Hyperspectral Sparse Unmixing based on Spectral Library Pruning and Adaptive Total Variation Constraint

Chenguang Xu, Yande Liu, Zhixuan Li, Yinglong Peng, Zhaoming Wu, Chengzhi Deng

***Abstract*—Hyperspectral sparse unmixing is a method for unmixing hyperspectral data by utilizing a vast spectral library as prior information. However, the large size of spectral libraries in sparse unmixing leads to several challenges, including high computational costs and unmixing errors caused by hyperspectral noise, which hinder fast and accurate data unmixing. To address these issues, this paper proposes a sparse unmixing method based on spectral library pruning and adaptive total variation constraint (PSU-ATV). The method employs an interactive learning strategy between the abundance matrix and the spectral library to perform multiple rounds of spectral library pruning, retaining spectral signatures with higher likelihood. This approach significantly reduces computational complexity and improves unmixing speed. Additionally, to enhance robustness and mitigate the impact of hyperspectral noise on spectral pruning, an ATV-based denoising term is introduced to improve unmixing accuracy. Simulation and real-data experiments demonstrate that the proposed method achieves superior unmixing accuracy and computational efficiency compared to several state-of-the-art methods.**

***Index Terms*—spectral library pruning, adaptive total variation, sparse unmixing, hyperspectral images**

I. Introduction

In hyperspectral images, mixed pixels—where materials with distinct spectral characteristics coexist within a single pixel — result from three primary factors: the spatial distribution of ground objects, atmospheric mixing effects during signal propagation, and the intrinsic limitations of hyperspectral sensors [1]. These mixed pixels substantially interfere with subsequent hyperspectral data processing [2]. To decompose mixed pixels into their constituent pure materials (endmembers) and corresponding proportions (abundances) [39], researchers [3-5] have developed numerous unmixing approaches. Notably, sparse unmixing methods [6] demonstrate superior performance by leveraging large prior spectral libraries.

Hyperspectral sparse unmixing leverages a large spectral library as prior knowledge and incorporates various constraint terms [7]. By applying sparse regression [8] to the abundance matrix, it estimates endmember information and the corresponding fractional abundances. As a classical sparse unmixing method, SUnSAL [9] has significantly advanced subsequent research. It utilizes 240 randomly selected endmembers from the U.S. Geological Survey (USGS) [10] spectral library as the dataset, solving the unmixing problem through sparsity and non-negativity constraints. Since SUnSAL relies solely on the L1-norm for unmixing, its performance is suboptimal. Consequently, many researchers [11-14] have proposed improvements, introducing various sparse unmixing methods based on L1/2 [12], L2 [13], and Lp [14] norms. Considering the spatial characteristics of hyperspectral data, Iordache [15] enhanced the SUnSAL method and proposed SUnSAL-TV, which is a sparse unmixing method incorporating total variation (TV) spatial regularization. Subsequently, researchers have proposed TV-regularized sparse unmixing methods [16-18] based on L1/2, L2, and Lp norms. In addition to regularization terms, many scholars [19-21] have incorporated weighting factors into the sparse terms (e.g., spectral weighting [19], spatial weighting [20], and spatial-spectral weighting [21]) to further improve unmixing accuracy. Xu et al. [22, 38] proposed the SU-ATV model, which enhances the TV term to adaptively adjust the ratio of horizontal and vertical differences in spatial total variation. This was based on the observation that different regions of the abundance matrix exhibit distinct texture characteristics [23]. The approach achieved better denoising and improved unmixing accuracy. Extending this work, Ma et al. [24] introduced a spectral weighting factor in the sparse term, resulting in the WSU-ATV model, which further improved accuracy. Xu et al. [40] combined low-rank characteristics with an adaptive total variation term to propose the ATVLRSU model. Shen et al. [25] proposed a local and global sparse unmixing method (LGSU), incorporating a superpixel-based local sparse regularization to improve performance. Furthermore, Shen et al. [26] proposed the layered sparse regression unmixing (LSU) method, which

Manuscript received May 19, 2025; revised July 26, 2025.

This work was supported in part by Jiangxi Provincial Postgraduate Innovation Special Funds Project (YC2024-B200).

C. G. Xu is a PhD candidate of Intelligent Electromechanical Equipment Innovation Research Institute, East China Jiaotong University, Nanchang 330013, Jiangxi, China (e-mail: xcg@nit.edu.cn).

Y. D. Liu is a professor of Intelligent Electromechanical Equipment Innovation Research Institute, East China Jiaotong University, Nanchang 330013, Jiangxi, China (corresponding author, e-mail: jxliuyd@163.com).

Z. X. Li is an undergraduate student of School of Information Engineering, Nanchang Institute of Technology, Nanchang 330099, Jiangxi, China (e-mail: 1470548161@qq.com).

Y. L. Peng is an undergraduate student of School of Information Engineering, Nanchang Institute of Technology, Nanchang 330099, Jiangxi, China (e-mail: 3069335114@qq.com).

Z. M. Wu is an associate professor of School of Information Engineering, Nanchang Institute of Technology, Nanchang 330099, Jiangxi, China (e-mail: zmwunit@foxmail.com).

