Joint-Sparse-Blocks and Low-Rank Representation for Hyperspectral Unmixing

Jie Huang, Ting-Zhu Huang, Liang-Jian Deng, and Xi-Le Zhao

***Abstract*— Hyperspectral unmixing has attracted much attention in recent years. Single sparse unmixing assumes that a pixel in a hyperspectral image consists of a relatively small number of spectral signatures from large, ever-growing, and available spectral libraries. Joint-sparsity (or row-sparsity) model typically enforces all pixels in a neighborhood to share the same set of spectral signatures. The two sparse models are widely used in the literature. In this paper, we propose a joint-sparsity-blocks model for abundance estimation problem. Namely, the abundance matrix of size $m \times n$ is partitioned to have one row block and s column blocks and each column block itself is joint-sparse. It generalizes both the single (i.e., $s = n$) and the joint (i.e., $s = 1$) sparsities. Moreover, concatenating the proposed joint-sparsity-blocks structure and low rankness assumption on the abundance coefficients, we develop a new algorithm called *joint-sparse-blocks and low-rank unmixing*. In particular, for the joint-sparse-blocks regression problem, we develop a two-level reweighting strategy to enhance the sparsity along the rows within each block. Simulated and real-data experiments demonstrate the effectiveness of the proposed algorithm.**

***Index Terms*— Abundance estimation, hyperspectral images (HSIs), joint-sparse-blocks regression, low-rank matrix, spectral unmixing.**

I. Introduction

SPECTRAL unmixing for hyperspectral images (HSIs) has attracted much interest in recent few decades [1]. It consists of identifying the pure spectral signatures, called *endmembers*, and estimating their corresponding fractions, called *abundances*. Many research works have been dedicated to address either one task or both two. The first task is commonly known as endmember extraction. A number of endmember extraction algorithms have been proposed from statistical and geometrical aspects (see [2]–[6] and references therein). Many abundance estimation algorithms, including our proposed one, address the latter task under the assumption that the spectral signatures of the endmembers are available [7]–[13].

A modeling mixture process is a fundamental but difficult task in abundance estimation. It seeks a balance between model accuracy and tractability. The linear mixture model (LMM) has been adopted by diverse abundance estimation approaches. It assumes that each pixel's spectral signature is a linear combination of the spectral signatures of the endmembers, weighted by their corresponding abundances. Recall that the spectral signatures of the endmembers are assumed available and belong to a predefined set usually called *dictionary*. By natural, the abundance vector of a mixed pixel in an HSI should satisfy *abundance nonnegative constraint* (ANC) and *abundance sum-to-one constraint* (ASC) [14]. This leads to a constrained spectral unmixing problem.

Sparse constraint exploits another physically prior knowledge for spectral unmixing. It assumes that each mixed pixel in an HSI linearly consists of, according to LMM, only a few of spectral signatures compared with large-scale available dictionaries. In addition, suitable sparsity-inducing prior distributions are used for fractional abundances in Bayesian schemes [15]–[17]. The sparsity assumption usually incorporates with the classic ANC and ASC in the literature [7]. The sparse constrained spectral unmixing shows significant advantages, lying at the center of interest of unmixing algorithms. Since the sparsity is imposed on the abundance vector of each pixel, we shall call it as *single sparsity* to distinguish from other sparse structures mentioned later. In other words, the single sparse assumption only focuses on each single pixel, ignoring potential information, specifically spatial information, offered by other pixels.

Besides single sparse assumption, *spatial correlation* between each pixel and its neighbors has been exploited for better spectral unmixing results. Suppose that nearby pixels in the homogeneous regions of HSIs have a high degree of correlation among their spectral signatures. It probably leads to a high correlation among the corresponding abundance vectors. Hyperspectral unmixing with spatial correlation constraint becomes a powerful unmixing scheme, leading to many state-of-the-art algorithms (see [8], [10], [12], [18]–[21] and reference therein).

Specifically, assuming that the abundance matrix is piecewise smooth for the same endmember among adjacent pixels, Iordache *et al.* [8] include the total variation (TV) spatial regularization to the sparse unmixing scheme, providing a promising unmixing performance. The TV regularizer term is, however, an overstrict assumption that adjacent pixels should have both similar mixing endmembers and similar abundance

Manuscript received April 10, 2018; revised August 14, 2018; accepted September 21, 2018. Date of publication October 26, 2018; date of current version March 25, 2019. This work was supported in part by NSFC under Grant 61772003, Grant 61702083 and Grant 61876203, in part by the Science Strength Promotion Programme of UESTC, and in part by the Fundamental Research Funds for the Central Universities under Grant ZYGX2016J129, Grant ZYGX2016KYQD142, and Grant ZYGX2016J132. *(Corresponding authors: Jie Huang; Ting-Zhu Huang.)*

The authors are with the School of Mathematical Sciences, University of Electronic Science and Technology of China, Chengdu 611731, China (e-mail: happyjie07mo@163.com; tingzhuhuang@126.com).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TGRS.2018.2873326

fractions. A joint-sparsity (or row-sparsity) model provides a less strict assumption that neighboring pixels share the same support set of endmembers; that is, the pixels consist of similar endmembers, but they do not necessarily have similar abundances for the same endmember [10]. Nevertheless, the joint-sparsity model might cause some *aliasing artifacts* for the pixels on the boundaries of different constituent endmembers [21]. Many other joint-sparse models have been proposed for a hyperspectral unmixing problem. For example, nonlocal self-similarity regularization is incorporated into the classical joint-sparsity regression model to further exploit the spatial contextual information [22]. Also, a probabilistic joint-sparse regression model assumes an exponential prior distribution for fractional abundances and utilizes implicit relations of neighboring pixels [23]. Recently, pixels within general segments (superpixels) of HSIs are assumed to share the same sparse structure in abundances [24], [25], saying again that utilizing spatial information can lead to an improvement in unmixing performance.

On the other hand, a low-rank constraint of the abundance matrix has been increasingly adopted for sparse unmixing, providing a new perspective for spatial correlation [12], [21], [26]–[28], and as well as in other applications, such as compressive sensing [29] and tensor completion [30], [31]. The spatial correlation among pixels in an HSI translates into linear dependence among their corresponding abundance vectors. The resulting abundance matrix thus admits a low-rank property. Notice that the low-rank property does by no means invalidate sparsity; Giampouras *et al.* [12] simultaneously impose single sparsity and low rankness on abundance matrices, taking into account both sparsity and spatial correlation information in HSIs. Recently, a low-rank local abundance regularizer has collaborated with joint sparsity and TV spatial regularizer for unmixing problem [32]. The effectiveness of these unmixing algorithms is illustrated by extensive simulated and real-data experiments.

In this paper, we propose a *joint-sparsity-blocks* model for spectral unmixing. Namely, the abundance matrix of size $m \times n$ is partitioned to have one row block and s column blocks, and each column block itself is joint-sparse. The proposed sparse model promotes that pixels in one block share the same support set and is thus a special case of typical joint sparsity. On the other hand, the joint-sparsity-blocks assumption generalizes both the single and the joint sparsities by varying the value of s. That is, if $s = n$, then each joint-sparse column block reduces to a single sparse vector, whereas if $s = 1$, then all the pixels are in one block and we get exactly a typical joint-sparse abundance matrix.

For a hyperspectral unmixing problem, we propose to simultaneously impose the joint-sparsity-blocks structure and low rankness on abundance matrices for pixels in a sliding window. We then develop a new algorithm called *joint-sparse-blocks and low-rank unmixing* (JSpBLRU) under the classic *alternating direction method of multipliers* (ADMM) framework. Specifically, we develop a new two-level reweighting strategy for joint-sparse-blocks regression problem. That is, weighting coefficients are assigned differently to each row within each block at each iteration. The first simulated experiment in Section IV-C demonstrates the effectiveness of this strategy. Other simulated and real-data experiments in Section IV show the efficacy of the proposed algorithm.

The rest of this paper is structured as follows. Section II introduces a joint-sparse-blocks regression problem and presents some related properties. In Section III, we derive a JSpBLRU algorithm. The effectiveness of JSpBLRU is demonstrated by both simulated experiments in Section IV-A and a real-data experiment in Section IV-B. Section IV-C discusses the parameters selection of JSpBLRU. Finally, concluding remarks are given in Section V.

II. Joint-Sparse-Blocks Regression

Sparse representation has been well studied in many applications, such as compressive sensing [33], image processing [34]–[38], hyperspectral unmixing [7], [8], [10], and HSI super-resolution [39], to name a few. Suppose $\mathbf{X} \in \mathbb{R}^{m\times n}$, and the ℓ_0 quasi-norm, to describe the sparsity of $\mathbf{X}$, is defined as

$$\|\mathbf{X}\|_0 = \text{the number of nonzero elements in } \mathbf{X}.$$

The discrete, nonconvex, and non-Lipschitz character of ℓ_0 quasi-norm makes applications difficult. Its usual replacement in the literature is ℓ_1 norm defined as

$$\|\mathbf{X}\|_{1,1} = \sum_{i=1}^{m}\sum_{j=1}^{n}|x_{i,j}|$$

where $x_{i,j}$ denotes the (i, j)th element of $\mathbf{X}$. We note that the ℓ_1 norm is the best convex approximate of ℓ_0 norm. Another favorable sparse representation is *joint sparsity* by applying mixed $\ell_{2,1}$ norm

$$\|\mathbf{X}\|_{2,1} = \sum_{i=1}^{m}\|\mathbf{x}^{[i]}\|_2 \tag{1}$$

where $\mathbf{x}^{[i]}$ is the ith row of $\mathbf{X}$. Joint sparsity is an important extension of the single sparsity and has been well studied in, e.g., [10], [11], [21], [25], [34], and [40]–[45]. The $\ell_{2,1}$ regularization promotes structured sparse recovery with a small number of nonzero rows of $\mathbf{X}$. It reduces the degrees of freedom in the solution and the possible computational difficulty occurring in hyperspectral unmixing problems, compressive sensing problems, and so on. It is worth noting that the $\ell_{2,1}$ norm is a convex relaxation of a row-ℓ_0 quasi-norm [46]

$$\|\mathbf{X}\|_{\text{row},0} = \text{the number of nonzero rows in } \mathbf{X}. \tag{2}$$

Two typical variant $\ell_{2,1}$ regularizations have been well considered. One is the $\ell_{p,q}$ regularization in [47]–[49] measured by an $\ell_{p,q}$ norm,

$$\|\mathbf{X}\|_{p,q} = \left(\sum_{i=1}^{m}\|\mathbf{x}^{[i]}\|_p^q\right)^{\frac{1}{q}}, \quad 0 \le q \le 1 \le p.$$

The other utilizes weighted $\ell_{2,1}$ norm defined by

$$\|\mathbf{X}\|_{\mathbf{w},2,1} = \sum_{i=1}^{m} w_i\|\mathbf{x}^{[i]}\|_2$$

for $\mathbf{w} = [w_1, \ldots, w_m] \geq \mathbf{0}$ (see [41], [50]). The variants promote structured sparse solutions and improve the performance of sparsity recovery results.

Block sparsity is also an important extension of single sparsity [51]–[53]. It assumes that sparse signals have nonzero entries occurring in blocks. That is, a vector (respectively, a matrix) with M blocks is called block-sparse if only $k < M$ of its blocks have nonzero Euclidean (respectively, Frobenius) norm. Similarly, group sparsity assumes solutions having a natural grouping of its components and the components within a group are likely to be either all zeros or all nonzeros [54], [55]. The joint single and group sparsity, namely, concatenates the group sparsity and the single sparsity to yield sparse solutions at both the group and individual feature levels [56], [57]. These sparse models improve the performance of single sparsity in many applications. Recently, a block-row sparse regularizer [58] concatenates the joint sparsity via an $\ell_{2,1}$ norm and the block sparsity via a Frobenius norm. It avoids the adverse impact of not only the redundant views but also the noisy features for multiview image classification.

Here, we propose a joint-sparsity-blocks structure, that is, $\mathbf{X}$ is first partitioned to have one row block and several column blocks, and then, each column block is joint-sparse. To be more specific, we partition $\mathbf{X}$ as

$$\mathbf{X} = [\mathbf{X}_1, \cdots, \mathbf{X}_s] \tag{3}$$

where $\mathbf{X}_j \in \mathbb{R}^{m \times d_j}$, for $j = 1, \ldots, s$, $\sum_{j=1}^{s} d_j = n$, and block number s is a positive integer for $1 \leq s \leq n$. Then, define *row-ℓ_0-blocks quasi-norm* of $\mathbf{X}$ as

$$\|\mathbf{X}\|_{\text{row},0,\text{b}} := \sum_{j=1}^{s} \|\mathbf{X}_j\|_{\text{row},0} \tag{4}$$

a sum of the number of nonzero rows of each block of $\mathbf{X}$. It immediately follows two extreme cases of $\|\mathbf{X}\|_{\text{row},0,\text{b}}$ with respect to s.

1) If $s = 1$, correspondently, $d_1 = n$, then $\mathbf{X}$ itself is one block and $\|\mathbf{X}\|_{\text{row},0,\text{b}}$ becomes $\|\mathbf{X}\|_{\text{row},0}$.
2) If $s = n$, correspondently, $d_1 = \cdots = d_n = 1$, then each column of $\mathbf{X}$ is one block and $\|\mathbf{X}\|_{\text{row},0,\text{b}}$ becomes $\|\mathbf{X}\|_0$.

Hence, the proposed row-ℓ_0-blocks quasi-norm generalizes the ℓ_0 norm and row-ℓ_0 quasi-norm. The generalization is novel, as far as we know.

We now consider the joint-sparse-blocks regression problem via row-ℓ_0-blocks regularization

$$\min_{\mathbf{X}} \ \frac{1}{2}\|\mathbf{X} - \mathbf{Y}\|_F^2 + w \sum_{j=1}^{s} \|\mathbf{X}_j\|_{\text{row},0} \tag{5}$$

where $\mathbf{Y} \in \mathbb{R}^{m \times n}$ is given, $\mathbf{X}$ is partitioned as in (3), $w > 0$ is a regularization parameter, and s denotes the number of blocks. Fig. 1 graphically shows the effectiveness of the proposed row-ℓ_0-blocks regularizer. Since the problem (5) is nonconvex and non-Lipschitz, it is generally hard to solve directly. A common approach is to relax the row-ℓ_0 regularization to $\ell_{2,1}$ regularization, and consequently, row-ℓ_0-blocks regularization becomes, here we called, *$\ell_{2,1}$-blocks* regularization. To be more general, we adopt a weighted $\ell_{2,1}$ norm for each block. Thus, we attain the following weighted $\ell_{2,1}$-blocks minimization problem:

$$\min_{\mathbf{X}} \ \frac{1}{2}\|\mathbf{X} - \mathbf{Y}\|_F^2 + \sum_{j=1}^{s} \|\mathbf{X}_j\|_{\mathbf{w}_j,2,1} \tag{6}$$

where the weighted $\ell_{2,1}$ norm of $\mathbf{X}_j$ is defined as

$$\|\mathbf{X}_j\|_{\mathbf{w}_j,2,1} = \sum_{i=1}^{m} w_{i,j} \left\|\mathbf{X}_j^{[i]}\right\|_2 \tag{7}$$

$\mathbf{X}_j^{[i]}$ is the ith row of the jth block of $\mathbf{X}$, $\mathbf{w}_j = [w_{1,j}, \ldots, w_{m,j}]^T \in \mathbb{R}^m$ is a nonnegative weighting vector, for $i = 1, \ldots, m$, $j = 1, \ldots, s$, and ^{T} denotes the transposition. Clearly, different weights are assigned to different rows in different blocks.

To solve (6), we first similarly partition $\mathbf{Y} = [\mathbf{Y}_1, \ldots, \mathbf{Y}_s]$ as $\mathbf{X}$ and obtain the reformulated problem

$$\min_{\mathbf{X}_1,\ldots,\mathbf{X}_s} \ \sum_{j=1}^{s} \left(\frac{1}{2}\|\mathbf{X}_j - \mathbf{Y}_j\|_F^2 + \|\mathbf{X}_j\|_{\mathbf{w}_j,2,1}\right). \tag{8}$$

Since the above-mentioned object function is proper, strictly convex, and *separable*, we equivalently decouple the minimization problem (8) to s subproblems

$$\min_{\mathbf{X}_j} \ \frac{1}{2}\|\mathbf{X}_j - \mathbf{Y}_j\|_F^2 + \|\mathbf{X}_j\|_{\mathbf{w}_j,2,1} \tag{9}$$

for $j = 1, \ldots, s$. Clearly, from [34], each subproblem admits a unique block solution, and the ith row of the unique block solution $\hat{\mathbf{X}}_j$, i.e., $\hat{\mathbf{X}}_j^{[i]}$, of (9), can be written explicitly as

$$\hat{\mathbf{X}}_j^{[i]} = \textbf{vect-soft}_{w_{i,j}}\left(\mathbf{Y}_j^{[i]}\right)$$

for $i = 1, \ldots, m$, $j = 1, \ldots, s$, where $\textbf{vect-soft}_{\alpha}(\cdot)$ is a nonlinear operator defined by

$$\textbf{vect-soft}_{\alpha}(\mathbf{x}) = \mathbf{x}\frac{\max\{\|\mathbf{x}\|_2 - \alpha, 0\}}{\max\{\|\mathbf{x}\|_2 - \alpha, 0\} + \alpha} \tag{10}$$

for $\forall\ \mathbf{x} \in \mathbb{R}^N$ and $\alpha > 0$. Thus, we arrive at a useful result that extends the work about group separable ℓ_2 regularization problem in [34].

