')} \left\|\boldsymbol{S}^{(t'+1)} - \boldsymbol{S}^{(t')}\right\|_F^2 \\
&\quad + c_1 L_{\boldsymbol{C}}^{(t)} \left\|\boldsymbol{C}^{(t+1)} - \boldsymbol{C}^{(t)}\right\|_F^2 + c_3 L_{\boldsymbol{S}}^{(t)} \left\|\boldsymbol{S}^{(t+1)} - \boldsymbol{S}^{(t)}\right\|_F^2 \\
&\geq \sum_{t'=0}^{t} \left(c_1 - c_2\tau_1^2\right) L_{\boldsymbol{C}}^{(t')} \left\|\boldsymbol{C}^{(t'+1)} - \boldsymbol{C}^{(t')}\right\|_F^2 \\
&\quad + \sum_{t'=0}^{t} \left(c_3 - c_4\tau_2^2\right) L_{\boldsymbol{S}}^{(t')} \left\|\boldsymbol{S}^{(t'+1)} - \boldsymbol{S}^{(t')}\right\|_F^2 \\
&\geq \sum_{t'=0}^{t} \frac{1-\tau_1^2}{\sup_t \alpha^{(t)}} \left\|\boldsymbol{C}^{(t'+1)} - \boldsymbol{C}^{(t')}\right\|_F^2 \\
&\quad + \sum_{t'=0}^{t} \frac{1-\tau_2^2}{\sup_t \beta^{(t)}} \left\|\boldsymbol{S}^{(t'+1)} - \boldsymbol{S}^{(t')}\right\|_F^2 \\
&\geq \sum_{t'=0}^{t} C_2 \left\|\boldsymbol{Z}^{(t'+1)} - \boldsymbol{Z}^{(t')}\right\|_F^2,
\end{aligned}
$$

where $C_2 = \min\left\{(1-\tau_1^2)/\sup_t \alpha^{(t)}, (1-\tau_2^2)/\sup_t \beta^{(t)}\right\}$, and the penultimate inequality is due to $c_2 L_{\boldsymbol{C}}^{(t)} \leq 1/\alpha^{(t)} \leq c_1 L_{\boldsymbol{C}}^{(t)}$, and $c_4 L_{\boldsymbol{S}}^{(t)} \leq 1/\beta^{(t)} \leq c_3 L_{\boldsymbol{S}}^{(t)}$.

From the above equation, we get

$$
\begin{aligned}
&\mathcal{J}\left(\boldsymbol{Z}^{(0)}\right) - \mathcal{J}^\star \\
&\geq \mathcal{J}\left(\boldsymbol{Z}^{(0)}\right) - \mathcal{J}\left(\boldsymbol{Z}^{(t+1)}\right) \\
&\geq C_2 \frac{t}{2} \min_{t'=0,1,\ldots,t} \left\|\boldsymbol{Z}^{(t'+1)} - \boldsymbol{Z}^{(t')}\right\|_F^2 + \left\|\boldsymbol{Z}^{(t')} - \boldsymbol{Z}^{(t'+1)}\right\|_F^2.
\end{aligned}
$$

By using $a + b \leq \sqrt{2(a^2+b^2)}$, we have

$$
\begin{aligned}
&\min_{t'=0,1,\ldots,t} \left\|\boldsymbol{Z}^{(t'+1)} - \boldsymbol{Z}^{(t')}\right\|_F + \left\|\boldsymbol{Z}^{(t')} - \boldsymbol{Z}^{(t'+1)}\right\|_F \\
&\leq \sqrt{\frac{4}{C_2 t}\left(\mathcal{J}\left(\boldsymbol{Z}^{(0)}\right) - \mathcal{J}^\star\right)}. \qquad (37)
\end{aligned}
$$

Substituting (37) into (36) yields

$$
\begin{aligned}
&\min_{t'=0,1,\ldots,t} \operatorname{dist}\left(\mathbf{0}, \partial\left(\mathcal{J}\left(\boldsymbol{Z}^{(t'+1)}\right) + \mathcal{C}\left(\boldsymbol{Z}^{(t'+1)}\right)\right)\right) \\
&\leq C_1 \sqrt{\frac{4}{C_2 t}\left(\mathcal{J}\left(\boldsymbol{Z}^{(0)}\right) - \mathcal{J}^\star\right)}.
\end{aligned}
$$

This completes the proof.