D. Z. Deng is a professor of School of Information Engineering, Nanchang Institute of Technology, Nanchang 330099, Jiangxi, China (corresponding author, e-mail: dengcz@nit.edu.cn).

decomposes the sparse regression process into multiple layers. This refines the spectral library atoms while promoting sparse regression of the abundance matrix.

The aforementioned methods have successfully applied sparse unmixing techniques to hyperspectral data, achieving effective results. However, these approaches still face several limitations:

1) High computational cost: Sparse unmixing requires selecting a small subset of materials from spectral libraries containing hundreds or thousands of entries. Although additional constraints can improve performance, they inevitably increase computational complexity.
2) Low robustness under noise: In low signal-to-noise ratio (SNR) scenarios, hyperspectral unmixing becomes highly sensitive to noise, degrading algorithm performance. The high coherence of spectra in large libraries further exacerbates this issue, leading to inaccurate abundance estimation.
3) Lack of mutual learning capability: Conventional sparse unmixing methods rely on static spectral libraries and focus solely on optimizing the abundance matrix through constraints, neglecting the potential to refine the spectral library based on abundance matrix feedback.

To address these challenges, we propose a hyperspectral sparse unmixing method combining spectral library pruning and adaptive total variation (ATV) regularization. Our key contributions include:

1) Spectral library pruning for efficiency: By analyzing the abundance matrix through deep learning, we identify significant abundance rows and iteratively prune the spectral library. This reduces its size while improving accuracy, significantly accelerating the unmixing process.
2) Denoising for reliable pruning: The abundance matrix's reliability directly affects spectral library pruning accuracy. However, under low SNR conditions, noise corrupts the matrix. To mitigate this, we introduce two enhancements: First, neighborhood-weighted averaging is applied to the abundance matrix prior to learning to improve denoising. Second, a novel ATV regularization term adaptively adjusts horizontal and vertical TV weights based on local texture, further enhancing denoising further. Together, these steps ensure robust spectral library pruning.
3) Efficient optimization via ADMM: We solve the proposed model using the alternating direction method of multipliers (ADMM), ensuring computational efficiency.

II. Related work

Hyperspectral sparse unmixing refers to the problem of solving the abundance by sparse regression, using a rich spectral library as prior knowledge. The model is as follows:

$$\mathbf{Y} = \mathbf{AX} + \Psi, \text{ s. t. } \mathbf{X} \geq 0, \tag{1}$$

where $\mathbf{Y} \in \Re^{M \times N}$ denotes hyperspectral data, $\mathbf{A} \in \Re^{M \times L}$ denotes spectral library, $\mathbf{X} \in \Re^{L \times N}$ denotes abundance matrix, and $\Psi \in \Re^{M \times N}$ denotes noise. M denotes the number of bands, N denotes the number of pixels, and L denotes the number of pure material spectra included in the spectral library. $\mathbf{X} \geq 0$ denotes a non-negativity constraint on the abundance matrix $\mathbf{X}$.

Due to the presence of dozens or even hundreds of pure material spectra in the spectral library $\mathbf{A}$, while the actual real endmembers are often only a few, the resulting $\mathbf{X}$ becomes very sparse. Therefore, the sparsity of $\mathbf{X}$ can be used as a constraint for the unmixing process. The sparse characteristics of $\mathbf{X}$ can be represented by the expression of $\|\mathbf{X}\|_0$ [27], and the unmixing model can be expressed as follows:

$$\min_{\mathbf{X}} \frac{1}{2}\|\mathbf{Y} - \mathbf{AX}\|_F^2 + \lambda \|\mathbf{X}\|_0 \text{, s. t. } \mathbf{X} \geq 0, \tag{2}$$

where λ is the regularization parameter of the sparse term $\|\mathbf{X}\|_0$.

However, the solution of the model (2) is challenging. SUnSAL uses the $\|\mathbf{X}\|_1$ approximation to represent $\|\mathbf{X}\|_0$, which is modelled as follows:

$$\min_{\mathbf{X}} \frac{1}{2}\|\mathbf{Y} - \mathbf{AX}\|_F^2 + \lambda \|\mathbf{X}\|_1 \text{, s. t. } \mathbf{X} \geq 0. \tag{3}$$

Subsequently, Iordache improved SUnSAL by proposing the SUnSAL-TV [15], which enhances the unmixing accuracy by incorporating a TV regularization term to the abundance matrix. The model is as follows:

$$\min_{\mathbf{X}} \frac{1}{2}\|\mathbf{Y} - \mathbf{AX}\|_F^2 + \lambda \|\mathbf{X}\|_{1,1} + \lambda_{TV} TV(\mathbf{X}) \text{, s. t. } \mathbf{X} \geq 0, \tag{4}$$

where λ_{TV} is the regularisation parameter of $TV(\mathbf{X})$, $TV(\mathbf{X}) = \|\nabla \mathbf{X}\|_{g,1} = \left\| \begin{bmatrix} \nabla_1 \mathbf{X} \\ \nabla_2 \mathbf{X} \end{bmatrix} \right\|_{g,1}$, $\nabla_1 \mathbf{X}$ and $\nabla_2 \mathbf{X}$ are the horizontal and vertical differences of the abundance matrix, respectively. The number of paradigms g is usually defined as 1 or 2.