Theorem 1: For any matrices $\mathbf{X}, \mathbf{Y} \in \mathbb{R}^{m \times n}$. Partition $\mathbf{X} = [\mathbf{X}_1, \ldots, \mathbf{X}_s]$, and correspondingly, $\mathbf{Y} = [\mathbf{Y}_1, \ldots, \mathbf{Y}_s]$, where s is a positive integer for $1 \leq s \leq n$. Denote the ith row of the jth block of $\mathbf{Y}$ as $\mathbf{Y}_j^{[i]}$, for $i = 1, \ldots, m$ and$j = 1, \ldots, s$. Then, the minimization problem in (6) has a unique solution

$$\hat{\mathbf{X}} = [\hat{\mathbf{X}}_1, \ldots, \hat{\mathbf{X}}_s], \quad \hat{\mathbf{X}}_j^{[i]} = \textbf{vect-soft}_{w_{i,j}}\left(\mathbf{Y}_j^{[i]}\right)$$

for $i = 1, \ldots, m$ and $j = 1, \ldots, s$, where $\textbf{vect-soft}_{w_{i,j}}(\cdot)$ is a nonlinear operator defined in (10).

We now consider the solution $\hat{\mathbf{X}}$ of (6) when $s = n$. For this purpose, we first define that $\textbf{soft}_{\boldsymbol{\alpha}}(\cdot)$ is a nonlinear soft-thresholding operator defined componentwise by

$$(\textbf{soft}_{\boldsymbol{\alpha}}(\mathbf{x}))_i = x_i \frac{\max\ \{|x_i| - \alpha_i, 0\}}{\max\ \{|x_i| - \alpha_i, 0\} + \alpha_i} \tag{11}$$

for $\forall\ \mathbf{x} = [x_1, \ldots, x_N]^T \in \mathbb{R}^N$ and a weighting vector $\boldsymbol{\alpha} = [\alpha_1, \ldots, \alpha_N]^T \geq \mathbf{0}$. Naturally, the **vect-soft** operator is

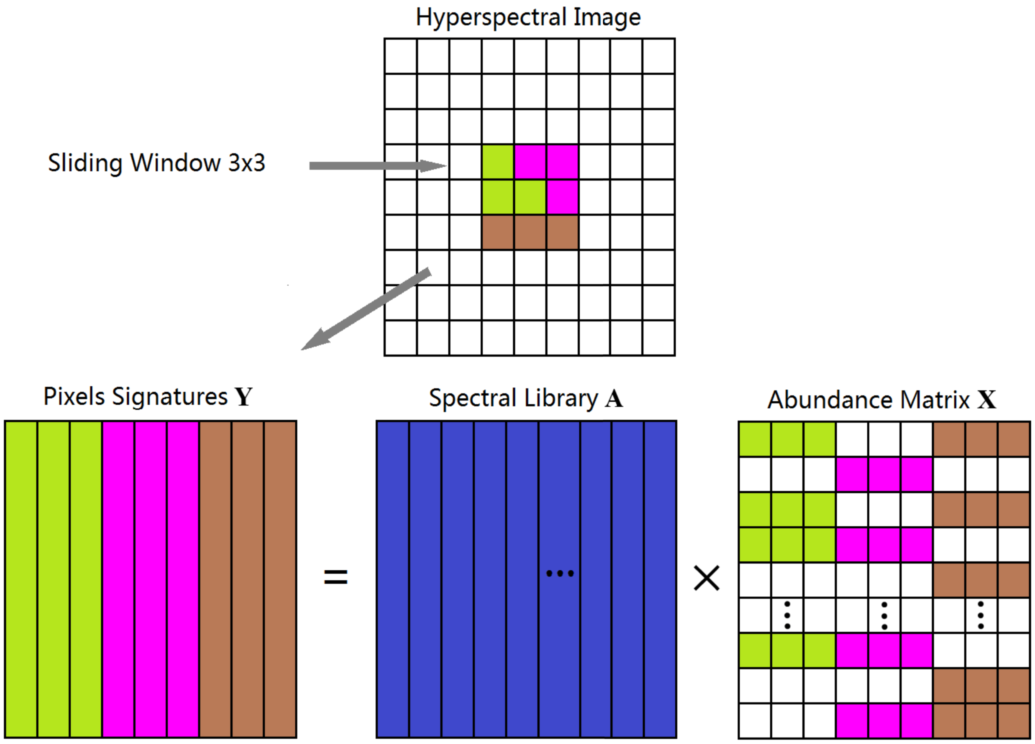


Fig. 1. Graphical illustration of the effect of the proposed row-ℓ_0-blocks quasi-norm regularizer. In this example, the abundance matrix is composed of three column blocks and *each column block is joint-sparse*. Within each block, non-active members of the considered spectral library $\mathbf{A}$ are represented in white color.

a vectorial **soft**. If $s = n$, then each column of $\hat{\mathbf{X}}$ is one block. Consequently, $\mathbf{Y}_j^{[i]}$ reduces to the (i, j)th element of $\mathbf{Y}$, for $i = 1, \ldots, m$ and $j = 1, \ldots, n$. If we handle $\mathbf{Y}$ columnwise, $\mathbf{vect\text{-}soft}_{w_{i,j}}(\mathbf{Y}_j^{[i]})$, $i = 1, \ldots, m$, compactly reformulates to the soft-threshold $\mathbf{soft}_{\mathbf{w}_j}(\mathbf{Y}_j)$. This says that each column of the unique solution $\hat{\mathbf{X}}$ of (6) can be obtained by

$$\hat{\mathbf{X}}_j = \mathbf{soft}_{\mathbf{w}_j}(\mathbf{Y}_j), \quad j = 1, \ldots, n.$$

III. JSpBLRU

In this section, we propose to simultaneously impose the joint-sparsity-blocks structure and low rankness on abundance estimation for HSI unmixing problem. We will first give the problem formulation and then the proposed unmixing algorithm, followed by a new reweighting strategy.

A. Problem Formulation

Let $\mathbf{Y} \in \mathbb{R}^{L \times n}$ be the observed data matrix, where L is the number of bands and n is the number of pixels. Let $\mathbf{A} = [\mathbf{a}_1, \ldots, \mathbf{a}_m] \in \mathbb{R}^{L \times m}$ be the dictionary, in which each column $\mathbf{a}_j = [a_{1,j}, \ldots, a_{L,j}]^T$ is the spectral signature of the jth endmember, for $j = 1, \ldots, m$. The mixing process, under the LMM, can be modeled as

$$\mathbf{Y} = \mathbf{A}\mathbf{X} + \mathbf{N}$$

where $\mathbf{X} \in \mathbb{R}^{m \times n}$ is the fractional abundance matrix, each column of which corresponds with the abundance fractions of the endmembers in a pixel, and $\mathbf{N} \in \mathbb{R}^{L \times n}$ is an independent and identically distributed (i.i.d.) zero-mean Gaussian noise matrix. According to physical background, two constraints are often imposed on $\mathbf{X}$

$$\mathbf{X} \geq \mathbf{0}, \quad \mathbf{1}^T\mathbf{X} = \mathbf{1}^T$$

namely, the ANC and the ASC, respectively. Here, the inequality $\mathbf{X} \geq \mathbf{0}$ is considered elementwise nonnegative and $\mathbf{1}$ is a column vector of 1s. Similarly, as in [7] and [12], we relax the sum-to-one constraint to focus on the exploitation of structural characters of $\mathbf{X}$.

Giampouras *et al.* [12] simultaneously impose single sparsity and low rankness on the abundance matrix for pixels lying in the homogeneous regions of HSIs. The resulting unmixing algorithms exploit both spatial correlation by the weighted nuclear norm and single sparse structure by the weighted ℓ_1 norm for pixels in a small, e.g., 3 × 3, sliding window. The unmixing results are promising. Instead of using single sparsity in [12], we utilize the joint-sparse-blocks representation to further exploit the spatial information in HSIs. At this point, we propose to simultaneously impose the *joint-sparsity-blocks structure* and *low rankness* on the abundance matrix for the pixels lying in a sliding window. Then, we obtain the following optimization problem for spectral unmixing:

$$\begin{aligned} \min_{\mathbf{X} \in \mathbb{R}^{m \times K}} \; & \frac{1}{2}\|\mathbf{Y} - \mathbf{A}\mathbf{X}\|_F^2 + \lambda\|\mathbf{X}\|_{\text{row},0,\text{b}} + \tau \ \text{rank}(\mathbf{X}) \\ \text{s.t.} \; & \mathbf{X} \geq \mathbf{0} \end{aligned} \tag{12}$$

where λ and τ are nonnegative regularization parameters and K denotes the number of pixels in a sliding window.

The above-mentioned optimization problem is nonconvex and NP-hard and, thus, difficult to solve.

Instead of directly solving the model in (12), we first give a convex relaxation by replacing the row-ℓ_0-blocks quasi-norm with the $\ell_{2,1}$-blocks norm and replacing the rank of $\mathbf{X}$ with the well-known nuclear norm: $\|\mathbf{X}\|_* = \sum_{i=1}^r \sigma_i(\mathbf{X})$, where $r = \text{rank}(\mathbf{X})$ and $\sigma_i(\mathbf{X})$ is the ith singular value of $\mathbf{X}$, for $i = 1, \ldots, r$. Recall from the definition in (4) and partition $\mathbf{X}$ as in (3), the surrogate convex optimization problem is

$$\min_{\mathbf{X}\in\mathbb{R}^{m\times K}} \frac{1}{2}\|\mathbf{Y}-\mathbf{A}\mathbf{X}\|_F^2 + \lambda\sum_{j=1}^{s}\|\mathbf{X}_j\|_{2,1} + \tau\|\mathbf{X}\|_* \quad \text{s.t. } \mathbf{X} \geq \mathbf{0}. \tag{13}$$

We remark that the above-mentioned model can reduce to three state-of-the-art unmixing models: the ADSpLRU model of [12] if $s = n$, the CLSUnSAL model of [10] if $s = 1$ and $\tau = 0$, and the SUnSAL model of [7] if $s = n$ and $\tau = 0$.

In an attempt to enhance sparsity along the rows in each block in (13), we update the $\ell_{2,1}$ norm of $\mathbf{X}_j$ to the *weighted* $\ell_{2,1}$ norm defined as in (7). In addition, we use the *weighted* nuclear norm defined by

$$\|\mathbf{X}\|_{\mathbf{b},*} = \sum_{i=1}^{r} b_i\sigma_i(\mathbf{X}) \tag{14}$$

where $\mathbf{b} = [b_1, \ldots, b_r]$ is a nonnegative weighting vector. The weighted nuclear norm treats the individual singular values differently and enhances the sparsity on the singular values [29], [59]–[61]. Thus, we attain the optimization model as

$$\min_{\mathbf{X}\in\mathbb{R}^{m\times K}} \frac{1}{2}\|\mathbf{Y}-\mathbf{A}\mathbf{X}\|_F^2 + \lambda\sum_{j=1}^{s}\|\mathbf{X}_j\|_{\mathbf{w}_j,2,1} + \tau\|\mathbf{X}\|_{\mathbf{b},*} \quad \text{s.t. } \mathbf{X} \geq \mathbf{0} \tag{15}$$

where $\mathbf{w}_j = [w_{1,j}, \ldots, w_{m,j}]^T$ is a nonnegative weighting vector, for $j = 1, \ldots, s$. Clearly, the weighted $\ell_{2,1}$-blocks norm is always convex for nonnegative $w_{i,j}$ values. Also, it is known that if nonnegative weights b_i, $i = 1, \ldots, r$, are descending, then the weighted nuclear norm is convex [29], [60]. Under these conditions, thus, the overall cost function in (15) is convex. For simplicity, we will solve the model (15) under the assumption of $b_i = b$, for $i = 1, \ldots, r$, in Section III-B. We will propose an adaptive selection of b_i and $w_{i,j}$ values by computing weights used for the next iteration from the obtained estimations in Section III-C.

B. Joint-Sparse-Blocks and Low-Rank Unmixing Algorithm

In this section, we solve the proposed model in (15) under the ADMM framework. The ADMM is a convex optimization method, a variant of the classic augmented Lagrangian method [62]. It has been widely used in a number of areas, such as machine learning and image processing (see [63], [64] and references therein).

To begin, we partition $\mathbf{X}$ with s blocks as in (3)

$$\mathbf{X} = [\mathbf{X}_1, \ldots, \mathbf{X}_s].$$

Introducing three variables $\mathbf{V}_1$, $\mathbf{V}_2$, and $\mathbf{V}_3$ and partitioning $\mathbf{V}_1 = [\mathbf{V}_{1,1}, \ldots, \mathbf{V}_{1,s}]$ as $\mathbf{X}$, we transform (15) to an equivalent model

$$\begin{aligned}\min_{\mathbf{X}} \ & \frac{1}{2}\|\mathbf{Y}-\mathbf{A}\mathbf{X}\|_F^2 + \lambda\sum_{j=1}^{s}\|\mathbf{V}_{1,j}\|_{\mathbf{w}_j,2,1} \\ & + \tau\|\mathbf{V}_2\|_{\mathbf{b},*} + \iota_{\mathbb{R}^+}(\mathbf{V}_3) \\ \text{s.t. } & \mathbf{X} = \mathbf{V}_1, \quad \mathbf{X} = \mathbf{V}_2, \ \mathbf{X} = \mathbf{V}_3\end{aligned} \tag{16}$$

where ι_Ω is the indicator function of a set $\boldsymbol{\Omega}$, i.e., $\iota_\Omega(x) = 0$ if $x \in \boldsymbol{\Omega}$ and $\iota_\Omega(x) = +\infty$ otherwise.

To make notations more concise, we let

$$\begin{aligned} g(\mathbf{X}, \mathbf{V}) = \frac{1}{2}\|\mathbf{Y}-\mathbf{A}\mathbf{X}\|_F^2 + \lambda\sum_{j=1}^{s}\|\mathbf{V}_{1,j}\|_{\mathbf{w}_j,2,1} \\ + \tau\|\mathbf{V}_2\|_{\mathbf{b},*} + \iota_{\mathbb{R}^+}(\mathbf{V}_3)\end{aligned} \tag{17}$$

and define

$$\mathbf{V} = \begin{pmatrix}\mathbf{V}_1\\ \mathbf{V}_2\\ \mathbf{V}_3\end{pmatrix}, \quad \mathbf{G} = \begin{pmatrix}\mathbf{I}\\ \mathbf{I}\\ \mathbf{I}\end{pmatrix} \in \mathbb{R}^{3m\times m}. \tag{18}$$

Then, we obtain a compact form of (16)

$$\min_{\mathbf{X},\mathbf{V}} \ g(\mathbf{X}, \mathbf{V}) \quad \text{s.t. } \mathbf{G}\mathbf{X} = \mathbf{V}. \tag{19}$$

Let

$$\mathcal{L}_\mu(\mathbf{X}, \mathbf{V}; \boldsymbol{\Lambda}) = g(\mathbf{X}, \mathbf{V}) + \frac{\mu}{2}\|\mathbf{G}\mathbf{X} - \mathbf{V} - \boldsymbol{\Lambda}\|_F^2 \tag{20}$$

where $\mu > 0$ is a penalty parameter and $\boldsymbol{\Lambda} = (\boldsymbol{\Lambda}_1^T, \boldsymbol{\Lambda}_2^T, \boldsymbol{\Lambda}_3^T)^T \in \mathbb{R}^{3m\times n}$. Then, the ADMM framework is derived

$$\begin{cases}\mathbf{X}^{k+1} = \text{argmin}_{\mathbf{X}} \ \mathcal{L}_\mu(\mathbf{X}, \mathbf{V}^k; \boldsymbol{\Lambda}^k)\\ \mathbf{V}^{k+1} = \text{argmin}_{\mathbf{V}} \ \mathcal{L}_\mu(\mathbf{X}^{k+1}, \mathbf{V}; \boldsymbol{\Lambda}^k)\\ \boldsymbol{\Lambda}^{k+1} = \boldsymbol{\Lambda}^k - (\mathbf{G}\mathbf{X}^{k+1} - \mathbf{V}^{k+1}).\end{cases} \tag{21}$$

We now show that each subproblem of (21) has a closed-form solution. To begin, the $\mathbf{X}$-subproblem, after dropping constant terms, is equivalent to solve

$$\begin{aligned}\mathbf{X}^{k+1} &= \underset{\mathbf{X}}{\text{argmin}}\, g(\mathbf{X}, \mathbf{V}^k) + \frac{\mu}{2}\|\mathbf{G}\mathbf{X} - \mathbf{V}^k - \boldsymbol{\Lambda}^k\|_F^2\\ &= \underset{\mathbf{X}}{\text{argmin}}\, \frac{1}{2}\|\mathbf{Y}-\mathbf{A}\mathbf{X}\|_F^2 + \frac{\mu}{2}\left\|\mathbf{X} - \mathbf{V}_1^k - \boldsymbol{\Lambda}_1^k\right\|_F^2\\ &\quad + \frac{\mu}{2}\left\|\mathbf{X} - \mathbf{V}_2^k - \boldsymbol{\Lambda}_2^k\right\|_F^2 + \frac{\mu}{2}\left\|\mathbf{X} - \mathbf{V}_3^k - \boldsymbol{\Lambda}_3^k\right\|_F^2.\end{aligned}$$

It is a least-squares problem. A simple calculation gives

$$\begin{aligned}&\mathbf{X}^{k+1}\\ &= (\mathbf{A}^T\mathbf{A}+3\mu\mathbf{I})^{-1}\left(\mathbf{A}^T\mathbf{Y}+\mu\left(\mathbf{V}_1^k+\boldsymbol{\Lambda}_1^k+\mathbf{V}_2^k+\boldsymbol{\Lambda}_2^k+\mathbf{V}_3^k+\boldsymbol{\Lambda}_3^k\right)\right).\end{aligned}$$

We decouple the $\mathbf{V}$-subproblem of (21) to three independent subparts with respect to $\mathbf{V}_1$, $\mathbf{V}_2$, and $\mathbf{V}_3$, and each subproblem has a closed-form solution. For $\mathbf{V}_1$-subproblem, after dropping constant terms, we obtain

$$\mathbf{V}_1^{k+1} = \underset{\mathbf{V}_1}{\text{argmin}} \ \lambda\sum_{j=1}^{s}\|\mathbf{V}_{1,j}\|_{\mathbf{w}_j,2,1} + \frac{\mu}{2}\left\|\mathbf{X}^{k+1} - \mathbf{V}_1 - \boldsymbol{\Lambda}_1^k\right\|_F^2. \tag{22}$$

Recall $\mathbf{X} = [\mathbf{X}_1, \ldots, \mathbf{X}_s]$ and $\mathbf{V}_1 = [\mathbf{V}_{1,1}, \ldots, \mathbf{V}_{1,s}]$, and we correspondently partition $\mathbf{\Lambda}_1 = [\mathbf{\Lambda}_{1,1}, \ldots, \mathbf{\Lambda}_{1,s}]$. Then, from Theorem 1, we obtain the ith row of the jth block of $\mathbf{V}_1^{k+1}$

$$\left(\mathbf{V}_{1,j}^{k+1}\right)^{[i]} = \textbf{vect-soft}_{\frac{\lambda}{\mu} w_{i,j}}\left(\left(\mathbf{X}_j^{k+1} - \mathbf{\Lambda}_{1,j}^{k}\right)^{[i]}\right) \tag{23}$$

for $i = 1, \ldots, m$ and $j = 1, \ldots, s$.