REFERENCES

[1] J. M. Bioucas-Dias, A. Plaza, N. Dobigeon, M. Parente, Q. Du, P. Gader, and J. Chanussot, "Hyperspectral unmixing overview: Geometrical, statistical, and sparse regression-based approaches," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 5, no. 2, pp. 354–379, 2012.

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

[3] J. M. P. Nascimento and J. M. Bioucas-Dias, "Vertex component analysis: a fast algorithm to unmix hyperspectral data," *IEEE Trans. Geosci. Remote Sens.*, vol. 43, no. 4, pp. 898–910, 2005.

[4] T.-H. Chan, W.-K. Ma, A. Ambikapathi, and C.-Y. Chi, "A simplex volume maximization framework for hyperspectral endmember extraction," *IEEE Trans. Geosci. Remote Sens.*, vol. 49, no. 11, pp. 4177–4193, 2011.

[5] J. M. Bioucas-Dias, "A variable splitting augmented Lagrangian approach to linear spectral unmixing," in *Proc. WHISPERS*, 2009, pp. 1–4.

[6] L. Miao and H. Qi, "Endmember extraction from highly mixed data using minimum volume constrained nonnegative matrix factorization," *IEEE Trans. Geosci. Remote Sens.*, vol. 45, no. 3, pp. 765–777, 2007.

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

[8] X. Fu, W.-K. Ma, J. M. Bioucas-Dias, and T.-H. Chan, "Semiblind hyperspectral unmixing in the presence of spectral library mismatches," *IEEE Trans. Geosci. Remote Sens.*, vol. 54, no. 9, pp. 5171–5184, 2016.

[9] M. D. Craig, "Minimum-volume transforms for remotely sensed data," *IEEE Trans. Geosci. Remote Sens.*, vol. 32, no. 3, pp. 542–552, 1994.

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

[11] M. E. Winter, "N-FINDR: An algorithm for fast autonomous spectral end-member determination in hyperspectral data," in *Proc. SPIE*, vol. 3753, 1999, pp. 266–275.

[12] X. Fu, W.-K. Ma, T.-H. Chan, and J. M. Bioucas-Dias, "Self-dictionary sparse regression for hyperspectral unmixing: Greedy pursuit and pure pixel search are related," *IEEE J. Sel. Topics Signal Process.*, vol. 9, no. 6, pp. 1128–1141, 2015.

[13] T. Nguyen, X. Fu, and R. Wu, "Memory-efficient convex optimization for self-dictionary separable nonnegative matrix factorization: A frank-wolfe approach," *Arxiv preprint: arXiv:2109.11135*, 2021.

[14] C.-H. Lin, W.-K. Ma, W.-C. Li, C.-Y. Chi, and A. Ambikapathi, "Identifiability of the simplex volume minimization criterion for blind hyperspectral unmixing: The no-pure-pixel case," *IEEE Trans. Geosci. Remote Sens.*, vol. 53, no. 10, pp. 5530–5546, 2015.

[15] G. Martín and J. M. Bioucas-Dias, "Hyperspectral blind reconstruction from random spectral projections," *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, vol. 9, no. 6, pp. 2390–2399, 2016.

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

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

[18] N. Gillis, "Successive nonnegative projection algorithm for robust nonnegative blind source separation," *SIAM J. Imaging Sci.*, vol. 7, no. 2, pp. 1420–1450, 2014.

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

[20] X. Fu, K. Huang, and N. D. Sidiropoulos, "On identifiability of nonnegative matrix factorization," *IEEE Signal Process. Lett.*, vol. 25, no. 3, pp. 328–332, 2018.

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

[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 Sens.*, vol. 55, no. 3, pp. 1776–1792, 2017.

[23] 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, 2019.

[24] H.-C. Li, S. Liu, X.-R. Feng, and S.-Q. Zhang, "Sparsity-constrained coupled nonnegative matrix–tensor factorization for hyperspectral unmixing," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 13, pp. 5061–5073, 2020.

[25] H.-C. Li, S. Liu, X.-R. Feng, R. Wang, and Y.-J. Sun, "Double weighted sparse nonnegative tensor factorization for hyperspectral unmixing," *Int. J. Remote Sens.*, vol. 42, no. 8, pp. 3180–3191, 2021.

[26] S. Zhang, G. Zhang, C. Deng, J. Li, S. Wang, J. Wang, and A. Plaza, "Spectral-spatial weighted sparse nonnegative tensor factorization for hyperspectral unmixing," in *Proc. IGARSS*, 2020, pp. 2177–2180.