III. Proposed Method

A. Spectral Library Pruning

Sparse unmixing algorithms can significantly improve unmixing accuracy [28]. However, reliance on a large spectral library as a prior leads to huge amounts of data to be processed, resulting in excessively high computational costs. Therefore, this section focuses on spectral library streamlineing to improve computational efficiency while maintaining algorithm correctness and enhancing accuracy.

During sparse unmixing, each spectrum in the spectral library corresponds to a unique row in the abundance matrix. Through algorithm iterations, constraint application, and decorrelation, the algorithm ultimately generates a sparse abundance matrix. Rows with significant non-zero values in the abundance matrix correspond to actual endmembers in the spectral library, whereas near-zero rows represent noise or redundant spectra. Typically, the number of redundant spectra is tens or even hundreds of times that of actual endmembers [29], and these excess spectra significantly increase computational costs. To reduce redundant spectra in the library, we divide the unmixing process into multiple stages and iteratively prune the library, retaining only the most probable spectra. This reduces the library's size and improves unmixing efficiency.

Before pruning the spectral library, it is necessary to assess the sparsity of each row in the abundance matrix. The row sparsity of the abundance matrix $\mathbf{X}$ can be measured by the sum of the absolute values of each row's elements [30], with smaller sums indicating greater sparsity. The row sparsity of the abundance matrix can be obtained using the following formula:

$$\mathbf{M}_t = \sqrt{\mathbf{X}_t \otimes \mathbf{X}_t}\mathbf{I}_N, \tag{5}$$

where $\mathbf{X}_t \in \Re^{p \times N}$ denotes the abundance matrix after the *t-th* spectral pruning, and the pruned spectral library contains *p* spectra. $\mathbf{M}_t \in \Re^{p \times 1}$ is a column vector of *p* rows, and each member in the vector is used to save the sparsity of each row in the $\mathbf{X}_t$. $\mathbf{I}_N \in \Re^{N \times 1}$ is a unit vector of *N* rows.

In order to filter out the more likely substance spectra, we sort the weight values in $\mathbf{M}_t$, eliminate the corresponding rows of the abundance matrix and the corresponding substance spectra in the spectral library with very small or zero weights, and keep only the ones with larger weights as candidate endmembers, thus effectively pruning the spectral library. The above pruning process performs well under high SNR conditions. However, when dealing with hyperspectral data that has low SNR, spectral pruning becomes much more challenging. In the unmixing process of high-noise hyperspectral data, the abundance matrix is often affected by substantial noise interference, and the accuracy of the abundance matrix in the unmixing plays a crucial role in the success of spectral library pruning. Excessive noise interference can lead to incorrect pruning, which may have serious consequences for subsequent unmixing processes. To address these issues, this paper applies a neighborhood-weighted averaging method [31] for denoising the abundance matrix before assessing its sparsity, ensuring the accuracy of the sparse vector. In this paper, a new abundance reference matrix $\mathbf{E}_t \in \Re^{p \times N}$ after applying the neighborhood-weighted averaging to $\mathbf{X}_t$ is added to replace the original abundance matrix in order to take a more accurate abundance sparsity. The size of this matrix remains the same as $\mathbf{X}_t$. It is assumed that the value of an element in $\mathbf{E}_t$ is jointly determined by the value of an element in $\mathbf{X}_t$ and its neighbourhood weight. The neighbourhood weights are defined as follows:

$$D_{ij} = 1 \Big/ \sqrt{(a-b)^2 + (c-d)^2}, \tag{6}$$

D_{ij} denotes the neighbourhood distance weight between matrix centroid *i* (*a*, *c*) and neighbourhood point *j* (*b*, *d*) denotes the neighbourhood weight of matrix centroid *i* with coordinates (*a*, *c*) and neighbourhood point j with coordinates (*b*, *d*). The larger the difference between their coordinates, the smaller the value of the neighbourhood weight, and the weaker the effect on the centroid.

$$f_{h\in\Pi(ij)}(x_{ij}) = \sum\nolimits_{h\in\Pi(ij)} D_h x_h \Big/ \sum\nolimits_{h\in\Pi(ij)} D_h, \tag{7}$$

where $\Pi(ij)$ denotes the set of neighbours of x_{ij}. $f_{h\in\Pi(ij)}(x_{ij})$ denotes the weighted average of the centroid $x(i,j)$ in its neighbourhood $h \in \Pi(ij)$. Thus, the abundance reference matrix after the *t-th* pruning is obtained as:

$$\mathbf{E}_t = \begin{bmatrix} e_{11,t} & \cdots & e_{1n,t} \\ \vdots & e_{ij,t} & \vdots \\ e_{m1,t} & \cdots & e_{mn,t} \end{bmatrix} = \begin{bmatrix} f_{h\in\Pi(ij)}(x_{11,t}) & \cdots & f_{h\in\Pi(ij)}(x_{1n,t}) \\ \vdots & f_{h\in\Pi(ij)}(x_{ij,t}) & \vdots \\ f_{h\in\Pi(ij)}(x_{m1,t}) & \cdots & f_{h\in\Pi(ij)}(x_{mn,t}) \end{bmatrix}. \tag{8}$$

Then use Eq. (5) to compute the row sparsity vectors of $\mathbf{E}_t$

$$\mathbf{M}_t = \sqrt{\mathbf{E}_t \otimes \mathbf{E}_t}\mathbf{I}_N. \tag{9}$$

The sparse vectors obtained after denoising using Eq. (9) are more accurate. Each value in $\mathbf{M}_t$ denotes the sparse weight of the row corresponding to it in $\mathbf{X}_t$ and the spectral weight of the column corresponding to it in $\mathbf{A}_t$. By sorting the values in $\mathbf{M}_t$, the abundance and spectral libraries were screened down. Spectra with small or zero weights can be eliminated and not subsequently retained in the spectral library.