Before solving the $\mathbf{V}_2$-subproblem, we give some definitions. Recall $r = \operatorname{rank}(\mathbf{X})$ and let σ_i denote the ith singular value of $\mathbf{X}$, $i = 1, \ldots, r$. Denote elementwise $(\cdot)_+ = \max(\cdot, 0)$ and $\mathbf{X} = \tilde{\mathbf{U}}\mathbf{Diag}(\sigma_1, \ldots, \sigma_r)\tilde{\mathbf{V}}^T$ is the singular value decomposition (SVD) of $\mathbf{X}$. Define the singular value thresholding operator $\mathbf{SVT}_{\mathbf{b},\beta}(\cdot)$ on $\mathbf{X}$ as

$$\mathbf{SVT}_{\mathbf{b},\beta}(\mathbf{X}) = \tilde{\mathbf{U}}\mathbf{Diag}((\sigma_1 - \beta b_1)_+, \ldots, (\sigma_r - \beta b_r)_+)\tilde{\mathbf{V}}^T.$$

The closed-form solution to the $\mathbf{V}_2$-subproblem is

$$\begin{aligned}\mathbf{V}_2^{k+1} &= \underset{\mathbf{V}_2}{\operatorname{argmin}}\, \tau\|\mathbf{V}_2\|_{\mathbf{b},*} + \frac{\mu}{2}\left\|\mathbf{X}^{k+1} - \mathbf{V}_2 - \mathbf{\Lambda}_2^k\right\|_F^2 \\ &= \mathbf{SVT}_{\mathbf{b},\frac{\tau}{\mu}}\left(\mathbf{X}^{k+1} - \mathbf{\Lambda}_2^k\right).\end{aligned} \tag{24}$$

For $\mathbf{V}_3$-subproblem, we have

$$\mathbf{V}_3^{k+1} = \underset{\mathbf{V}_3}{\operatorname{argmin}}\ \imath_{\mathbb{R}^+}(\mathbf{V}_3) + \frac{\mu}{2}\left\|\mathbf{X}^{k+1} - \mathbf{V}_3 - \mathbf{\Lambda}_3^k\right\|_F^2.$$

It is easy to obtain that

$$\mathbf{V}_3^{k+1} = \max\ \left(\mathbf{X}^{k+1} - \mathbf{\Lambda}_3^k, \mathbf{0}\right).$$

Finally, we update the multipliers

$$\begin{cases}\mathbf{\Lambda}_1^{k+1} = \mathbf{\Lambda}_1^k - \left(\mathbf{X}^{k+1} - \mathbf{V}_1^{k+1}\right) \\ \mathbf{\Lambda}_2^{k+1} = \mathbf{\Lambda}_2^k - \left(\mathbf{X}^{k+1} - \mathbf{V}_2^{k+1}\right) \\ \mathbf{\Lambda}_3^{k+1} = \mathbf{\Lambda}_3^k - \left(\mathbf{X}^{k+1} - \mathbf{V}_3^{k+1}\right).\end{cases}$$

To make it more clear, we summarize the proposed JSpBLRU algorithm in the following.

Algorithm 1 Pseudocode of the JSpBLRU Algorithm

```
**Input:** $\mathbf{Y}$, $\mathbf{A}$.
**Selected parameters:** $\lambda$, $\tau$, $\mu$, partition strategy of $\mathbf{X}$,
  $\mathbf{b}$, $\mathbf{w}_j$ $(j = 1, \ldots, s)$, maximum iterations.
**Initialization:** $\mathbf{\Lambda}_l^0$, $\mathbf{V}_l^0$, $l = 1, 2, 3$, and set $k = 0$.
**Repeat:**
  $\mathbf{X}^{k+1} = (\mathbf{A}^T\mathbf{A} + 3\mu\mathbf{I})^{-1}\Big(\mathbf{A}^T\mathbf{Y} + \mu(\mathbf{V}_1^k + \mathbf{\Lambda}_1^k + \mathbf{V}_2^k + \mathbf{\Lambda}_2^k + \mathbf{V}_3^k + \mathbf{\Lambda}_3^k)\Big)$.
  Partition
      $\mathbf{X}^{k+1} = [\mathbf{X}_1^{k+1}, \ldots, \mathbf{X}_s^{k+1}]$,
      $\mathbf{\Lambda}_1^k = [\mathbf{\Lambda}_{1,1}^k, \ldots, \mathbf{\Lambda}_{1,s}^k]$.
    Compute the $i$th row of the $j$th block of $\mathbf{V}_1^{k+1}$ by
     $(\mathbf{V}_{1,j}^{k+1})^{[i]} = \textbf{vect-soft}_{\frac{\lambda}{\mu}w_{i,j}}((\mathbf{X}_j^{k+1} - \mathbf{\Lambda}_{1,j}^k)^{[i]})$,
     for $i = 1, \ldots, m$, $j = 1, \ldots, s$.
  $\mathbf{V}_2^{k+1} = \mathbf{SVT}_{\mathbf{b},\frac{\tau}{\mu}}(\mathbf{X}^{k+1} - \mathbf{\Lambda}_2^k)$.
  $\mathbf{V}_3^{k+1} = \max(\mathbf{X}^{k+1} - \mathbf{\Lambda}_3^k, \mathbf{0})$.
  $\mathbf{\Lambda}_l^{k+1} = \mathbf{\Lambda}_l^k - (\mathbf{X}^{k+1} - \mathbf{V}_l^{k+1})$, $l = 1, 2, 3$.
**until** some stopping criterion is satisfied.
**Output:** $\hat{\mathbf{X}} = \mathbf{X}^{k+1}$.
```

Clearly, the most expensive step in JSpBLRU is to perform an **SVT** operator, specifically, the SVD, with complexity $\mathcal{O}(m^2K)$ at each iteration. Here, we recall that K is the number of pixels in one sliding window and m is the number of endmembers. Computing $\mathbf{X}^{k+1}$ also requires complexity $\mathcal{O}(m^2K)$, whereas computing the others requires $\mathcal{O}(mK)$. Hence, the overall computational complexity of JSpBLRU is $\mathcal{O}(m^2K)$ per iteration. Concerning the convergence, recall that the cost function g in (19) is closed, proper, and convex if $b_i = b$, for $i = 1, \ldots, r$. The matrix $\mathbf{G}$ in (19) clearly has full column rank. Under these conditions, [65, Th. 1] ensures that if (19) has a solution, JSpBLRU converges, for any $\mu > 0$.

C. Weighting Coefficients Selection

To enhance the sparsity along the rows in each block and the sparsity on the singular values, we now consider the selection of weighting coefficients of $w_{i,j}$ values in (23) for weighted $\ell_{2,1}$-blocks norm and b_is in (24) for weighted nuclear norm.

For the weighted $\ell_{2,1}$-blocks norm regularization, first, we set $w_{i,j}$ for the next iteration computed from the argument of the **vect-soft** operator in (23). Specifically, we propose to use

$$w_{i,j}^{k+1} = \frac{1}{\left\|\left(\mathbf{X}_j^{k+1} - \mathbf{\Lambda}_{1,j}^k\right)^{[i]}\right\|_2 + \epsilon} \tag{25}$$

where $\epsilon = 10^{-16}$ is a small constant added to avoid singularities. Clearly, it is a *two-level reweighting strategy*. On one hand, different blocks are treated differently, leading to different joint-sparsity patterns among blocks. It is expected to enhance the structured sparsity for pixels on the boundaries between different regions in one sliding window. Within each block, on the other hand, different endmembers are treated differently, which promotes sparsity along the rows. Noticeably, the proposed weights in (25) is an extension of weights for reweighted ℓ_1 minimization. To be specific, if $s = n$, then weighted $\ell_{2,1}$-blocks norm reduces to weighted ℓ_1 norm, and consequently, $w_{i,j}^{k+1}$ becomes the reweighting coefficient for weighted ℓ_1 minimization in [12] and [66]. Our numerical test in Section IV-C will show that JSpBLRU with the reweighting coefficients (25) provides significant advantages over one without weights, i.e., $w_{i,j}^{k+1} = 1$. To the best of our knowledge, the reweighting strategy (25) is new for the mixed $\ell_{2,1}$ norm.

Similarly, we select the weights b_i^{k+1} for weighted nuclear norm based on the singular values of the argument in (24). That is,

$$b_i^{k+1} = \frac{1}{\tilde{\sigma}_i^{k+1} + \epsilon} \tag{26}$$

where $\tilde{\sigma}_i^{k+1}$ is the ith singular value of the argument $\mathbf{X}^{k+1} - \mathbf{\Lambda}_2^k$, $i = 1, \ldots, \operatorname{rank}(\mathbf{X}^{k+1} - \mathbf{\Lambda}_2^k)$. The reweighting strategy is widely used for many practical problems (see [12], [29], [61]).

It should be mentioned that the above-mentioned reweights in (25) and (26) render the minimization problem in (15) nonconvex. Though theoretical convergence analysis is hard to estimate, a series of research works has numerically shown the remarkable performance of the reweighting ℓ_1 in [12] and

[66] as well as reweighted nuclear norm in [29], [48], [60], and [61]. In this spirit, we adopt the reweighting strategies for JSpBLRU to promote the sparsity not only on the singular values but also on the lines within each block.

IV. Experiments

In this section, we demonstrate the proposed JSpBLRU algorithm on both simulated and real data. For simulated test problems, three spectral libraries are considered.

1) $\mathbf{A}_1 \in \mathbb{R}^{224\times 498}$: The Chapter 1 of the U.S. Geological Survey (USGS) spectral library (splib06a).[1] It comprises 498 spectral signatures with reflectance values measured in 224 spectral bands, distributed uniformly ranging from 0.4 to 2.5 μm.
2) $\mathbf{A}_2 \in \mathbb{R}^{224\times 240}$: A randomly selected subset of $\mathbf{A}_1$.
3) $\mathbf{A}_3 \in \mathbb{R}^{100\times 120}$: A subset of a library of 262 spectral signatures with 100 spectral bands generally found on satellites from the National Aeronautics and Space Administration Johnson Space Center Spacecraft Materials Spectral Database.

We will compare JSpBLRU with four state-of-the-art algorithms: SUnSAL [7], CLSUnSAL [10], SUnSAL-TV[2] [8], and ADSpLRU[3] [12]. Notice that SUnSAL-TV has been widely used as a nonsliding window approach (see [13], [67] and reference therein); hence, we do not give the comparison for Examples 1–4 in one sliding window. Our tests were done by using MATLAB R2016a on a MacBook Pro laptop with 2.3 GHz Intel Core i7 and 8 GB memory. The floating-point precision is 10^{-16}.

For all five algorithms, regularization parameters are tuned to their best performance with respect to root-mean-square error (RMSE) defined by

$$\text{RMSE} = \sqrt{\frac{1}{mn}\sum_{i=1}^{n}\|\hat{\mathbf{x}}_i - \mathbf{x}_i\|_2^2}$$

where n is the number of pixels, m is the number of endmembers, and $\hat{\mathbf{x}}_i$ and $\mathbf{x}_i$ are estimated and exact abundance vectors of the ith pixel, respectively. Generally speaking, the smaller the RMSE, the higher quality of the unmixing results. Another metric evaluating the performance of unmixing results is the signal-to-reconstruction error (SRE)

$$\text{SRE (dB)} = 10\log_{10}\left(\frac{\frac{1}{n}\sum_{i=1}^{n}\|\hat{\mathbf{x}}_i\|_2^2}{\frac{1}{n}\sum_{i=1}^{n}\|\hat{\mathbf{x}}_i - \mathbf{x}_i\|_2^2}\right).$$

Specifically, we select optimal regularization parameters in all compared algorithms: SUnSAL, CLSUnSAL, SUnSAL-TV, ADSpLRU, and JSpBLRU from the following sequence:

$$\{0,\ 10^{-6},\ 10^{-5},\ 0.0001,\ 0.0005,\ 0.001,\ 0.005,\ 0.01,\ 0.05,\ 0.1,\ 0.5,\ 1,\ 5\} \quad (27)$$

for Examples 1–4 and from a slightly wider range

$$\{0,\ 10^{-6},\ 10^{-5},\ 0.0001,\ 0.0005,\ 0.001,\ 0.005,\ 0.01,\ 0.05,\ 0.1,\ 0.5,\ 1,\ 5,\ 10,\ 50,\ 100\} \quad (28)$$

for Examples 5 and 6. Notice that each of SUnSAL and CLSUnSAL only has one regularization parameter, whereas each of SUnSAL-TV, ADSpLRU, and JSpBLRU has two. Thus, all possible combinations of these parameters from (27) or (28) were considered for SUnSAL-TV, ADSpLRU, and JSpBLRU. We also note that choosing optimal regularization parameters of SUnSAL-TV, ADSpLRU, and JSpBLRU costs much more computational time than that of SUnSAL and CLSUnSAL. In addition, for JSpBLRU, we empirically and initially set the augmented Lagrangian penalty parameter $\mu = 0.01$ for all simulated test problems and $\mu = 0.1$ for a real-data test problem. For ADSpLRU, we initially set $\mu = 0.01$ for all test problems in a sliding window, i.e., Examples 1–4, as in [12], and $\mu = 0.1$ for nonsliding-window experiments, i.e., Examples 5 and 6 and the real-data experiment (after empirically optimization). For SUnSAL-TV, we choose optimal penalty parameter μ from $10^{-3}, 10^{-2}, 10^{-1}$, and 1 to get the best RMSE values.

To make JSpBLRU easy to implement, we now propose a partition strategy of $\mathbf{X}$ in (3). Recall that n denotes the total number of pixels. Assume that d is a positive integer for $1 \le d \le n$. If n can be exactly divided by d, then we set the number of blocks $s = n/d$ and each block $\mathbf{X}_j$ in (3) has d columns, i.e.,

$$d = d_1 = d_2 = \cdots = d_s. \quad (29)$$

Otherwise, let $s = \lfloor n/d \rfloor$ be the largest integer no greater than n/d, and each of the first $s-1$ blocks contains d columns and the last block contains the remains, that is,

$$d = d_1 = d_2 = \cdots = d_{s-1}, \quad d_s = n - (s-1)\times d. \quad (30)$$

For both cases, it is easy to check that $n = \sum_{j=1}^{s} d_j$. Particularly, we empirically set block size $d = 3$ for all test problems. Other choices of d are discussed in Section IV-C.

For JSpBLRU, we use a similar adaptive strategy based on the primal and the dual ADMM variables as in [10]–[12]. Define the primal residual $\mathbf{r}^k$ and the dual residuals $\mathbf{d}^k$ at the kth JSpBLRU iteration as

$$\begin{aligned}\mathbf{r}^k &= \mathbf{G}\mathbf{X}^k - \mathbf{V}^k\\ \mathbf{d}^k &= \mu\mathbf{G}^T(\mathbf{V}^k - \mathbf{V}^{k-1}).\end{aligned}$$

We stop the JSpBLRU iteration if both of the termination criteria

$$\|\mathbf{r}^k\|_F \le \zeta, \quad \|\mathbf{d}^k\|_F \le \zeta,$$

are satisfied or when the number of iterations has reached 2000. In our tests, we set $\zeta = ((3m+L)K)^{(1/2)}\zeta^{\text{rel}}$ as in [11] and [12], where K is the number of pixels in the sliding window, and the relative tolerance $\zeta^{\text{rel}} > 0$ is empirically set to 5×10^{-6}.

In the following, we first test JSpBLRU on six simulated examples in Section IV-A. The first four examples are in a sliding window and the last two are nonsliding-window

[1] Available online: http://speclab.cr.usgs.gov/spectral.lib06.

[2] The MATLAB codes of SUnSAL, CLSUnSAL, and SUnSAL-TV are available at http://www.lx.it.pt/ bioucas/publications.html.

[3] Available online: http://members.noa.gr/parisg/demo_splr_unmixing.zip.