[27] P. Zheng, H. Su, and Q. Du, "Sparse and low-rank constrained tensor factorization for hyperspectral image unmixing," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 14, pp. 1754–1767, 2021.

[28] F. Xiong, K. Qian, J. Lu, J. Zhou, and Y. Qian, "Nonlocal low-rank nonnegative tensor factorization for hyperspectral unmixing," in *Proc. IGARSS*, 2020, pp. 2157–2160.

[29] 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.

[30] 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, no. 3, pp. 1067–1083, 2008.

[31] 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, 2015.

[32] D. D. Lee and H. S. Seung, "Algorithms for non-negative matrix factorization," in *Proc. NeurIPS*, 2001, pp. 556–562.

[33] C.-J. Lin, "On the convergence of multiplicative update algorithms for nonnegative matrix factorization," *IEEE Trans. Neural Networks*, vol. 18, no. 6, pp. 1589–1596, 2007.

[34] N. Gillis and F. Glineur, "Accelerated multiplicative updates and hierarchical ALS algorithms for nonnegative matrix factorization," *Neural Comput.*, vol. 24, no. 4, pp. 1085–1105, 2012.

[35] X. Fu and K. Huang, "Block-term tensor decomposition via constrained matrix factorization," in *Proc. MLSP*, 2019, pp. 1–6.

[36] Y. Xu and W. Yin, "Block stochastic gradient iteration for convex and nonconvex optimization," *SIAM J. Optimiz.*, vol. 25, no. 3, pp. 1686–1716, 2015.

[37] M. Ding, X. Fu, T.-Z. Huang, and X.-L. Zhao, "Constrained block-term tensor decomposition-based hyperspectral unmixing via alternating gradient projection," in *Proc. EUSIPCO*, 2021, pp. 1060–1064.

[38] J. M. Nascimento and J. M. Bioucas-Dias, "Hyperspectral unmixing based on mixtures of dirichlet components," *IEEE Trans. Geosci. Remote Sens.*, vol. 50, no. 3, pp. 863–878, 2012.

[39] N. D. Sidiropoulos, L. De Lathauwer, X. Fu, K. Huang, E. E. Papalexakis, and C. Faloutsos, "Tensor decomposition for signal processing and machine learning," *IEEE Trans. Signal Process.*, vol. 65, no. 13, pp. 3551–3582, 2017.

[40] X. Fu, C. Gao, H.-T. Wai, and K. Huang, "Block-randomized stochastic proximal gradient for low-rank tensor factorization," *IEEE Trans. Signal Process.*, vol. 68, pp. 2170–2185, 2010.

[41] M. Razaviyayn, M. Hong, and Z.-Q. Luo, "A unified convergence analysis of block successive minimization methods for nonsmooth optimization," *SIAM J. Optimiz.*, vol. 23, no. 2, pp. 1126–1153, 2013.

[42] M. Ding, X. Fu, T.-Z. Huang, J. Wang, and X.-L. Zhao, "Hyperspectral super-resolution via interpretable block-term tensor modeling," *IEEE J. Sel. Topics Signal Process.*, vol. 15, no. 3, pp. 641–656, 2021.

[43] G. H. Golub and C. F. Van Loan, *Matrix computations*. JHU Press, 2012, vol. 3.

[44] D. Garber, "On the convergence of projected-gradient methods with low-rank projections for smooth convex minimization over trace-norm balls and related problems," *SIAM J. Optimiz.*, vol. 31, no. 1, pp. 727–753, 2021.

[45] Y. Nesterov, "A method for solving the convex programming problem with convergence rate $O(1/k^2)$," *Dokl. Akad. Nauk SSSR*, vol. 269, pp. 543–547, 1983.

[46] Y. Xu and W. Yin, "A block coordinate descent method for regularized multiconvex optimization with applications to nonnegative tensor factorization and completion," *SIAM J. Imaging Sci.*, vol. 6, no. 3, pp. 1758–1789, 2013.

[47] U. Araújo, B. Saldanha, R. Galvão, T. Yoneyama, H. Chame, and V. Visani, "The successive projections algorithm for variable selection in spectroscopic multicomponent analysis," *Chemometr. Intell. Lab. Syst.*, vol. 57, no. 2, pp. 65–73, 2001.

[48] Y. Xu and W. Yin, "A globally convergent algorithm for nonconvex optimization based on block coordinate update," *J. Sci. Comput.*, vol. 72, no. 2, pp. 700–734, 2017.