Fig. 1 shows the unmixing results of the proposed method with and without abundance denoising at 20 dB. In the abundance map, the horizontal axis represents hyperspectral image pixels, and the vertical axis corresponds to the endmembers in the spectral library. In the endmember map, the horizontal axis denotes the endmembers in the spectral library, and the vertical axis represents the band information of the hyperspectral image. It can be observed that the unmixing result without abundance denoising contains an erroneous abundance row, and the corresponding endmember spectrum also exhibits a spurious signature (the differential information is marked with a red box in the figure). In contrast, the unmixing result with abundance denoising closely matches the true values. This indicates that the unmixing performance with abundance denoising is indeed significantly better than that without abundance denoising.

B. Hyperspectral sparse unmixing

Pruning the spectral library can effectively accelerate computational efficiency. However, in hyperspectral data with severe noise, merely applying abundance denoising is insufficient. To address this, we propose a hyperspectral sparse unmixing method based on spectral library pruning and adaptive total variation (ATV) constraints. This method jointly leverages the spatial and spectral information of hyperspectral data, simultaneously performing denoising and unmixing, thereby more effectively ensuring the pruning of the spectral library in sparse unmixing. The optimization model for each pruning step is as follows:

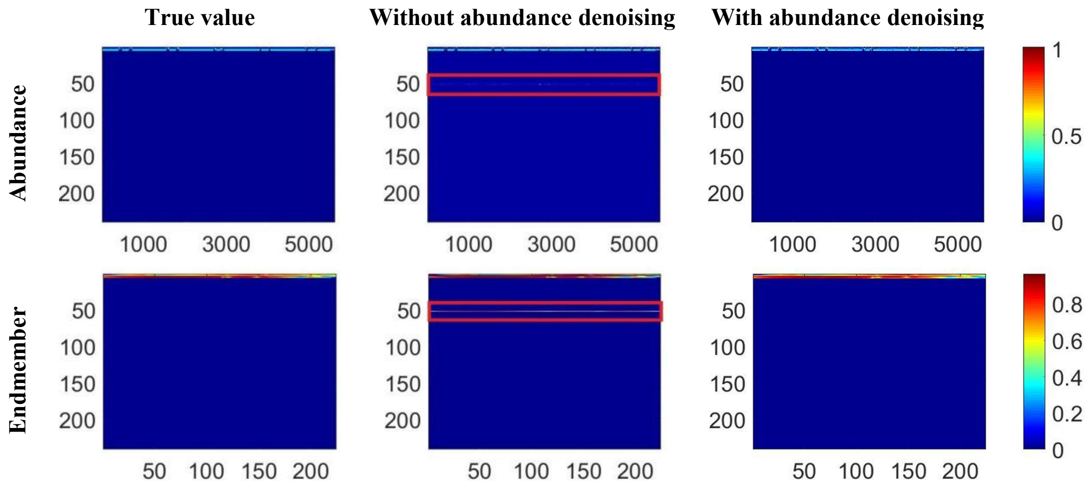


Fig. 1. Unmixing results of the proposed method with and without abundance denoising in the 20 dB.

$$\min_{\mathbf{X}_t}\frac{1}{2}\left\|\mathbf{Y}-\mathbf{A}_t\mathbf{X}_t\right\|_F^2+\lambda\left\|\mathbf{X}_t\right\|_{1,1}+\lambda_{TV}\left\|\mathbf{B}_t\left(\nabla\mathbf{X}_t\right)\right\|_{2,1},$$

$$\text{s. t. } \mathbf{X}\geq 0, \qquad (9)$$

where t denotes the number of times the spectral library is pruned. $\mathbf{A}_t\in\Re^{M\times p}$ denotes the spectral library after the *t-th* pruning. $\mathbf{X}_t\in\Re^{p\times N}$ denotes the abundance matrix after the *t-th* pruning. p is the number of pure substances of the spectral library after pruning, and its initial value is L. $\nabla\mathbf{X}_t=\left[\nabla_1\mathbf{X}_t \quad \nabla_2\mathbf{X}_t\right]^T$ is the abundance total variation term, and $\nabla_1\mathbf{X}_t$ and $\nabla_2\mathbf{X}_t$ are the horizontal and vertical differences of the abundance matrix, respectively. In Eq. (4), the total variation (TV) regularization term fixes the ratio of horizontal to vertical differences, which restricts the ability to process horizontal differences and vertical differences differently in images with different spatial structures, and thus does not give full play to the denoising efficacy of the abundance total variation. Therefore, this paper introduces an adaptive adjustment matrix $\mathbf{B}_t$, which can dynamically adjust the ratio of horizontal and vertical differences according to the varying structural characteristics of different regions, so it can denoise more effectively in the unmixing and arrive at better results. The expression for $\mathbf{B}_t$ is as follows:

$$\mathbf{B}_t=diag\begin{pmatrix}b_{1,t}\\ b_{2,t}\end{pmatrix}=diag\begin{pmatrix}1/\left(1+r\left|G_{\hat{\delta}}*\nabla_1\mathbf{X}_t\right|^2\right)\\ 1/\left(1+r\left|G_{\hat{\delta}}*\nabla_2\mathbf{X}_t\right|^2\right)\end{pmatrix}, \qquad (10)$$

where $diag(\vdots)$ denotes the diagonal matrix. r is a tuning parameter, $G_{\hat{\delta}}$ defines a Gaussian convolution with standard deviation $\hat{\delta}$ and $*$ is the convolution symbol.

C. Model Optimisation and Solution

A collation of Eq. (9) can be obtained:

$$\min_{\mathbf{X}_t}\frac{1}{2}\left\|\mathbf{Y}-\mathbf{A}_t\mathbf{X}_t\right\|_F^2+\lambda\left\|\mathbf{X}_t\right\|_{1,1}+ \\ \lambda_{TV}\left\|\mathbf{B}_t\left(\nabla\mathbf{X}_t\right)\right\|_{2,1}+i\_R_+\left(\mathbf{X}_t\right). \qquad (11)$$

The ADMM method was used to solve for Eq. (11) and obtained:

$$\text{Let } g\left(\mathbf{V}_t\right)=\frac{1}{2}\left\|\mathbf{Y}-\mathbf{A}_t\mathbf{U}_t\right\|_F^2+\lambda\left\|\mathbf{V}_{1,t}\right\|_{1,1}+ \\ \lambda_{TV}\left\|\mathbf{V}_{4,t}\right\|_{2,1}+i\_R_+\left(\mathbf{V}_{5,t}\right), \qquad (12)$$

where $\mathbf{U}_t=\mathbf{X}_t$, $\mathbf{V}_t=[\mathbf{V}_{1,t},\mathbf{V}_{2,t},\mathbf{V}_{3,t},\mathbf{V}_{4,t},\mathbf{V}_{5,t}]$, $\mathbf{V}_{1,t}=\mathbf{U}_t$, $\mathbf{V}_{2,t}=\mathbf{U}_t$, $\mathbf{V}_{3,t}=\nabla\mathbf{V}_{2,t}$, $\mathbf{V}_{4,t}=\mathbf{B}_t\mathbf{V}_{3,t}$, $\mathbf{V}_{5,t}=\mathbf{U}_t$.

Simplify to get the following equation:

$$\min_{\mathbf{U}_t,\mathbf{V}_t} g\left(\mathbf{V}_t\right),\text{ s. t. }\mathbf{R}\mathbf{U}_t=\mathbf{Z}_t\mathbf{V}_t, \qquad (13)$$

$$\text{where } \mathbf{R}=\begin{bmatrix}\mathbf{I}\\ \mathbf{I}\\ \mathbf{0}\\ \mathbf{0}\\ \mathbf{I}\end{bmatrix}\ \mathbf{Z}_t=\begin{bmatrix}\mathbf{I}&\mathbf{0}&\mathbf{0}&\mathbf{0}&\mathbf{0}\\ \mathbf{0}&\mathbf{I}&\mathbf{0}&\mathbf{0}&\mathbf{0}\\ \mathbf{0}&-\nabla&\mathbf{I}&\mathbf{0}&\mathbf{0}\\ \mathbf{0}&\mathbf{0}&-\mathbf{B}_t&\mathbf{I}&\mathbf{0}\\ \mathbf{0}&\mathbf{0}&\mathbf{0}&\mathbf{0}&\mathbf{I}\end{bmatrix}.$$

Solving Eq. (13) by the Lagrange multiplier method can be obtained:

$$\min_{\mathbf{U}_t,\mathbf{V}_t,\mathbf{D}_t}\zeta\left(\mathbf{U}_t,\mathbf{V}_t,\mathbf{D}_t\right)=\min_{\mathbf{U}_t,\mathbf{V}_t,\mathbf{D}_t} g\left(\mathbf{V}_t\right)+ \\ \frac{\mu}{2}\left\|\mathbf{R}\mathbf{U}_t-\mathbf{Z}_t\mathbf{V}_t-\mathbf{D}_t\right\|_F^2, \qquad (14)$$

where $\mu>0$ denotes the regularisation parameter and $\mathbf{D}_t=\left[\mathbf{D}_{1,t},\mathbf{D}_{2,t},\mathbf{D}_{3,t},\mathbf{D}_{4,t},\mathbf{D}_{5,t}\right]$ denotes the estimated deviation between the Lagrange multiplier and the correct value.