TABLE I

Performance of JSpBLRU With Different Parameter Values for Example 1

JSpBLRU	RMSE	SRE (dB)
$\lambda = 0$	0.0465	7.21
$\tau = 0$	0.0606	6.31
$d = 1$	0.0400	9.13
$d = 3$	0.0294	12.31
$d = K$	0.0437	8.07

TABLE II

Performance of Different Unmixing Algorithms With Different Low Rankness and Sparsity Levels for Example 2

(rank, sparsity)	Criteria	SUnSAL	CLSUnSAL	ADSpLRU	JSpBLRU
(1, 100%)	RMSE	0.3605	0.2957	0.1532	0.1527
	SRE (dB)	1.54	2.33	5.95	5.97
	Time (s)	0.05	0.09	0.62	0.20
(2, 10%)	RMSE	0.0267	0.0260	0.0178	0.0116
	SRE (dB)	11.24	11.44	15.83	21.37
	Time (s)	0.05	0.07	0.95	0.49
(2, 20%)	RMSE	0.0733	0.0691	0.0593	0.0521
	SRE (dB)	5.27	5.56	7.11	8.88
	Time (s)	0.04	0.06	0.89	0.38
(3, 10%)	RMSE	0.0317	0.0314	0.0251	0.0193
	SRE (dB)	9.44	9.44	11.94	15.56
	Time (s)	0.04	0.07	0.94	0.44
(3, 20%)	RMSE	0.0743	0.0733	0.0637	0.0585
	SRE (dB)	5.15	5.12	6.47	7.70
	Time (s)	0.04	0.06	0.93	0.36
(9, 5%)	RMSE	0.0075	0.0082	0.0060	0.0041
	SRE (dB)	19.07	18.15	21.71	25.35
	Time (s)	0.07	0.09	0.96	1.24

experiments. Specifically, we give a toy problem in Example 1. Examples 2 and 3 demonstrate the performance of JSpBLRU under different conditions. Example 4 considers two cases with more general distribution of the abundances. Two widely used synthetic HSIs are used to test different unmixing algorithms in Examples 5 and 6, respectively. Section IV-B applies JSpBLRU on a real HSI. Finally, we discuss the parameters selection of JSpBLRU in Section IV-C.

A. Experiments on Simulated Data

Example 1 (Toy Problem): Our purpose, in this test, is to demonstrate the performance of the combination of the joint-sparsity-blocks constraint and the low-rank constraint in JSpBLRU. We first consider JSpBLRU with a single constraint. To this end, on one hand, we consider only the low rankness by setting $\lambda = 0$ in (15). On the other hand, only the joint-sparsity-blocks constraint is considered by setting $\tau = 0$. It should be mentioned that when one of the regularization parameters λ and τ is fixed, the other is fine-tuned for a minimum RMSE. In addition, we consider three different values of d: $d = 1$, 3, and K in JSpBLRU with fine-tuned λ and τ.

For the test data, we generate an $m \times K$, with $m = 50$ and $K = 9$, abundance matrix with rank 3. The sparsity levels for three blocks are set to be 20% (i.e., 20% elements are nonzero), 15%, and 10%, respectively. The true abundance matrix is displayed in Fig. 2(a). We randomly select $m = 50$ endmembers from $\mathbf{A}_1$ to construct the spectral dictionary. The observed matrix $\mathbf{Y}$ is generated by the LMM and corrupted by Gaussian noise with an SNR of 30 dB.

Table I lists the RMSE and SRE (dB) values of JSpBLRU with different parameter values. From Table I, we see that JSpBLRU with only low rankness constraint gives lower RMSE and higher SRE (dB) values than one with only joint-sparsity-blocks constraint. It shows again that besides the sparsity constraint, assuming low rankness on the abundance matrix is effective for spectral unmixing. Moreover, concatenating low rankness and sparsity improves the abundance estimation performance. Particularly, JSpBLRU with $d = 3$ is better than with $d = 1$ and K. Estimated abundances by JSpBLRU with different parameter values are shown in Fig. 2(b)–(f). Clearly, visual comparison from Fig. 2 is consistent with the quantitative observation from Table I. From the results, we can see that imposing simultaneously joint-sparsity-blocks structure and low rankness in JSpBLRU gives more accurate and structured sparse estimations.

Example 2 (Different Low Rankness and Sparsity Levels): This example shows the effectiveness of JSpBLRU for abundance estimation with different low rankness and sparsity levels. We have done six experiments, each of which is performed independently 50 times. We set the number of endmembers $m = 50$ and the sliding window size 3×3, and, correspondingly, the number of pixels in the window $K = 9$. For the first experiment, we set rank 1 and sparsity level 100% so that the abundance is only low-rank. For the last experiment, we set rank 9 and sparsity level 5%, so the abundance matrix is sparse. In the remaining four experiments, we consider the abundance matrices are simultaneously low-rank and sparse, in which we set rank 2 or 3 with sparsity level 10% or 20%, respectively. For each experiment, we randomly select 50 endmembers from $\mathbf{A}_1$ to construct the spectral dictionary. The linearly mixed data are corrupted by the Gaussian noise of SNR = 30 dB.

Table II lists the RMSE and SRE (dB) values and the elapsed CPU time in seconds [denoted by Time (s)] of SUnSAL, CLSUnSAL, ADSpLRU, and JSpBLRU. We can see from Table II that SUnSAL and CLSUnSAL are much faster than ADSpLRU and JSpBLRU for all tests. It is consistent with the previous theoretical analysis that both the ADSpLRU and JSpBLRU algorithms perform an SVD per iteration. In addition, JSpBLRU is always faster than ADSpLRU except for the rank 9 and sparsity level 5% case.

We can also see from Table II that for the rank 1 and sparsity level 100% case, that is, the nonsparse fractional abundances, ADSpLRU and JSpBLRU provide better results than SUnSAL and CLSUnSAL. Particularly, the RMSE and SRE (dB) values of ADSpLRU and JSpBLRU are close. It can be expected that since ADSpLRU and JSpBLRU differ in the sparse representation of abundances, the two algorithms probably give similar unmixing results for nonsparse abundances. For simultaneous low rankness and sparsity, i.e., rank 2 or 3, sparsity level 10% or 20%, we can see that CLSUnSAL provides better or competitive RMSEs and SREs compared with SUnSAL. Clearly, ADSpLRU and JSpBLRU provide better results than SUnSAL and CLSUnSAL. Moreover, JSpBLRU provides the best RMSE and SRE (dB) values and it is

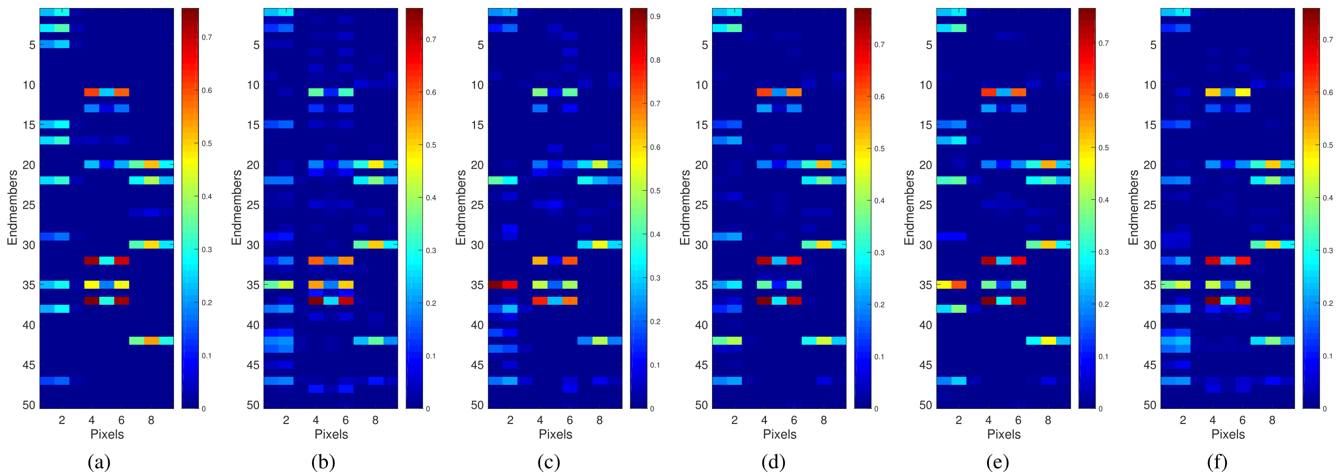


Fig. 2. Abundance estimation by JSpBLRU with different parameter values for Example 1. (a) Ground truth. (b) JSpBLRU with $\lambda = 0$. (c) JSpBLRU with $\tau = 0$. (d) JSpBLRU with $d = 1$. (e) JSpBLRU with $d = 3$. (f) JSpBLRU with $d = K$.

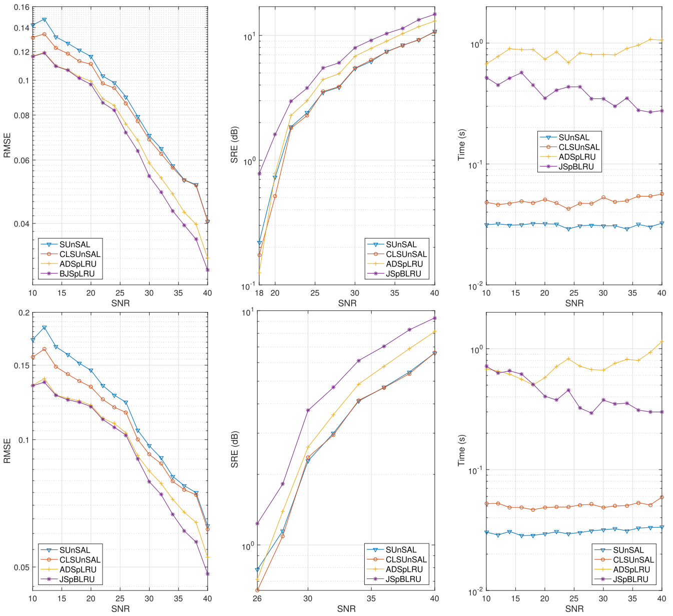


Fig. 3. Average RMSE, SRE (dB), and Time (s) after 50 runs against SNR with (Top row) Gaussian noise and (Bottom row) colored noise by different unmixing algorithms for Example 3.

TABLE III

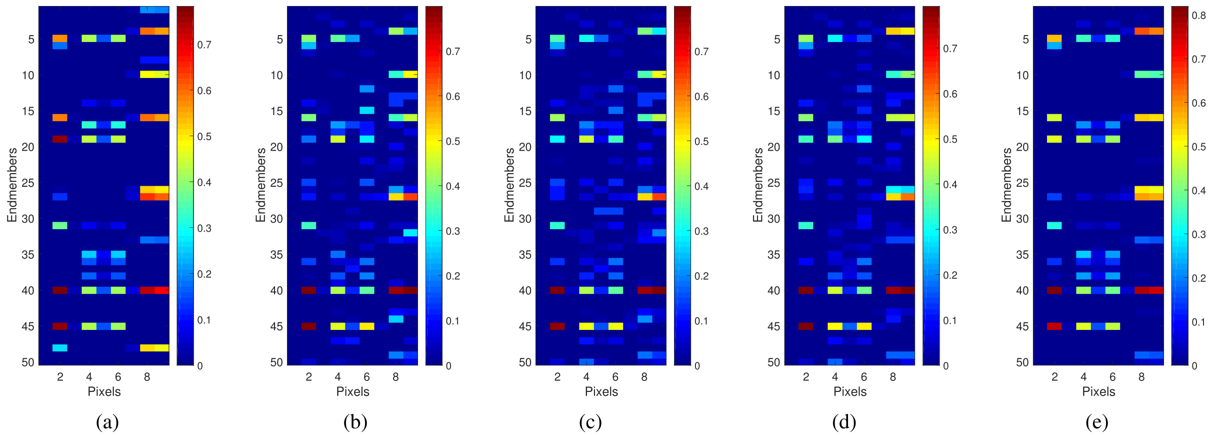
AVERAGE RMSE, SRE (dB), AND TIME (s) AFTER 50 RUNS BY DIFFERENT UNMIXING ALGORITHMS WHEN THE SIMULATED OBSERVATIONS ARE CORRUPTED BY GAUSSIAN OR COLORED NOISE FOR EXAMPLE 3

		Gaussian noise			
SNR	Criteria	SUnSAL	CLSUnSAL	ADSpLRU	JSpBLRU
20 dB	RMSE	0.1163	0.1107	0.0995	0.0973
	SRE (dB)	0.73	0.51	0.78	1.60
	Time (s)	0.03	0.05	0.74	0.35
30 dB	RMSE	0.0701	0.0683	0.0588	0.0541
	SRE (dB)	5.42	5.46	6.81	7.98
	Time (s)	0.03	0.05	0.80	0.35
40 dB	RMSE	0.0406	0.0406	0.0320	0.0296
	SRE (dB)	10.75	10.67	13.02	14.77
	Time (s)	0.03	0.06	1.06	0.28
		Colored noise			
SNR	Criteria	SUnSAL	CLSUnSAL	ADSpLRU	JSpBLRU
20 dB	RMSE	0.1458	0.1332	0.1206	0.1198
	SRE (dB)	-1.28	-1.50	-2.50	-2.35
	Time (s)	0.03	0.05	0.58	0.40
30 dB	RMSE	0.0967	0.0923	0.0843	0.0795
	SRE (dB)	2.28	2.36	2.61	3.75
	Time (s)	0.03	0.05	0.66	0.38
40 dB	RMSE	0.0625	0.0614	0.0527	0.0482
	SRE (dB)	6.59	6.62	8.13	9.32
	Time (s)	0.03	0.06	1.15	0.30

particularly better for the sparsity level 10% cases. This observation is also clear for sparse-only abundance estimation. Therefore, we conclude that the proposed JSpBLRU algorithm outperforms its rivals not only in either sparse-only or low-rank-only abundances but also in both low rank and sparse.

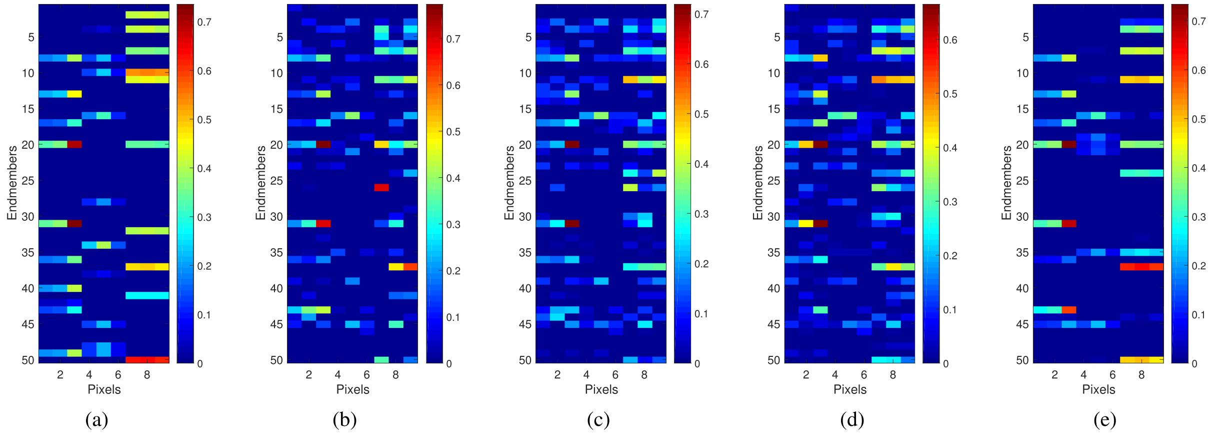
Example 3 (Different Noise Levels): In this experiment, we exploit the abundance estimation performance of different unmixing algorithms when observations are corrupted by Gaussian and correlated noise. True abundance matrices are of rank 3 and sparsity level 20%, the number of pixels $K = 9$, and the number of endmembers $m = 50$. The spectral dictionary is generated by randomly selecting 50 endmembers from $\mathbf{A}_1$. The observation data are generated by the LMM and then corrupted by the Gaussian noise (on the one hand) and also with spectrally correlated noise (on the other hand) resulting from low-pass i.i.d. Gaussian noise, with a normalized cutoff frequency of 20. The noise levels of SNR are ranging from 10 to 40 dB. Each test has been performed 50 times and average RMSE, SRE (dB), and Time (s) are recorded.

Fig. 3 shows the values of RMSE, SRE (dB), and Time (s) against the SNR of Gaussian noise and colored noise by different unmixing algorithms. We can see from the figure that ADSpLRU and JSpBLRU provide better SREs and RMSEs than SUnSAL and CLSUnSAL for all examined SNRs. In addition, JSpBLRU obtains comparable RMSEs for SNR $\leq$ 20 dB and lower RMSEs for SNR $>$ 20 dB compared with ADSpLRU. Clearly, the SRE (dB) values of JSpBLRU are higher than those of ADSpLRU for all examined SNRs. It should be noted that when SNR $\leq$ 20 dB, all four algorithms show a poor performance. We also see that SUnSAL and CLSUnSAL are



(a) (b) (c) (d) (e)

Fig. 4. Estimated abundance matrices of different unmixing algorithms for observation data corrupted by the Gaussian noise of SNR = 30 dB for Example 3. (a) True abundance matrix of rank 3 and sparsity level 20%. (b) SUnSAL with $\lambda_{\rm opt} = 0.005$, SRE (dB) = 4.13, RMSE = 0.0763, and Time (s) = 0.05. (c) CLSUnSAL with $\lambda_{\rm opt} = 0.005$, SRE (dB) = 4.38, RMSE = 0.0736, and Time (s) = 0.08. (d) ADSpLRU with $\lambda_{\rm opt} = 0.0005$ and $\tau_{\rm opt} = 0.01$, SRE (dB) = 5.76, RMSE = 0.0633, and Time (s) = 1.13. (e) JSpBLRU with $\lambda_{\rm opt} = 0.001$ and $\tau_{\rm opt} = 0.01$, SRE (dB) = 9.23, RMSE = 0.0466, and Time (s) = 0.31.