[49] H. Attouch, J. Bolte, P. Redont, and A. Soubeyran, "Proximal alternating minimization and projection methods for nonconvex problems: An approach based on the kurdyka-Łojasiewicz inequality," *Math. Oper. Res.*, vol. 35, no. 2, pp. 438–457, 2010.

[50] M. Shao, Q. Li, W.-K. Ma, and A. M.-C. So, "A framework for one-bit and constant-envelope precoding over multiuser massive miso channels," *IEEE Trans. Signal Process.*, vol. 67, no. 20, pp. 5309–5324, 2019.

[51] J. M. Bioucas-Dias and J. M. P. Nascimento, "Hyperspectral subspace identification," *IEEE Trans. Geosci. Remote Sens.*, vol. 46, no. 8, pp. 2435–2445, 2008.

[52] J. Kim, Y. He, and H. Park, "Algorithms for nonnegative matrix and tensor factorizations: A unified view based on block coordinate descent framework," *J. Glob. Optim.*, vol. 58, no. 2, pp. 285–319, 2014.

[53] L. Zhuang, C. Lin, M. A. T. Figueiredo, and J. M. Bioucas-Dias, "Regularization parameter selection in minimum volume hyperspectral unmixing," *IEEE Trans. Geosci. Remote Sens.*, vol. 57, no. 12, pp. 9858–9877, 2019.

[54] C. Prévost, R. Borsoi, K. Usevich, D. Brie, J. C. Bermudez, and C. Richard, "Hyperspectral super-resolution accounting for spectral variability: coupled tensor LL1-based recovery and blind unmixing of the unknown super-resolution image," *SIAM J. Imaging Sci.*, vol. 15, no. 1, pp. 110–138, 2021.

[55] C. Stamile, F. Cotton, D. Sappey-Marinier, and S. Van Huffel, "Tensor based blind source separation in longitudinal magnetic resonance imaging analysis," in *Proc. IEEE EMBC*, 2019, pp. 3879–3883.

[56] G. Zhang, X. Fu, J. Wang, X.-L. Zhao, and M. Hong, "Spectrum cartography via coupled block-term tensor decomposition," *IEEE Trans. Signal Process.*, vol. 68, pp. 3660–3675, 2020.

Meng Ding received the Ph.D. degree from the University of Electronic Science and Technology of China, Chengdu, China, in 2021. From 2019 to 2020, he was an exchange Ph.D. Student at Oregon State University, Corvallis, OR, USA, supported by the China Scholarship Council. He is currently an Assistant Professor with the School of Mathematics, Southwest Jiaotong University, Chengdu. His current research interests include image processing, tensor analysis, and machine learning.

Xiao Fu (Senior Member, IEEE) received the B.Eng. and M.Sc. degrees from the University of Electronic Science and Technology of China, Chengdu, China, in 2005 and 2010, respectively, and the Ph.D. degree in Electronic Engineering from The Chinese University of Hong Kong, Shatin, N.T., Hong Kong, in 2014. From 2014 to 2017, he was a Postdoctoral Associate with the Department of Electrical and Computer Engineering, University of Minnesota - Twin Cities, Minneapolis, MN, USA. Since 2017, he has been an Assistant Professor with the School of Electrical Engineering and Computer Science, Oregon State University, Corvallis, OR, USA. His research interests include signal processing and machine learning.

Dr. Fu was the recipient of the Best Student Paper Award at ICASSP 2014, the 2022 IEEE Signal Processing Society (SPS) Best Paper Award, and the 2022 IEEE SPS Donald G. Fink Overview Paper Award. He received the Outstanding Postdoctoral Scholar Award at University of Minnesota in 2016, and the National Science Foundation Early Career Development Program Award (NSF CAREER Award) in 2022. He is a Member of the Sensor Array and Multichannel Technical Committee (SAM-TC) and the Signal Processing Theory and Method Technical Committee (SPTM-TC). He was a Tutorial Speaker at ICASSP 2017 and SIAM Conference on Applied Linear Algebra 2021. He is an Associate Editor of IEEE TRANSACTIONS ON AN EDITOR OF SIGNAL PROCESSING and SIGNAL PROCESSING, Elsevier.

Xi-Le Zhao received the M.S. and Ph.D. degrees from the University of Electronic Science and Technology of China (UESTC), Chengdu, China, in 2009 and 2012, respectively. He is currently a Professor with the School of Mathematical Sciences, UESTC. His research interest mainly focuses on model- and data-driven methods for image processing problems. His homepage https://zhaoxile.github.io/.