Solve for $\mathbf{U}_t$:

$$\mathbf{U}_t^{k+1}\leftarrow\arg\min_{\mathbf{U}_t}\left(\frac{1}{2}\left\|\mathbf{Y}-\mathbf{A}_t\mathbf{U}_t\right\|_F^2+\right. \\ \frac{\mu}{2}\left(\left\|\mathbf{U}_t-\mathbf{V}_{1,t}^k-\mathbf{D}_{1,t}^k\right\|_F^2+\left\|\mathbf{U}_t-\mathbf{V}_{2,t}^k-\mathbf{D}_{2,t}^k\right\|_F^2\right. \\ \left.\left.+\left\|\mathbf{U}_t-\mathbf{V}_{5,t}^k-\mathbf{D}_{5,t}^k\right\|_F^2\right)\right). \qquad (15)$$

can be obtained by solving:

$$\mathbf{U}_t^{k+1}=\left(\mathbf{A}_t^{\mathrm{T}}\mathbf{A}_t+3\mu\mathbf{I}\right)^{-1}\Big[\mathbf{A}_t^{\mathrm{T}}\mathbf{Y}+\mu\Big((\mathbf{V}_{1,t}^{k}+\mathbf{D}_{1,t}^{k}) \\ +(\mathbf{V}_{2,t}^{k}+\mathbf{D}_{2,t}^{k})+(\mathbf{V}_{5,t}^{k}+\mathbf{D}_{5,t}^{k})\Big)\Big]. \tag{16}$$

Solve for $\mathbf{V}_{1,t}$:

$$\mathbf{V}_{1,t}^{k+1}\leftarrow\arg\min_{\mathbf{V}_{1,t}}\left(\lambda\left\|\mathbf{V}_{1,t}\right\|_{1,1}+\frac{\mu}{2}\left\|\mathbf{U}_t^{k+1}-\mathbf{V}_{1,t}-\mathbf{D}_{1,t}^{k}\right\|_F^2\right). \tag{17}$$

The solution is obtained:

$$\mathbf{V}_{1,t}^{k+1}=\operatorname{soft}\left(\mathbf{U}_t^{k+1}-\mathbf{D}_{1,\mathrm{t}}^{k},\frac{\lambda}{\mu}\right). \tag{18}$$

Solve for $\mathbf{V}_{2,t}$:

$$\mathbf{V}_{2,t}^{k+1}\leftarrow\arg\min_{\mathbf{V}_{2,t}}\begin{pmatrix}\left\|\mathbf{U}_t^{k+1}-\mathbf{V}_{2,t}-\mathrm{D}_{2,t}^{k}\right\|_F^2\\ +\left\|\nabla\mathbf{V}_{2,t}-\mathbf{V}_{3,t}^{k}-\mathbf{D}_{3,t}^{k}\right\|_F^2\end{pmatrix}. \tag{19}$$

The solution can be obtained:

$$\mathbf{V}_{2,t}^{k+1}=F^{-1}\left[\frac{F\left(\mathbf{U}_t^{k+1}-\mathbf{D}_{2,t}^{k}-div\left(\mathbf{V}_{3,t}^{k}+\mathbf{D}_{3,t}^{k}\right)\right)}{F\left(\mathbf{I}-div\nabla\right)}\right]. \tag{20}$$

Solve for $\mathbf{V}_{3,t}$:

$$\mathbf{V}_{3,t}^{k+1}\leftarrow\arg\min_{\mathbf{V}_{3,t}}\begin{pmatrix}\left\|\nabla\mathbf{V}_{2,t}^{k+1}-\mathbf{V}_{3,t}-\mathrm{D}_{3,t}^{k}\right\|_F^2+\\ \left\|\mathbf{B}_t\mathbf{V}_{3,t}-\mathbf{V}_{4,t}^{k}-\mathbf{D}_{4,t}^{k}\right\|_F^2\end{pmatrix}. \tag{21}$$

The solution can be obtained:

$$\mathbf{V}_{3,t}^{k+1}=\left[\mathbf{V}_{31,t}^{k+1}\quad\mathbf{V}_{32,t}^{k+1}\right]^{\mathrm{T}}=\Big[\Big(\nabla_1\mathbf{V}_{2,t}^{k+1}-\mathbf{D}_{31,t}^{k}+ \\ b_{1,t}\left(\mathbf{V}_{41,t}^{k}+\mathbf{D}_{41,t}^{k}\right)\Big)/\left(b_{1,t}^2+1\right)\quad\Big(\nabla_2\mathbf{V}_{2,t}^{k+1}-\mathbf{D}_{32,t}^{k} \\ +b_{2,t}\left(\mathbf{V}_{42,t}^{k}+\mathbf{D}_{42,t}^{k}\right)\Big)/\left(b_{2,t}^2+1\right)\Big]^{\mathrm{T}}, \tag{22}$$

where $\mathbf{D}_{3,t}^{k}=\left[\mathbf{D}_{31,t}^{k}\quad\mathbf{D}_{32,t}^{k}\right]^{\mathrm{T}}$, $\mathbf{V}_{4,t}^{k}=\left[\mathbf{V}_{41,t}^{k}\quad\mathbf{V}_{42,t}^{k}\right]^{\mathrm{T}}$, $\mathbf{D}_{4,t}^{k}=\left[\mathbf{D}_{41,t}^{k}\quad\mathbf{D}_{42,t}^{k}\right]^{\mathrm{T}}$.