(a) (b) (c) (d) (e)

Fig. 5. Estimated abundance matrices of different unmixing algorithms for observation data corrupted by the colored noise of SNR = 30 dB for Example 3. (a) True abundance matrix of rank 3 and sparsity level 20%. (b) SUnSAL with $\lambda_{\rm opt} = 0.1$, SRE (dB) = 0.36, RMSE = 0.1133, and Time (s) = 0.02. (c) CLSUnSAL with $\lambda_{\rm opt} = 0.05$, SRE (dB) = −0.18, RMSE = 0.1125, and Time (s) = 0.02. (d) ADSpLRU with $\lambda_{\rm opt} = 0.01$ and $\tau_{\rm opt} = 0.1$, SRE (dB) = 0.78, RMSE = 0.1031, and Time (s) = 1.16. (e) JSpBLRU with $\lambda_{\rm opt} = 0.01$ and $\tau_{\rm opt} = 0.1$, SRE (dB) = 2.48, RMSE = 0.0967, and Time (s) = 1.40.

much faster than ADSpLRU and JSpBLRU. This results from the computation burden of SVDs at each ADSpLRU and JSpBLRU iteration round. Noticeably, JSpBLRU is faster than ADSpLRU for SNR $\geq$ 20 dB.

To further compare the performance of different unmixing algorithms, we list the average SRE (dB), RMSE, and Time (s) values for SNR = 20, 30, and 40 dB in Table III. The results of other SNRs show a similar conclusion, and therefore, we omit them here. Furthermore, we perform the competing four algorithms one more realization for Gaussian noise and correlated noise of SNR = 30 dB and show the estimated abundance matrices in Figs. 4 and 5, respectively. Images for other SNRs, e.g., SNR $\geq$ 20 dB, show a similar conclusion, and so again, we omit them here. From Figs. 4 and 5, we see that JSpBLRU eliminates many low abundance values supposing to be zero, reduces the degree of freedom in solutions, and gives structural fractional abundances with joint-sparsity blocks as expected.

Example 4 (General Distribution of Abundances): In this test, we consider two cases with more general distribution of the abundances: there exists one pixel (case 1) or two pixels (case 2) having different constituent endmembers compared with other pixels in a sliding window. True abundance matrices are partitioned to three blocks with sparsity levels 20%, 15%, and 10%, respectively. The number of endmembers $m = 50$ and the number of pixels $K = 9$. Without loss of generality, the fifth pixel in case 1 and the third and the seventh pixels in case 2 contain different constituent endmembers. The true abundance matrices have ranks 4 and 5 for cases 1 and 2, respectively. The spectral library of each case is constructed by randomly selecting 50 endmembers from $\mathbf{A}_1$. The observation data are generated by the LMM and then corrupted by the Gaussian noise with SNR = 30 dB. For both cases 1 and 2, we perform 50 times independently. Average RMSE, SRE (dB), and Time (s) are recorded in Table IV. Figs. 6 and 7 show the true and estimated abundances by four different unmixing algorithms after one more realization for cases 1 and 2, respectively.

From Table IV, we see that SUnSAL and CLSUnSAL are fast for both cases. ADSpLRU and JSpBLRU provide better RMSE and SRE (dB) values with more computational time. Clearly, JSpBLRU gives best RMSE and SRE (dB) results.

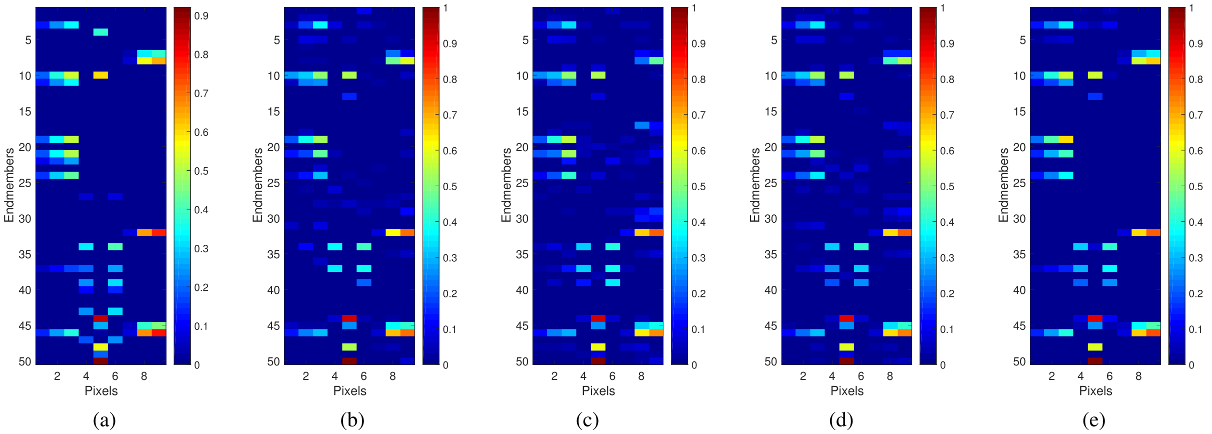


Fig. 6. Estimated abundance matrices of different unmixing algorithms when the simulated observations are corrupted by Gaussian noise with SNR = 30 dB for Example 4. (a) True abundance matrix. (b) SUnSAL with $\lambda_{\text{opt}} = 0.05$, SRE (dB) = 8.56, RMSE = 0.0500, and Time (s) = 0.03. (c) CLSUnSAL with $\lambda_{\text{opt}} = 0.01$, SRE (dB) = 8.04, RMSE = 0.0530, and Time (s) = 0.03. (d) ADSpLRU with $\lambda_{\text{opt}} = 0.005$ and $\tau_{\text{opt}} = 0.05$, SRE (dB) = 9.54, RMSE = 0.0445, and Time (s) = 0.84. (e) JSpBLRU with $\lambda_{\text{opt}} = 0.005$ and $\tau_{\text{opt}} = 0.05$, SRE (dB) = 10.68, RMSE = 0.0417, and Time (s) = 0.24.

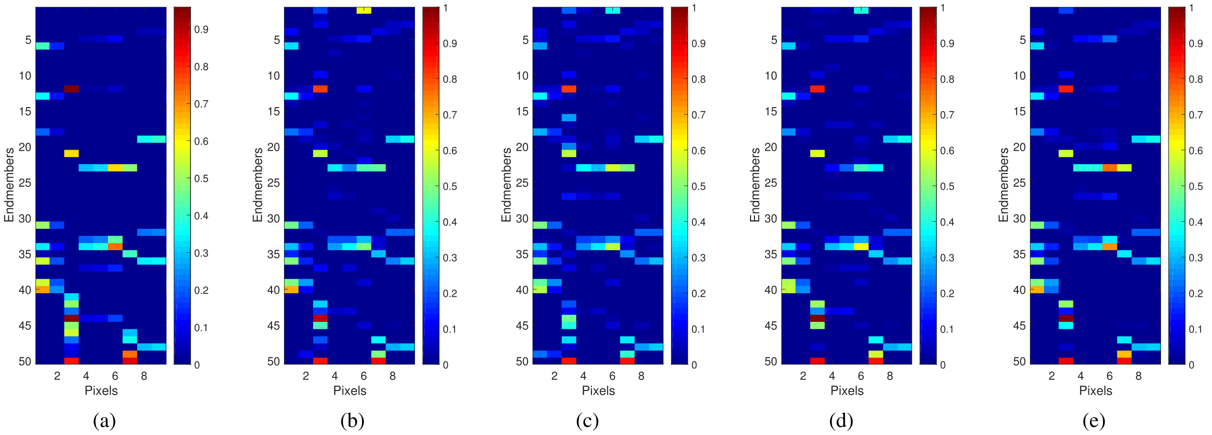


Fig. 7. Estimated abundance matrices of different unmixing algorithms when the simulated observations are corrupted by Gaussian noise with SNR = 30 dB for Example 4. (a) True abundance matrix. (b) SUnSAL with $\lambda_{\text{opt}} = 0.01$, SRE (dB) = 7.62, RMSE = 0.0570, and Time (s) = 0.02. (c) CLSUnSAL with $\lambda_{\text{opt}} = 0.01$, SRE (dB) = 6.22, RMSE = 0.0636, and Time (s) = 0.04. (d) ADSpLRU with $\lambda_{\text{opt}} = \tau_{\text{opt}} = 0.01$, SRE (dB) = 8.90, RMSE = 0.0505, and Time (s) = 0.71. (e) JSpBLRU with $\lambda_{\text{opt}} = \tau_{\text{opt}} = 0.001$, SRE (dB) = 10.00, RMSE = 0.0477, and Time (s) = 0.22.

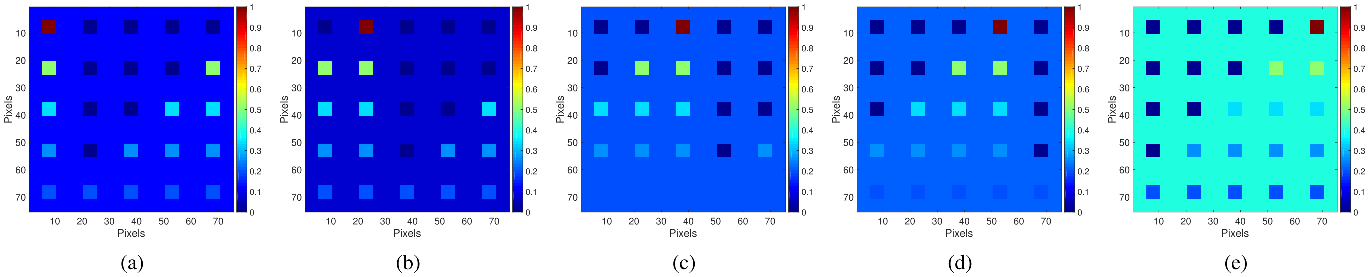


Fig. 8. True abundance maps of selected endmembers for Example 5. (a) Endmember #1. (b) Endmember #2. (c) Endmember #3. (d) Endmember #4. (e) Endmember #5.

In addition, for each algorithm, the RMSE and SRE (dB) values of case 1 are better than those of case 2.

Fig. 6(a) shows that the fifth pixel is composed of different constituent endmembers from other pixels. Fig. 6(b)–(d) shows that SUnSAL, CLSUnSAL, and ADSpLRU maintain the overall abundance structure but produce many low abundance values. JSpBLRU eliminates most of these values and provides clear structural abundances in Fig. 6(e). A similar conclusion can be obtained from Fig. 7. In addition, we see from Figs. 6 and 7 that for each algorithm, the estimated abundances in case 2 are less accurate than those in case 1. These observations are in line with those already recorded in Table IV. Finally, it is worth noting that in this test, JSpBLRU not only gives abundances with the joint-sparsity-blocks structure for pixels having similar constituent endmembers as expected but also attains competitive abundances for pixels having different constituent endmembers within one block.

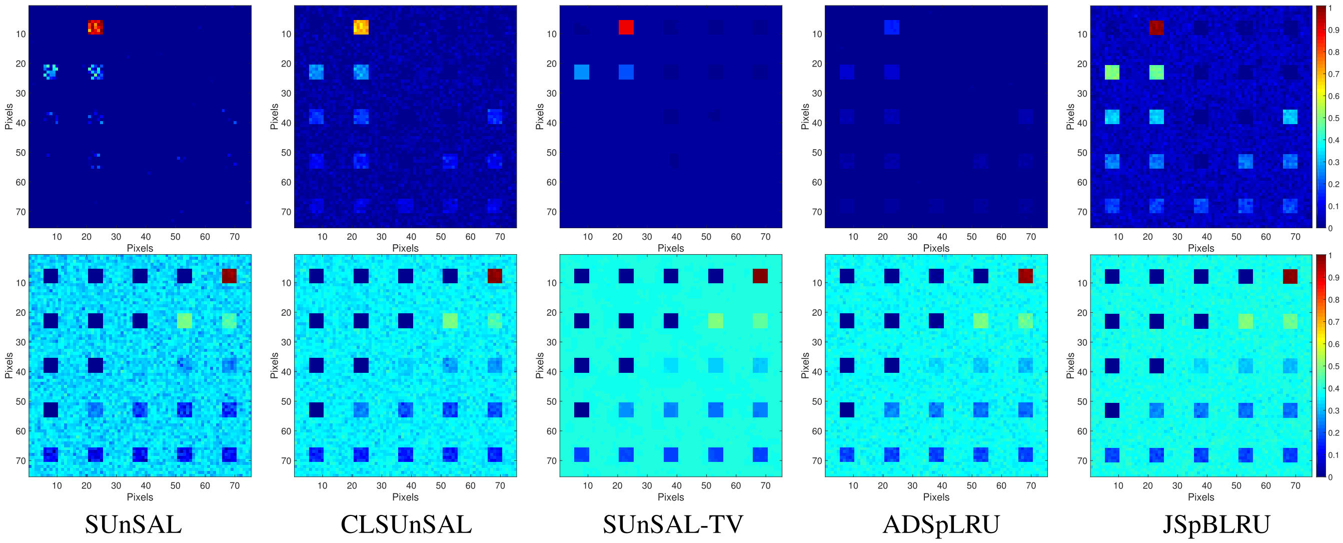


Fig. 9. Estimated abundance maps for (Top row) endmembers #2 and (Bottom row) #5 by different unmixing algorithms for one realization of Example 5. (a) SUnSAL with $\lambda_{\text{opt}} = 0.005$, SRE (dB) = 7.85, RMSE = 0.0119, and Time (s) = 101.20. (b) CLSUnSAL with $\lambda_{\text{opt}} = 0.1$, SRE (dB) = 10.60, RMSE = 0.0090, and Time (s) = 109.69. (c) SUnSAL-TV with $\lambda_{\text{opt}} = 5 \times 10^{-4}$ and $\lambda_{\text{TV,opt}} = 0.005$, SRE (dB) = 15.46, RMSE = 0.0055, and Time (s) = 1559.90. (d) ADSpLRU with $\lambda_{\text{opt}} = 5 \times 10^{-4}$ and $\tau_{\text{opt}} = 10$, SRE (dB) = 10.80, RMSE = 0.0090, and Time (s) = 681.37. (e) JSpBLRU with $\lambda_{\text{opt}} = 5 \times 10^{-4}$ and $\tau_{\text{opt}} = 1$, SRE (dB) = 17.51, RMSE = 0.0044, and Time (s) = 693.51.

TABLE IV

AVERAGE RMSE, SRE (dB), AND TIME (s) AFTER 50 RUNS OF DIFFERENT UNMIXING ALGORITHMS WHEN THE SIMULATED OBSERVATIONS ARE CORRUPTED BY GAUSSIAN NOISE WITH SNR = 30 dB FOR EXAMPLE 4

Case 1				
Criteria	SUnSAL	CLSUnSAL	ADSpLRU	JSpBLRU
RMSE	0.0574	0.0584	0.0509	0.0476
SRE (dB)	7.28	7.06	8.62	9.54
Time (s)	0.04	0.06	0.67	0.24
Case 2				
Criteria	SUnSAL	CLSUnSAL	ADSpLRU	JSpBLRU
RMSE	0.0650	0.0641	0.0607	0.0577
SRE (dB)	6.87	6.42	7.60	8.21
Time (s)	0.04	0.06	0.65	0.17

Example 5 (Synthetic HSI I): We illustrate the performance of JSpBLRU on a widely used simulated data cube, for instance, in [8] and [27], which contains 75 × 75 pixels with 224 bands per pixel. The spectral library $\mathbf{A}_2 \in \mathbb{R}^{224\times 240}$ is used in this experiment. We generate the test data according to the LMM by five randomly selected spectral signatures from $\mathbf{A}_2$ as endmembers. True fractional abundances for each of the five endmembers are shown in Fig. 8. After generated by the LMM, the data cube is corrupted by white Gaussian i.i.d. noise with SNR = 30 dB. We run 10 times independently. The estimated abundance maps of endmembers #2 and #5, after one realization, by different unmixing algorithms are shown in Fig. 9. Abundance maps for other endmembers show a similar behavior, so we omit here for space considerations. Notice that the abundance matrix is simultaneously low-rank and sparse. We thus apply the ADSpLRU and JSpBLRU algorithms as nonsliding-window approaches, i.e., K denotes the number of pixels in the synthetic HSI here.

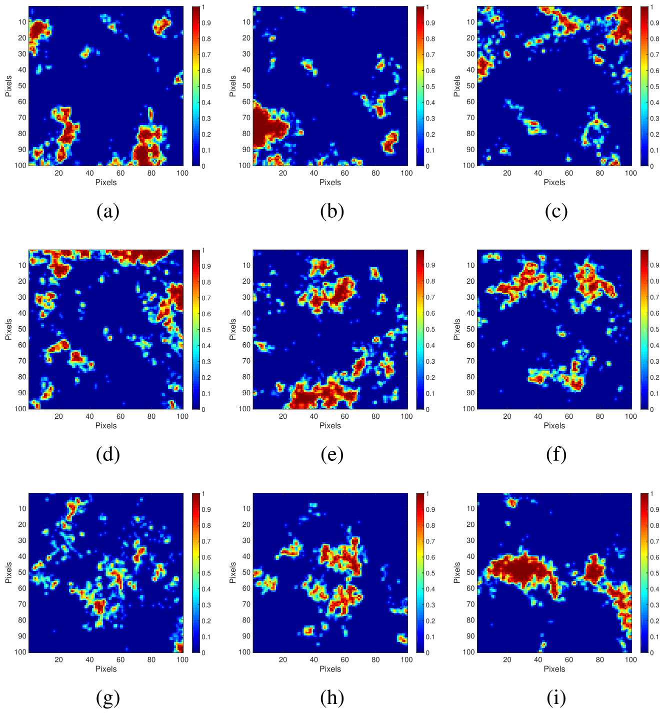
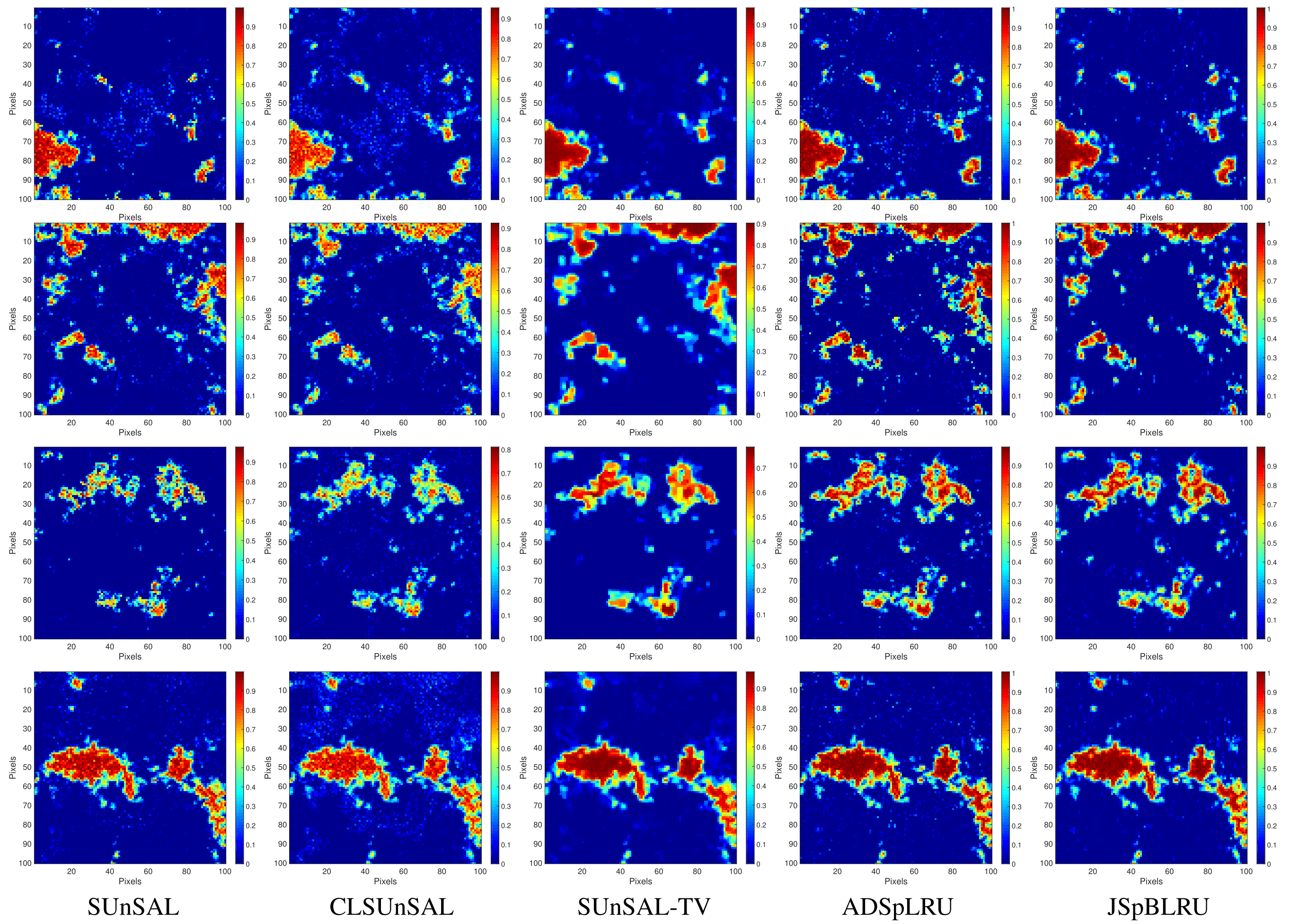


Fig. 10. True abundance maps of selected endmembers for Example 6. (a) Endmember #1. (b) Endmember #2. (c) Endmember #3. (d) Endmember #4. (e) Endmember #5. (f) Endmember #6. (g) Endmember #7. (h) Endmember #8. (i) Endmember #9.

From Fig. 9, we clearly observe that JSpBLRU delineates all square regions in the estimated abundance maps of both endmembers, especially, endmember #2, but with less smooth background. SUnSAL-TV provides a satisfactory abundance map for endmember #5. Meanwhile, though, it provides an oversmoothed one with several square regions vanished for endmember #2. Clearly, CLSUnSAL is better than SUnSAL



SUnSAL CLSUnSAL SUnSAL-TV ADSpLRU JSpBLRU

Fig. 11. Estimated abundance maps for (from top row to bottom row) endmembers #2, #4, #6, and #9 by different unmixing algorithms for one realization of Example 6. First column: SUnSAL with $\lambda_{\text{opt}} = 0.05$, SRE (dB) = 6.99, RMSE = 0.0280, and Time (s) = 85.67. Second column: CLSUnSAL with $\lambda_{\text{opt}} = 0.1$, SRE (dB) = 4.49, RMSE = 0.0344, and Time (s) = 100.08. Third column: SUnSAL-TV with $\lambda_{\text{opt}} = 0.01$ and $\lambda_{\text{TV,opt}} = 0.005$, SRE (dB) = 11.71, RMSE = 0.0166, and Time (s) = 1185.05. Fourth column: ADSpLRU with $\lambda_{\text{opt}} = 0.01$ and $\tau_{\text{opt}} = 10$, SRE (dB) = 16.46, RMSE = 0.0108, and Time (s) = 483.80. Fifth column: JSpBLRU with $\lambda_{\text{opt}} = 0.005$ and $\tau_{\text{opt}} = 1$, SRE (dB) = 18.23, RMSE = 0.0089, and Time (s) = 720.20.

for both endmembers. In addition, ADSpLRU and JSpBLRU produce similar fractional abundance maps for endmember #5. However, the estimated background by JSpBLRU is slightly smoother than that by ADSpLRU.

Moreover, the average RMSE and SRE (dB) values after 10 realizations of JSpBLRU, from Table V, are better than other algorithms, in line with the qualitative observations on abundance maps already shown in Fig. 9. It is worth noting that SUnSAL-TV costs more running time than others. The running time of JSpBLRU is more than that of ADSpLRU and much more than that of SUnSAL and CLSUnSAL due to the SVDs per JSpBLRU and ADSpLRU iteration.

Example 6 (Synthetic HSI II): This example demonstrates the effectiveness of JSpBLRU on another widely used simulated data set. The spectral library matrix is $\mathbf{A}_3$, which has also been used in [9], [13], and [68]. Nine signatures are randomly chosen from $\mathbf{A}_3$ and then used to generate a true 100 × 100-pixel data cube by the LMM. The true fractional abundances, also used in [8], [9], and [13], are shown in Fig. 10. After the above-mentioned procedure, the true data cube is contaminated by white Gaussian i.i.d. noise with SNR = 30 dB. We perform 10 independent realizations.

Fig. 11 shows the estimated abundance maps of endmembers #2, #4, #6, and #9 by five unmixing algorithms after one realization. Other estimated abundance maps show a similar behavior and therefore omit them here. From Fig. 11, we observe that all five algorithms attain abundance maps with accurate spatial distribution. SUnSAL and CLSUnSAL, however, delineate the regions with high fractional abundance of endmembers #4 and #6 with less accuracy. SUnSAL-TV provides abundance maps with spatial consistency as expected. Both ADSpLRU and JSpBLRU well delineate high fractional abundance regions for all considered endmembers, whereas JSpBLRU better reduces toward zero the low fractional abundance values that are not present in the true abundance maps. For further comparison, Fig. 12 shows the true abundances of selected adjacent 100 pixels in this realization and the estimations by the considered unmixing algorithms. Clearly, JSpBLRU produces more similar fractional abundances to the ground truth than other compared algorithms.

Moreover, we list the average RMSE, SRE (dB), and Time (s) values after 10 realizations of different unmixing algorithms for this example in Table V. From this table, we can see that SUnSAL and CLSUnSAL take much less

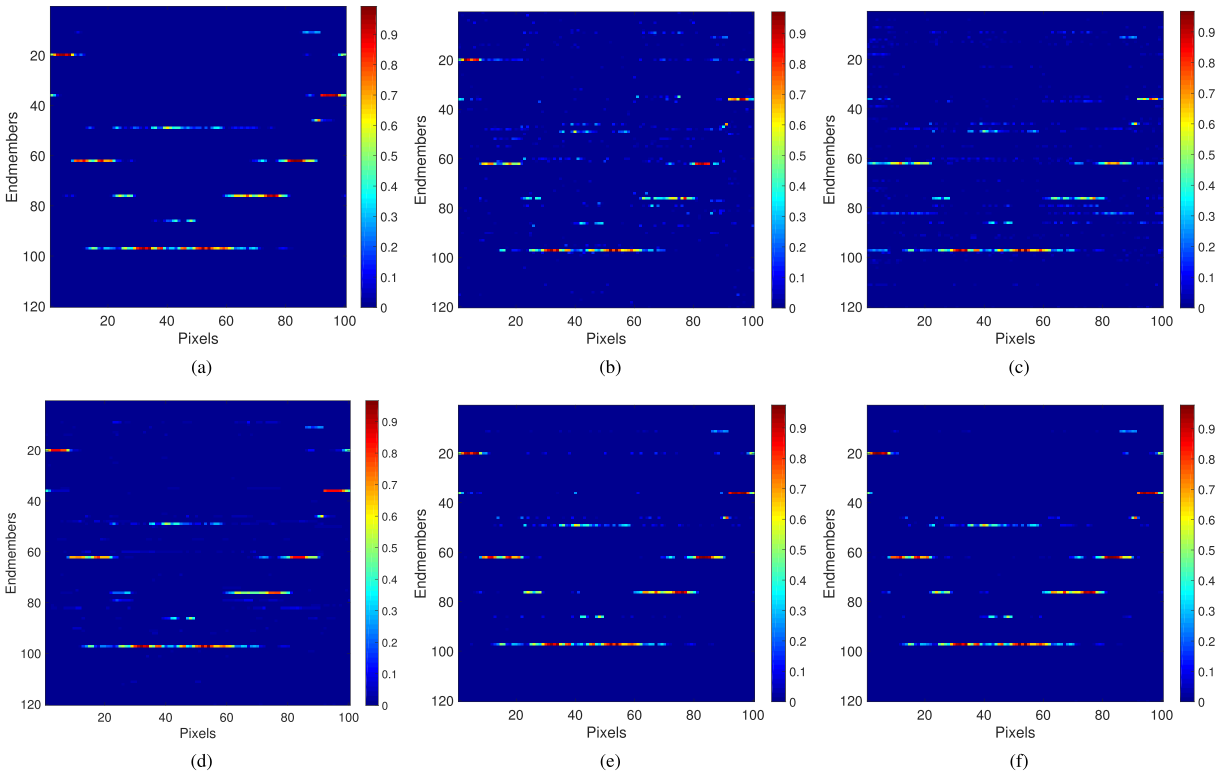


Fig. 12. True abundances of 100 selected adjacent pixels and the estimations by different algorithms for one realization of Example 6. (a) True abundances. (b) SUnSAL. (c) CLSUnSAL. (d) SUnSAL-TV. (e) ADSpLRU. (f) JSpBLRU.

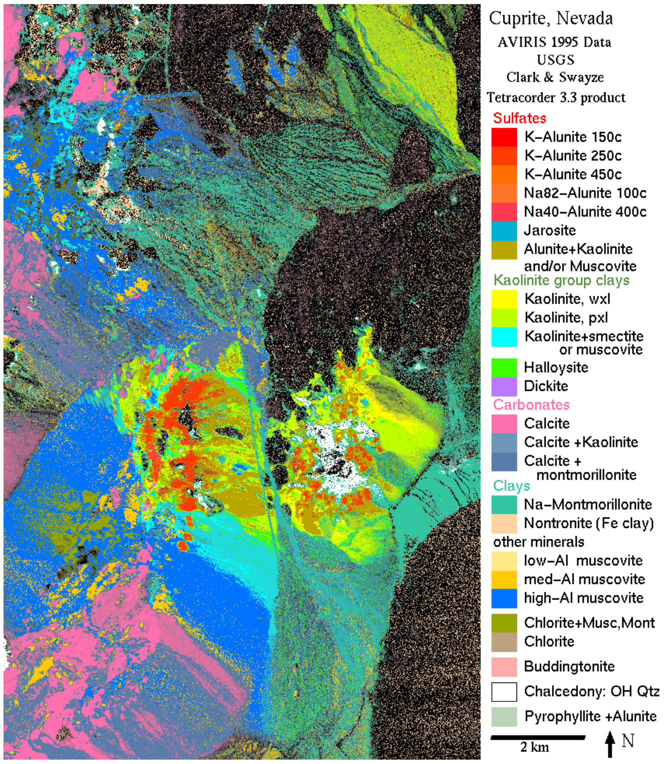


Fig. 13. USGS map showing the location of different minerals in the Cuprite mining disctrict in Nevada.

TABLE V

Average RMSE, SRE (dB), and Time (s) After 10 Runs by Different Unmixing Algorithms When the Simulated Observations Are Corrupted by Gaussian Noise With SNR = 30 dB for Examples 5 and 6

Example 5					
Criteria	SUnSAL	CLSUnSAL	SUnSAL-TV	ADSpLRU	JSpBLRU
RMSE	0.0162	0.0105	0.0057	0.0100	0.0041
SRE (dB)	5.94	9.75	16.55	11.03	19.90
Time (s)	97.47	105.44	1444.25	586.45	642.05
Example 6					
Criteria	SUnSAL	CLSUnSAL	SUnSAL-TV	ADSpLRU	JSpBLRU
RMSE	0.0360	0.0375	0.0230	0.0176	0.0134
SRE (dB)	4.45	3.50	8.68	12.81	15.66
Time (s)	76.42	87.68	1172.45	497.33	625.55

computational time than other algorithms. JSpBLRU uses more computation time than ADSpLRU and SUnSAL-TV costs most among all compared algorithms. We can also see that JSpBLRU attains the lowest RMSE and highest SRE (dB) values, in line with the qualitative observations in Figs. 11 and 12.

B. Experiment on Real Data

In this test, we demonstrate the performance of the proposed JSpBLRU algorithm on the well-known Airborne Visible/Infrared Imaging Spectrometer (AVIRIS) Cuprite data set.[4] Fig. 13 shows a mineral map produced in 1995 by USGS. We use a square 350 × 350 pixel subscene with 188 spectral bands. The 188 × 240 spectral library matrix in this experiment is generated from the USGS library that includes all exposed minerals of interest. This real data cube has been widely applied to validate the effectiveness of unmixing algorithms in the literature [9], [11], [13] (see [7], [8], [10] for more details).

[4] Available online: http://aviris.jpl.nasa.gov/html/aviris.freedata.html.

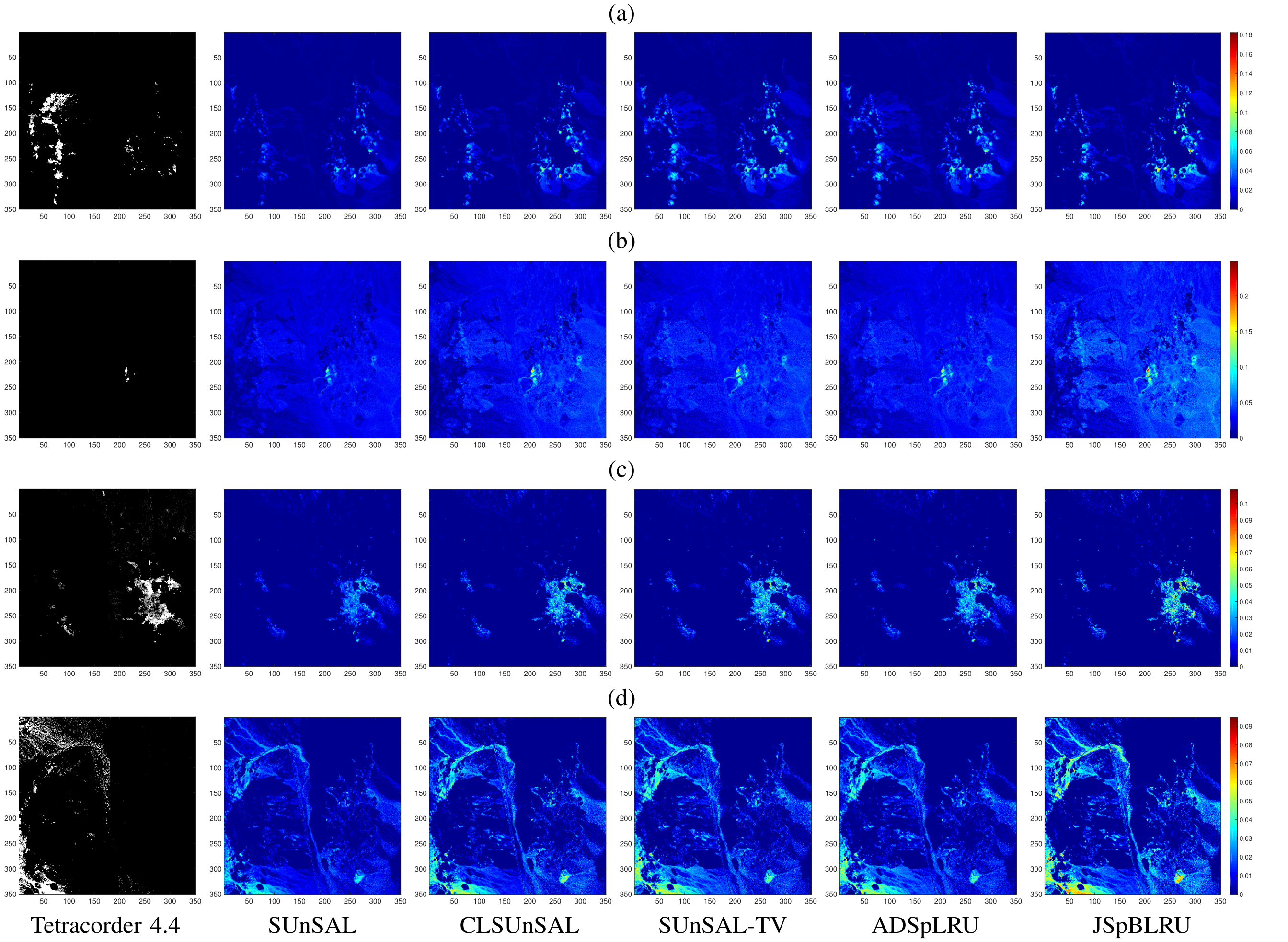


Fig. 14. Qualitative comparison between the classification maps of the AVIRIS Cuprite subscene produced by Tetracorder 4.4 and the abundance maps estimated by different unmixing algorithms for four different minerals. (a) Alunite. (b) Buddingtonite. (c) Chalcedony. (d) Muscovite.