Solve for $\mathbf{V}_{4,t}$:

$$\mathbf{V}_{4,t}^{k+1}\leftarrow\arg\min_{\mathbf{V}_{4,t}}\begin{pmatrix}\lambda_{TV}\left\|\mathbf{V}_{4,t}\right\|_{2,1}+\\ \frac{\mu}{2}\left\|\mathbf{B}_t\mathbf{V}_{3,t}^{k+1}-\mathbf{V}_{4,t}-\mathbf{D}_{4,t}^{k}\right\|_F^2\end{pmatrix}. \tag{23}$$

The solution is obtained:

$$\mathbf{V}_{4,t}^{k+1}=\operatorname{soft}\left(\mathbf{B}_t\mathbf{V}_{3,t}^{k+1}-\mathbf{D}_{4,t}^{k},\frac{\lambda_{TV}}{\mu}\right). \tag{24}$$

Solve for $\mathbf{V}_{5,t}$:

$$\mathbf{V}_{5,t}^{k+1}\leftarrow\arg\min_{\mathbf{V}_{5,t}}\left(i\_R_{+}\left(\mathbf{V}_{5,t}\right)+\frac{\mu}{2}\left\|\mathbf{U}_t^{k+1}-\mathbf{V}_{5,t}-\mathbf{D}_{5,t}^{k}\right\|_F^2\right. \tag{25}$$

The solution is obtained:

$$\mathbf{V}_{5,t}^{k+1}=\max\left(\mathbf{U}_t^{k+1}-\mathbf{D}_{5,t}^{k},0\right). \tag{26}$$

Solve for $\mathbf{D}_t$:

$$\begin{cases}\mathbf{D}_{1,t}^{k+1}=\mathbf{D}_{1,t}^{k}-\left(\mathbf{U}_t^{k+1}-\mathbf{V}_{1,t}^{k+1}\right),\\ \mathbf{D}_{2,t}^{k+1}=\mathbf{D}_{2,t}^{k}-\left(\mathbf{U}_t^{k+1}-\mathbf{V}_{2,t}^{k+1}\right),\\ \mathbf{D}_{3,t}^{k+1}=\mathbf{D}_{3,t}^{k}-\left(\nabla\mathbf{V}_{2,t}^{k+1}-\mathbf{V}_{3,t}^{k+1}\right),\\ \mathbf{D}_{4,t}^{k+1}=\mathbf{D}_{4,t}^{k}-\left(\mathbf{B}_t\mathbf{V}_{3,t}^{k+1}-\mathbf{V}_{4,t}^{k+1}\right),\\ \mathbf{D}_{5,t}^{k+1}=\mathbf{D}_{5,t}^{k}-\left(\mathbf{U}_t^{k+1}-\mathbf{V}_{5,t}^{k+1}\right).\end{cases} \tag{27}$$

TABLE I shows the flow of PSU-ATV , the first step requires the input of regularisation parameters and initial values of variables. t and k are the pruning count variable and the iteration count variable, both of which have an initial value of 0. t_max is the maximum number of prunings and p_min is the minimum number of pure substances retained by pruning of the spectral library, which needs to be judged at each spectral pruning. p_min is generally set to the number of true endmembers. The second step is the core of the algorithm. First of all, using the ADMM method [32] to solve the solution variable iteratively. It should be noted that in order to ensure the stability of the algorithm, we update $\mathbf{B}_t$ every i_b iterations, not every iteration. Similarly, in pruning the spectral library, we prune the spectral library once after i_a iterations instead of pruning the spectral library every iteration. This procedure sets i_b = 50 and i_a =80. $Address$ records the index of the $\mathbf{M}_t$ vector before the descending ordering, which is likewise the index of the position of each single substance in the spectral library $\mathbf{A}_t$ before the ordering, and the index of the position of the corresponding abundance of each single substance in the abundance matrix $\mathbf{U}_t^k$ before the ordering. We eliminate some of the single substances that have very small or zero sparsity after descending ordering for the purpose of spectral library construction. The pruning progress q indicates the magnitude of each spectral library pruning, the larger q is, the faster the pruning is, and $q=2$ is taken in this paper.