Similarly, as in [10]–[12], the regularization parameters for SUnSAL and CLSUnSAL in this experiment were empirically set to 0.001 and 0.01, respectively. Also, we set both regularization parameters $\lambda = \lambda_{\text{TV}} = 0.001$ for SUnSAL-TV, as in [8], and use fine-tuned parameters $\lambda = \tau = 0.001$ for both ADSpLRU and JSpBLRU. Notice that the abundance matrix is already low-rank and sparse, and therefore, in this experiment, we apply ADSpLRU and JSpBLRU as nonsliding-window approaches, as in Examples 5 and 6 in Section IV-A. Since the detailed ground-truth information is unavailable, we just make a qualitative comparison with reference to the Tetracorder 4.4 software product[5] [69].

Fig. 14 shows the estimated abundance maps obtained by SUnSAL, CLSUnSAL, SUnSAL-TV, ADSpLRU, and JSpBLRU for four minerals: *alunite*, *buddingtonite*, *chalcedony*, and *muscovite*. From Fig. 14, it can be observed that all five unmixing algorithm produce similar abundance maps. Nevertheless, the fractional abundances estimated by JSpBLRU are generally comparable or higher in the regions considered as respective materials in comparison with those by other algorithms. Generally speaking, by simultaneously imposing joint-sparse blocks and low rankness on abundances, JSpBLRU qualitatively gives the comparable unmixing results of the considered real HSI.

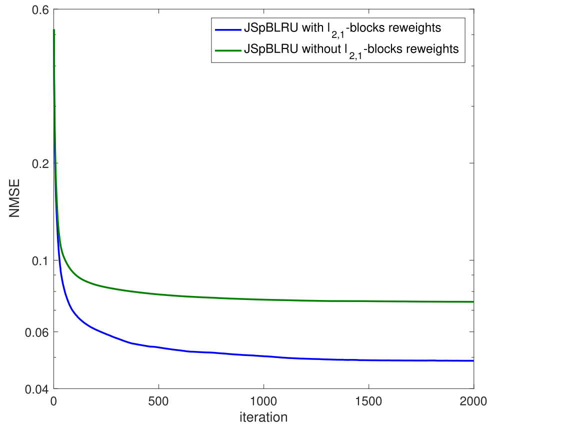


Fig. 15. Convergence histories of JSpBLRU with and without reweighting coefficients for $\ell_{2,1}$-blocks regularization.

C. Parameters Selection

In this section, we discuss four parameters selection of JSpBLRU, including the reweighting $\ell_{2,1}$-blocks coefficients

[5] Available online: https://speclab.cr.usgs.gov/PAPERS/tetracorder/.

TABLE VI

PERFORMANCE OF JSpBLRU WITH DIFFERENT BLOCK SIZE d VALUES FOR ONE REALIZATION OF EXAMPLES 5 AND 6 EACH. NOTE THAT n IS THE TOTAL NUMBER OF PIXELS

Example 5									
d	1	3	5	7	9	20	50	200	n
RMSE	0.0089	0.0083	0.0085	0.0078	0.0078	0.0079	0.0075	0.0075	0.0080
SRE (dB)	11.04	11.75	11.49	12.48	12.55	12.34	12.70	12.98	12.23
Time (s)	900.18	836.29	823.47	774.34	772.34	760.39	841.76	771.60	350.72
λ_{opt}	0.001	0.001	0.001	0.01	0.01	0.01	0.05	0.1	0.5
τ_{opt}	10	10	5	10	10	5	5	5	5
Example 6									
d	1	3	5	7	9	20	50	200	n
RMSE	0.0101	0.0089	0.0093	0.0118	0.0122	0.0179	0.0243	0.0292	0.0294
SRE (dB)	17.12	18.23	17.65	15.72	15.38	11.74	9.03	6.99	6.75
Time (s)	674.13	592.39	575.15	549.02	546.10	517.05	200.03	231.95	299.9
λ_{opt}	0.01	0.005	0.005	0.05	0.05	0.005	0.01	0.005	0.1
τ_{opt}	5	1	0.5	1	1	0.1	0.1	0.5	0.5

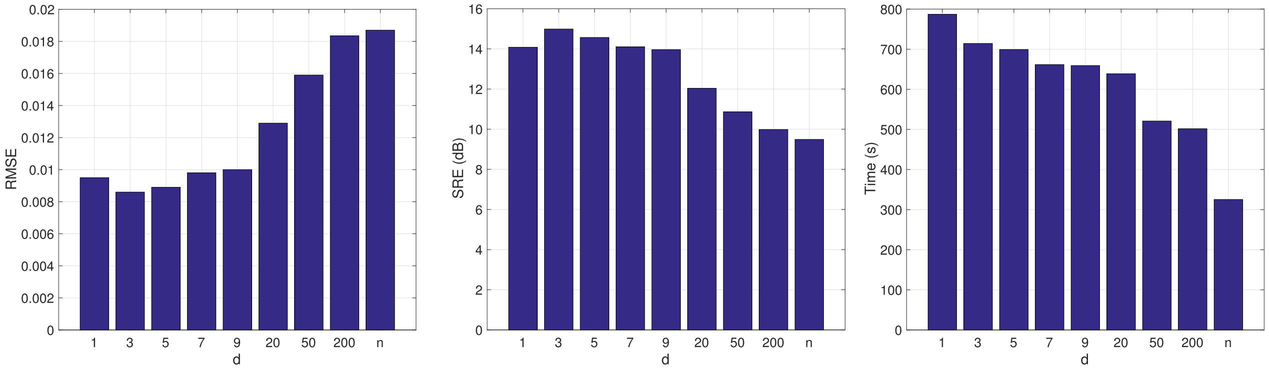


Fig. 16. Average RMSE, SRE (dB), and Time (s) of Examples 5 and 6 (one realization each) against d in JSpBLRU. Note that n is the total pixel number.

$w_{i,j}^k$ in (25), each block size d in (29) or (30), the sliding window size, and the regularization parameters λ and τ.

1) Comparisons of JSpBLRU With and Without $\ell_{2,1}$-Blocks Reweighting Coefficients: We aim to demonstrate the efficiency of the reweighting $\ell_{2,1}$-blocks coefficients $w_{i,j}^k$ in (25) of JSpBLRU. We mention that the weighted nuclear norm strategy in (26) is applied for JSpBLRU with and without $\ell_{2,1}$-blocks reweighting coefficients. We do 50 realizations. True 50×9 abundance matrices are of rank 3 and can be partitioned to three blocks with three pixels per one block. The three blocks are of sparsity levels 15%, 10%, and 5%, respectively. We randomly select 50 endmembers from $\mathbf{A}_1$ to construct the spectral library. The spectrum matrix is generated according to the LMM and contaminated by Gaussian noise with the SNR of 30 dB. The maximum number of iterations is set to be 2000. Regularization parameters λ and τ are fine-tuned for JSpBLRU with and without $\ell_{2,1}$-blocks reweights.

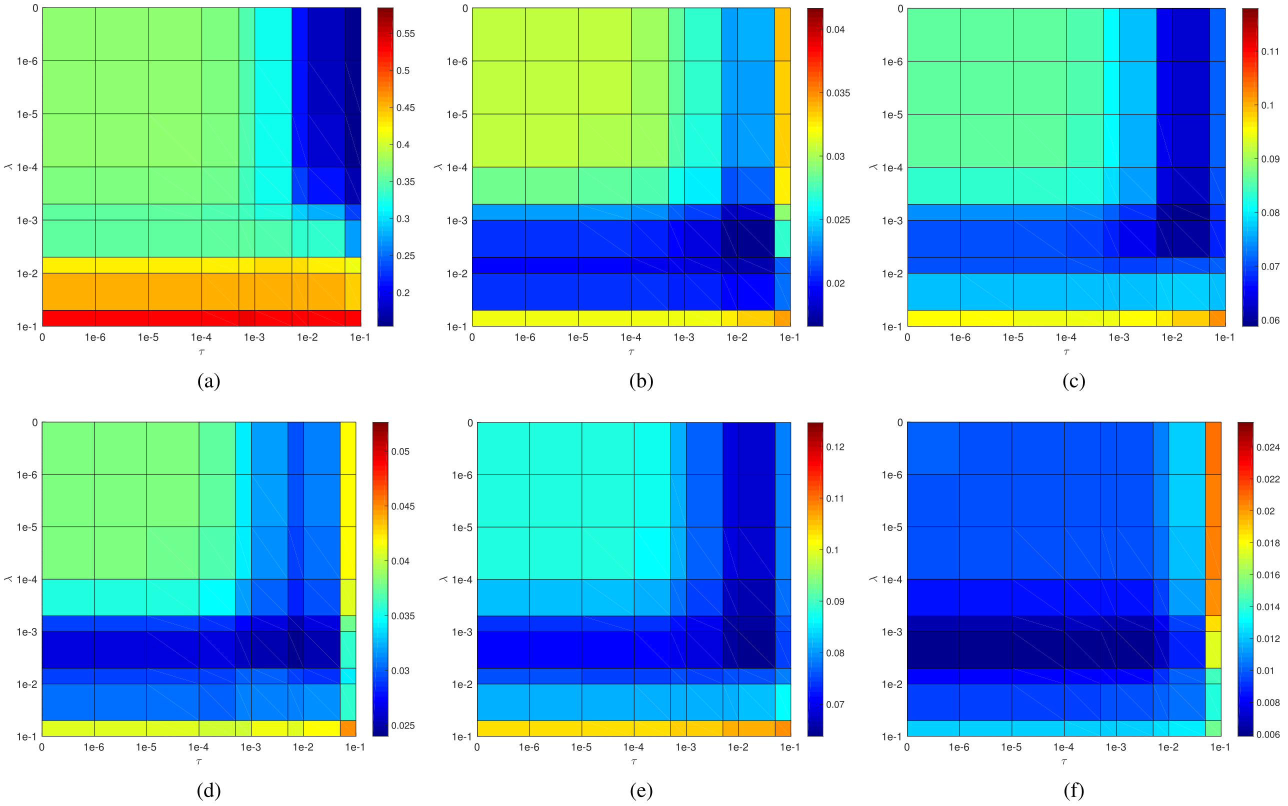
Fig. 15 shows the normalized mean square estimation error (NMSE) that is defined as

$$\text{NMSE}(k) = \frac{1}{t}\sum_{i=1}^{t}\frac{\|\hat{\mathbf{X}}_{(i)}^k - \mathbf{X}_{(i)}\|_F^2}{\|\mathbf{X}_{(i)}\|_F^2}$$

where $\mathbf{X}_{(i)}$ and $\hat{\mathbf{X}}_{(i)}^k$ are the exact abundance matrix and its estimation at the kth iteration of the ith realization, respectively, as the iteration number increases. Here, we set $t = 50$. From Fig. 15, we observe that JSpBLRU provides lower NMSE values with $\ell_{2,1}$-blocks reweights than without $\ell_{2,1}$-blocks reweights. Moreover, though the reweighting strategies in (25) and (26) render the model (15) nonconvex and convergence has not been theoretically guaranteed yet, JSpBLRU presents a robust convergence behavior.

2) Each Block Size Selection in JSpBLRU: We explore the influence of number of pixels per one block, denoted as d in (29) or (30), to choose an optimal d for JSpBLRU. We consider $d = 1, 3, 5, 7, 9, 20, 50, 200$, and the total number of pixels n for Examples 5 and 6 in Section IV-A. Regularization parameters λ and τ are fine-tuned for optimal RMSE values. Table VI lists the RMSE, SRE (dB), and Time (s) values by JSpBLRU with different values of d, and corresponding optimal λ and τ, i.e., λ_{opt} and τ_{opt}, respectively.

From Table VI, we see that JSpBLRU with $d \geq 7$ gives competitive RMSE and SRE (dB) values. It says that Example 5 prefers a larger d value in JSpBLRU. Moreover, $d = 200$ is optimal among examined values of d for Example 5. A simple calculation gives that the five true abundance maps shown in Fig. 8 have total 95.56% nonzero elements, meaning that each mixed pixel probably has nonzero fractions at the same positions. As a result, forcing the joint sparsity within a larger block gives better unmixing results, since spatial information among adjacent pixels has been better considered. For Example 6, however, $d \leq 5$ gives competitive abundance estimations and $d = 3$ is the best. JSpBLRU here shows a preference for a smaller d value, in line with the observation from Fig. 12 that the local neighboring pixels



(a) (b) (c)

(d) (e) (f)

Fig. 17. Average RMSE as a function of sparsity parameter λ and low-rank parameter τ in JSpBLRU for different low rankness and sparsity levels for Example 2. (a) Rank 1, sparsity level 100%. (b) Rank 2, sparsity level 10%. (c) Rank 2, sparsity level 20%. (d) Rank 3, sparsity level 10%. (e) Rank 3, sparsity level 20%. (f) Rank 9, sparsity level 5%.

TABLE VII

PERFORMANCE OF JSPBLRU FOR DIFFERENT SLIDING WINDOW SIZES AND LRLS

K	1	9	25	49	81
LRL	1	4.5	4.17	3.77	3.125
RMSE	0.0543	0.0514	0.0551	0.0640	0.0627
SRE (dB)	5.97	9.07	7.86	6.69	6.76
Time (s)	0.03	0.17	0.90	2.28	4.14

within a smaller block exhibit a stronger joint-sparse structure. In other words, it is hard to get an optimal value of d for both Examples 5 and 6.

In order to obtain a good estimation of d, we plot average RMSE, SRE (dB), and Time (s) values of Examples 5 and 6 against d in Fig. 16. We can observe from this figure that $d = 3$ gives better average RMSE and SRE (dB) values. The runtime decreases as d increases. Overall, $d = 3$ provides competitive unmixing results, though it may demand more runtime compared with other larger values.

3) Sliding Window Size Selection: We compare the performance of JSpBLRU against the pixel number K in the square sliding window to choose an optimal sliding window size for hyperspectral unmixing problems. We also consider different low rankness levels (LRLs) defined in [12]

$$\text{LRL} = \frac{\min\{K, m\}}{\text{rank } (\mathbf{X})}$$

where m is the number of endmembers and $\mathbf{X}$ is the true abundance matrix. Similarly, as in [12], the LRL decreases as the size of K increases in our tests. Here, we set the number of endmembers $m = 50$ and the sparsity level of the exact abundance matrix to be 20%. We randomly select 50 endmembers from $\mathbf{A}_1$ to built the spectral library. The exact data are generated according to the LMM and then corrupted by Gaussian noise with SNR $=$ 30 dB. We perform 50 independent realizations. The RMSE, SRE (dB), and Time (s) values of different values of K and LRL are reported in Table VII.

From Table VII, we observe that $K = 9$, i.e., a 3×3 sliding window, provides best RMSE and SRE (dB) results. The computation time increases as the number of pixels K increases. This is expected since the size of the abundance matrix increasing requires more runtime.

4) Role of Regularization Parameters λ and τ: From the model (15), we observe that regularization parameters λ and τ control the tradeoff between sparsity and low rankness. They are crucial but hard to obtain their optimal values in real applications. In all our simulations, unless specifically stated, we consider all possible combinations of λ and τ in (27) and choose the optimal parameter pair according to the minimum RMSE value. Without loss of generality, we consider the influence of sparse parameter λ and low-rank parameter τ on JSpBLRU for the six experiments in Example 2 in Section IV-A. For each experiment, we show an average RMSE, after 50 independent runs, as a function of λ and τ in Fig. 17.

From Fig. 17(a), we see that optimal sparse parameter λ can be small, even close to zero. It is reasonable because the corresponding abundance matrix is low-rank but dense, and thus, the sparsity prior has little impact on abundance estimation. From Fig. 17(b)–(e), we see that the average optimal parameters λ and τ are about 0.001 and 0.01, respectively. This means that both two parameters have an impact on abundance estimation. From Fig. 17(f), we see that average optimal τ can be values from 0 to 0.01, but optimal λ is about 0.001. It implies that compared with the sparse parameter λ, the low-rank parameter τ plays a less important role for estimating abundance matrix with rank 9 and sparsity level 5%. This results from the observation that the sparsity prior of the abundance matrix is more dominant than the low rankness prior. It should be noted that optimal λ and τ are often problem-dependent, and thus, fine-tuning techniques are suggested to use in JSpBLRU to achieve better unmixing results.

V. Conclusion

In this paper, we have proposed a joint-sparse-blocks regression problem, which promotes pixels belonging to the same block to share the same support set of endmembers. The proposed joint-sparsity-blocks structure considers not only sparsity in a single pixel but also spatial correlation among nearby pixels. It generalizes the classic single and joint sparsities. For the hyperspectral unmixing problem, we have proposed a simultaneously JSpBLRU model. We solve this model under the ADMM framework, leading to a new algorithm called JSpBLRU. In particular, we have designed a two-level $\ell_{2,1}$-blocks reweighting strategy for JSpBLRU to enhance the sparsity along the rows in each block. Simulated and real-data experiments have demonstrated the efficacy of the proposed algorithm. In the future, we will extend the proposed joint-sparsity-blocks structure to blind unmixing or its tensor edition for HSI processing.

Acknowledgment

The authors would like to thank the editors and the three anonymous reviewers for their constructive, detailed, and helpful advice regarding this paper.

References

[1] W.-K. Ma *et al.*, "A signal processing perspective on hyperspectral unmixing: Insights from remote sensing," *IEEE Signal Process. Mag.*, vol. 31, no. 1, pp. 67–81, Jan. 2014.

[2] J. Boardman, F. A. Kruse, and R. O. Green, "Mapping target signatures via partial unmixing of AVIRIS data," in *Proc. JPL Airborne Earth Sci. Workshop*, vol. 1, 1995, pp. 23–26.

[3] J. Li and J. Bioucas-Dias, "Minimum volume simplex analysis: A fast algorithm to unmix hyperspectral data," in *Proc. IEEE IGARSS*, vol. 3, Jul. 2008, pp. III-250–III-253.

[4] 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, Apr. 2005.

[5] A. Plaza, G. Martín, J. Plaza, M. Zortea, and S. Sánchez, "Recent developments in endmember extraction and spectral unmixing," in *Optical Remote Sensing: Advances in Signal Processing and Exploitation Techniques*, S. Prasad, L. Bruce, and J. Chanussot, Eds. Berlin, Germany: Springer, 2011, pp. 235–267.

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

[7] M.-D. Iordache, J. Bioucas-Dias, and A. Plaza, "Sparse unmixing of hyperspectral data," *IEEE Trans. Geosci. Remote Sens.*, vol. 49, no. 6, pp. 2014–2039, Jun. 2011.

[8] M.-D. Iordache, J. Bioucas-Dias, and A. Plaza, "Total variation spatial regularization for sparse hyperspectral unmixing," *IEEE Trans. Geosci. Remote Sens.*, vol. 50, no. 11, pp. 4484–4502, Nov. 2012.

[9] X.-L. Zhao, F. Wang, T.-Z. Huang, M. K. Ng, and R. J. Plemmons, "Deblurring and sparse unmixing for hyperspectral images," *IEEE Trans. Geosci. Remote Sens.*, vol. 51, no. 7, pp. 4045–4058, Jul. 2013.

[10] M.-D. Iordache, J. M. Bioucas-Dias, and A. Plaza, "Collaborative sparse regression for hyperspectral unmixing," *IEEE Trans. Geosci. Remote Sens.*, vol. 52, no. 1, pp. 341–354, Jan. 2014.

[11] W. Tang, Z. Shi, Y. Wu, and C. Zhang, "Sparse unmixing of hyperspectral data using spectral a priori information," *IEEE Trans. Geosci. Remote Sens.*, vol. 53, no. 2, pp. 770–783, Feb. 2015.

[12] P. V. Giampouras, K. E. Themelis, A. A. Rontogiannis, and K. D. Koutroumbas, "Simultaneously sparse and low-rank abundance matrix estimation for hyperspectral image unmixing," *IEEE Trans. Geosci. Remote Sens.*, vol. 54, no. 8, pp. 4775–4789, Aug. 2015.

[13] G. Zhang, Y. Xu, and F. Fang, "Framelet-based sparse unmixing of hyperspectral images," *IEEE Trans. Image Process.*, vol. 25, no. 4, pp. 1516–1529, Apr. 2016.

[14] D. C. Heinz and C.-I. Chang, "Fully constrained least squares linear spectral mixture analysis method for material quantification in hyperspectral imagery," *IEEE Trans. Geosci. Remote Sens.*, vol. 39, no. 3, pp. 529–545, Mar. 2001.

[15] N. Dobigeon, J.-Y. Tourneret, and C.-I. Chang, "Semi-supervised linear spectral unmixing using a hierarchical Bayesian model for hyperspectral imagery," *IEEE Trans. Signal Process.*, vol. 56, no. 7, pp. 2684–2695, Jul. 2008.

[16] K. E. Themelis, A. Rontogiannis, and K. D. Koutroumbas, "A novel hierarchical Bayesian approach for sparse semisupervised hyperspectral unmixing," *IEEE Trans. Signal Process.*, vol. 60, no. 2, pp. 585–599, Feb. 2012.

[17] A. A. Rontogiannis, K. E. Themelis, O. Sykioti, and K. D. Koutroumbas, "A fast variational Bayes algorithm for sparse semi-supervised unmixing of OMEGA/Mars express data," in *Proc. 5th WHISPERS*, Jun. 2013, pp. 1–4.

[18] D. Cai, X. He, J. Han, and T. S. Huang, "Graph regularized nonnegative matrix factorization for data representation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 33, no. 8, pp. 1548–1560, Aug. 2011.

[19] O. Eches, N. Dobigeon, and J.-Y. Tourneret, "Enhancing hyperspectral image unmixing with spatial correlations," *IEEE Trans. Geosci. Remote Sens.*, vol. 49, no. 11, pp. 4239–4247, Nov. 2011.

[20] X. Lu, H. Wu, Y. Yuan, P. Yan, and X. Li, "Manifold regularized sparse NMF for hyperspectral unmixing," *IEEE Trans. Geosci. Remote Sens.*, vol. 51, no. 5, pp. 2815–2826, May 2013.

[21] Q. Qu, N. M. Nasrabadi, and T. D. Tran, "Abundance estimation for bilinear mixture models via joint sparse and low-rank representation," *IEEE Trans. Geosci. Remote Sens.*, vol. 52, no. 7, pp. 4404–4423, Jul. 2014.

[22] R. Wang and H.-C. Li, "Nonlocal similarity regularization for sparse hyperspectral unmixing," in *Proc. IEEE Int. Geosci. Remote Sens. Symp. (IGARSS)*, Jul. 2014, pp. 2926–2929.

[23] S. F. Seyyedsalehi, H. R. Rabiee, A. Soltani-Farani, and A. Zarezade, "A probabilistic joint sparse regression model for semisupervised hyperspectral unmixing," *IEEE Geosci. Remote Sens. Lett.*, vol. 14, no. 5, pp. 592–596, May 2017.

[24] X. Wang, Y. Zhong, L. Zhang, and Y. Xu, "Spatial group sparsity regularized nonnegative matrix factorization for hyperspectral unmixing," *IEEE Trans. Geosci. Remote Sens.*, vol. 55, no. 11, pp. 6287–6304, Nov. 2017.

[25] S. Huang, H. Zhang, and A. Pižurica, "A robust sparse representation model for hyperspectral image classification," *Sensors*, vol. 17, no. 9, p. 2087, Sep. 2017.

[26] R. Wang, W. Liao, H.-C. Li, H. Zhang, and A. Pižurica, "Hyperspectral unmixing by reweighted low rank and total variation," in *Proc. Workshop Hyperspectral Image Signal Process., Evol. Remote Sens.*, Los Angeles, CA, USA, Aug. 2016, pp. 1–4.

[27] J. Yang, Y.-Q. Zhao, J. C.-W. Chan, and S. G. Kong, "Coupled sparse denoising and unmixing with low-rank constraint for hyperspectral image," *IEEE Trans. Geosci. Remote Sens.*, vol. 54, no. 3, pp. 1818–1833, Mar. 2016.

[28] P. V. Giampouras, A. A. Rontogiannis, and K. D. Koutroumbas, "Low-rank and sparse NMF for joint endmembers' number estimation and blind unmixing of hyperspectral images," in *Proc. 25th Eur. Signal Process. Conf.*, Aug./Sep. 2017, pp. 1430–1434.

[29] W. Dong, G. Shi, X. Li, Y. Ma, and F. Huang, "Compressive sensing via nonlocal low-rank regularization," *IEEE Trans. Image Process.*, vol. 23, no. 8, pp. 3618–3632, Aug. 2014.

[30] T.-Y. Ji, T.-Z. Huang, X.-L. Zhao, T.-H. Ma, and G. Liu, "Tensor completion using total variation and low-rank matrix factorization," *Inf. Sci.*, vol. 326, pp. 243–257, Jan. 2016.

[31] T.-Y. Ji, N. Yokoya, X. Zhu, and T.-Z. Huang, "Nonlocal tensor completion for multitemporal remotely sensed images' inpainting," *IEEE Trans. Geosci. Remote Sens.*, vol. 56, no. 6, pp. 3047–3061, Jun. 2018.

[32] M. Rizkinia and M. Okuda, "Joint local abundance sparse unmixing for hyperspectral images," *Remote Sens.*, vol. 9, no. 12, p. 1224, 2017.

[33] W. Yin, S. Osher, D. Goldfarb, and J. Darbon, "Bregman iterative algorithms for ℓ_1-minimization with applications to compressed sensing," *SIAM J. Imag. Sci.*, vol. 1, no. 1, pp. 143–168, 2008.

[34] S. J. Wright, R. D. Nowak, and M. A. T. Figueiredo, "Sparse reconstruction by separable approximation," *IEEE Trans. Signal Process.*, vol. 57, no. 7, pp. 2479–2493, Jul. 2009.

[35] I. Daubechies, M. Defrise, and C. De Mol, "An iterative thresholding algorithm for linear inverse problems with a sparsity constraint," *Commun. Pure Appl. Math.*, vol. 57, no. 11, pp. 1413–1457, Nov. 2004.

[36] J. Huang, M. Donatelli, and R. Chan, "Nonstationary iterated thresholding algorithms for image deblurring," *Inverse Problems Imag.*, vol. 7, no. 3, pp. 717–736, 2013.

[37] X.-L. Zhao, F. Wang, and M. K. Ng, "A new convex optimization model for multiplicative noise and blur removal," *SIAM J. Imag. Sci.*, vol. 7, no. 1, pp. 456–475, 2014.

[38] L.-J. Deng, W. Guo, and T.-Z. Huang, "Single-image super-resolution via an iterative reproducing kernel Hilbert space method," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 26, no. 11, pp. 2001–2014, Nov. 2016.

[39] W. Dong *et al.*, "Hyperspectral image super-resolution via non-negative structured sparse representation," *IEEE Trans. Image Process.*, vol. 25, no. 5, pp. 2337–2352, May 2016.

[40] Y. Chen, N. M. Nasrabadi, and T. D. Tran, "Hyperspectral image classification using dictionary-based sparse representation," *IEEE Trans. Geosci. Remote Sens.*, vol. 49, no. 10, pp. 3973–3985, Oct. 2011.

[41] W. Deng, W. Yin, and Y. Zhang, "Group sparse optimization by alternating direction method," *Proc. SPIE*, vol. 8858, Sep. 2013, Art. no. 88580R.

[42] H. Zhang, J. Li, Y. Huang, and L. Zhang, "A nonlocal weighted joint sparse representation classification method for hyperspectral imagery," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 7, no. 6, pp. 2056–2065, Jun. 2014.

[43] L. Liu, L. Chen, C. L. P. Chen, Y. Y. Tang, and C. M. Pun, "Weighted joint sparse representation for removing mixed noise in image," *IEEE Trans. Cybern.*, vol. 47, no. 3, pp. 600–611, Mar. 2017, doi: 10.1109/TCYB.2016.2521428.

[44] Y. Chen, T.-Z. Huang, X.-L. Zhao, L.-J. Deng, and J. Huang, "Stripe noise removal of remote sensing images by total variation regularization and group sparsity constraint," *Remote Sens.*, vol. 9, no. 6, p. 559, Jun. 2017, doi: 10.3390/rs9060559.

[45] Y. Chen, T.-Z. Huang, L.-J. Deng, X.-L. Zhao, and M. Wang, "Group sparsity based regularization model for remote sensing image stripe noise removal," *Neurocomputing*, vol. 267, pp. 95–106, Dec. 2017.

[46] J. A. Tropp, A. C. Gilbert, and M. J. Strauss, "Algorithms for simultaneous sparse approximation. Part I: Greedy pursuit," *Signal Process.*, vol. 86, no. 3, pp. 572–588, 2006.

[47] Y. Hu, C. Li, K. Meng, J. Qin, and X. Yang, "Group sparse optimization via $\ell_{p,q}$ regularization," *J. Mach. Learn. Res.*, vol. 18, no. 30, pp. 1–52, 2017.

[48] 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, Sep. 2016.

[49] R. G. Baraniuk, V. Cevher, M. F. Duarte, and C. Hegde, "Model-based compressive sensing," *IEEE Trans. Inf. Theory*, vol. 56, no. 4, pp. 1982–2001, Apr. 2010.

[50] X. Tang, G. Feng, and J. Cai, "Weighted group sparse representation for undersampled face recognition," *Neurocomputing*, vol. 145, pp. 402–415, Dec. 2014.

[51] Y. C. Eldar and H. Bölcskei, "Block-sparsity: Coherence and efficient recovery," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, Taipei, Taiwan, Apr. 2009, pp. 2885–2888.

[52] Y. C. Eldar, P. Kuppinger, and H. Bolcskei, "Block-sparse signals: Uncertainty relations and efficient recovery," *IEEE Trans. Signal Process.*, vol. 58, no. 6, pp. 3042–3054, Jun. 2010.

[53] X.-T. Yuan and S. Yan, "Visual classification with multi-task joint sparse representation," in *Proc. CVPR*, Jun. 2010, pp. 3493–3500.

[54] M. Yuan and Y. Lin, "Model selection and estimation in regression with grouped variables," *J. Roy. Statist. Soc., B, Statist. Methodol.*, vol. 68, no. 1, pp. 49–67, 2006.

[55] F. R. Bach, "Consistency of the group lasso and multiple kernel learning," *J. Mach. Learn. Res.*, vol. 9, pp. 1179–1225, Jun. 2008.

[56] J. Friedman, T. Hastie, and R. Tibshirani. (2010). "A note on the group lasso and a sparse group lasso." [Online]. Available: https://arxiv.org/abs/1001.0736

[57] M.-D. Iordache, J. Bioucas-Dias, and A. Plaza, "Hyperspectral unmixingwith sparse group lasso," in *Proc. IEEE IGARSS*, Vancouver, BC, Canada, Jul. 2011, pp. 3586–3589.

[58] X. Zhu, X. Li, and S. Zhang, "Block-row sparse multiview multilabel learning for image Classification," *IEEE Trans. Cybern.*, vol. 46, no. 2, pp. 450–461, Feb. 2016.

[59] C. Lu, J. Tang, S. Yan, and Z. Lin, "Generalized nonconvex nonsmooth low-rank minimization," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2014, pp. 4130–4137.

[60] S. Gu, L. Zhang, W. Zuo, and X. Feng, "Weighted nuclear norm minimization with application to image denoising," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2014, pp. 2862–2869.

[61] S. Gu, Q. Xie, D. Meng, W. Zuo, X. Feng, and L. Zhang, "Weighted nuclear norm minimization and its applications to low level vision," *Int. J. Comput. Vis.*, vol. 121, no. 2, pp. 183–208, Jan. 2017.

[62] D. Gabay and B. Mercier, "A dual algorithm for the solution of nonlinear variational problems via finite element approximation," *Comput. Optim. Appl.*, vol. 2, no. 1, pp. 17–40, 1976.

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

[64] C. Wu and X.-C. Tai, "Augmented Lagrangian method, dual methods, and split Bregman iteration for ROF, vectorial TV, and high order models," *SIAM J. Imag. Sci.*, vol. 3, no. 3, pp. 300–339, Jan. 2010.

[65] J. Eckstein and D. P. Bertsekas, "On the Douglas–Rachford splitting method and the proximal point algorithm for maximal monotone operators," *Math. Program.*, vol. 55, no. 1, pp. 293–318, Apr. 1992.

[66] E. J. Candès, M. B. Wakin, and S. P. Boyd, "Enhancing sparsity by reweighted ℓ_1 minimization," *J. Fourier Anal. Appl.*, vol. 14, nos. 5–6, pp. 877–905, 2008.

[67] S. Zhang, J. Li, Z. Wu, and A. Plaza, "Spatial discontinuity-weighted sparse unmixing of hyperspectral images," *IEEE Trans. Geosci. Remote Sens.*, vol. 56, no. 10, pp. 5767–5779, Oct. 2018, doi: 10.1109/TGRS.2018.2825457.

[68] F. Li, M. Ng, and R. Plemmons, "Coupled segmentation and denoising/deblurring models for hyperspectral material identification," *Numer. Linear Algebra Appl.*, vol. 19, no. 1, pp. 153–173, Jan. 2012.

[69] R. Clark *et al.*, "Imaging spectroscopy: Earth and planetary remote sensing with the USGS Tetracorder and expert systems," *J. Geophys. Res.*, vol. 108, no. E12, pp. 5131–5135, Dec. 2003.

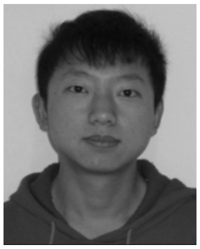
Jie Huang received the Ph.D. degree from the University of Electronic Science and Technology of China (UESTC), Chengdu, China, in 2013.

She is currently an Associate Professor with the School of Mathematical Sciences, UESTC. Her research interests include remote sensing and image processing.

Ting-Zhu Huang received the B.S., M.S., and Ph.D. degrees in computational mathematics from the Department of Mathematics, Xi'an Jiaotong University, Xi'an, China.

He is currently a Professor with the School of Mathematical Sciences, University of Electronic Science and Technology of China, Chengdu, China. His research interests include scientific computation and applications, numerical algorithms for image processing, numerical linear algebra, preconditioning technologies, and matrix analysis with applications.

Dr. Huang is an Editor of *The Scientific World Journal*, *Advances in Numerical Analysis*, the *Journal of Applied Mathematics*, the *Journal of Pure and Applied Mathematics: Advances in Applied Mathematics*, and the *Journal of Electronic Science and Technology*, China.



Liang-Jian Deng received the B.S. and Ph.D. degrees from the School of Mathematical Sciences, University of Electronic Science and Technology of China (UESTC), Chengdu, China, in 2010 and 2016, respectively.

He is currently an Associate Professor with the School of Mathematical Sciences, UESTC. His research interests include image processing and computer vision.

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 interests include the mathematical models and algorithms in high-dimensional image processing.