TABLE I
The pseudocode of PSU-ATV

```
Algorithm 1: PSU-ATV algorithm
 1. Input: $\lambda,\ \lambda_{TV},\ p\_min,\ t\_max$ .
  Initialization: set $\mathbf{U}_0^0=\left(\mathbf{A}_0^{\mathrm{T}}\mathbf{A}_0+3\mu\mathbf{I}\right)^{-1}\left(\mathbf{A}_0^{\mathrm{T}}\mathbf{Y}\right)$,
  $\mathbf{V}_{1,0}^0=\mathbf{V}_{2,0}^0=\mathbf{V}_{5,0}^0=\mathbf{U}_0^0$, $\mathbf{V}_{3,0}^0=\nabla\mathbf{V}_{2,0}^0$, $\mathbf{V}_{4,0}^0=\mathbf{B}\mathbf{V}_{3,0}^0$,
  $\mathbf{D}_0^0=\left[\mathbf{D}_{1,0}^0,\mathbf{D}_{2,0}^0,\mathbf{D}_{3,0}^0,\mathbf{D}_{4,0}^0,\mathbf{D}_{5,0}^0\right]=\mathbf{0}$,
  $t=0,k=0,\ b_{1,0}=1/\left(1+r\left|G_{\hat{\delta}}*\nabla_1\mathbf{U}_0^0\right|^2\right)$,
  $b_{2,0}=1/\left(1+r\left|G_{\hat{\delta}}*\nabla_2\mathbf{U}_0^0\right|^2\right).$,
 2. Repeat
   (1) Update variables:
        (a) $\mathbf{U}_t^{k+1} = \left(\mathbf{A}_t^{\mathrm{T}}\mathbf{A}_t + 3\mu\mathbf{I}\right)^{-1} \left[\mathbf{A}_t^{\mathrm{T}}\mathbf{Y} + \mu\left( \begin{array}{l} (\mathbf{V}_{1,t}^k + \mathbf{D}_{1,t}^k) + \\ (\mathbf{V}_{2,t}^k + \mathbf{D}_{2,t}^k) + (\mathbf{V}_{5,t}^k + \mathbf{D}_{5,t}^k) \end{array}\right)\right].$
        (b) $\mathbf{V}_{1,t}^{k+1} = \mathrm{soft}\left(\mathbf{U}_t^{k+1} - \mathbf{D}_{1,t}^k, \frac{\lambda}{\mu}\right).$
        (c) $\mathbf{V}_{2,t}^{k+1} = F^{-1}\left[\left(F\left(\mathbf{U}_t^{k+1} - \mathbf{D}_{2,t}^k - div\left(\mathbf{V}_{3,t}^k + \mathbf{D}_{3,t}^k\right)\right) \middle/ F\left(\mathbf{I} - div\nabla\right)\right].$
        (d) $\mathbf{V}_{3,t}^{k+1} = \left[\mathbf{V}_{31,t}^{k+1} \quad \mathbf{V}_{32,t}^{k+1}\right]^{\mathrm{T}} = \left[\left(\nabla_1\mathbf{V}_{2,t}^{k+1} - \mathbf{D}_{31,t}^k + b_{1,t}\left(\mathbf{V}_{41,t}^k + \mathbf{D}_{41,t}^k\right)\right)/\left(b_{1,t}^2 + 1\right), \left(\nabla_2\mathbf{V}_{2,t}^{k+1} - \mathbf{D}_{32,t}^k + b_{2,t}\left(\mathbf{V}_{42,t}^k + \mathbf{D}_{42,t}^k\right)\right)/\left(b_{2,t}^2 + 1\right)\right]^{\mathrm{T}},$
        (e) $\mathbf{V}_{4,t}^{k+1} = \mathrm{soft}\left(\mathbf{B}_t\mathbf{V}_{3,t}^{k+1} - \mathbf{D}_{4,t}^k, \frac{\lambda_{TV}}{\mu}\right).$
        (f) $\mathbf{V}_{5,t}^{k+1} = \max\left(\mathbf{U}_t^{k+1} - \mathbf{D}_{5,t}^k, 0\right).$
    (2) Update Lagrangian operator:
        (g) $\mathbf{D}_{1,t}^{k+1} = \mathbf{D}_{1,t}^k - \left(\mathbf{U}_t^{k+1} - \mathbf{V}_{1,t}^{k+1}\right).$
        (h) $\mathbf{D}_{2,t}^{k+1} = \mathbf{D}_{2,t}^k - \left(\mathbf{U}_t^{k+1} - \mathbf{V}_{2,t}^{k+1}\right).$
        (i) $\mathbf{D}_{3,t}^{k+1} = \mathbf{D}_{3,t}^k - \left(\nabla\mathbf{V}_{2,t}^{k+1} - \mathbf{V}_{3,t}^{k+1}\right).$
        (j) $\mathbf{D}_{4,t}^{k+1} = \mathbf{D}_{4,t}^k - \left(\mathbf{B}_t\mathbf{V}_{3,t}^{k+1} - \mathbf{V}_{4,t}^{k+1}\right).$
        (k) $\mathbf{D}_{5,t}^{k+1} = \mathbf{D}_{5,t}^k - \left(\mathbf{U}_t^{k+1} - \mathbf{V}_{5,t}^{k+1}\right).$
    (3) Replacement of variables.
        (l) If $k\%i_b = 0$, then
        (l1) $b_{1,t} = 1/\left(1 + r\left|G_{\hat{\delta}} * \nabla_1\mathbf{U}_t^k\right|^2\right),$
             $b_{2,t} = 1/\left(1 + r\left|G_{\hat{\delta}} * \nabla_2\mathbf{U}_t^k\right|^2\right).$
    (4) Spectral library pruning
        (m) If $k\%i_a = 0$ and $p > p\_min$ and
            $t < t\_max$ then
            (m1) $\mathbf{E}_t = \begin{bmatrix} e_{11,t} & \cdots & e_{1n,t} \\ \vdots & e_{ij,t} & \vdots \\ e_{m1,t} & \cdots & e_{mn,t} \end{bmatrix}$ where
                 $e_{ij,t} = f_{h\in\Pi(ij)}(x_{ij}) = \frac{\sum_{h\in\Pi(ij)} D_h x_h}{\sum_{h\in\Pi(ij)} D_h}.$
            (m2) $\mathbf{M}_t = \sqrt{\mathbf{E}_t \otimes \mathbf{E}_t \mathbf{I}_N}.$
            (m3) Sort the $\mathbf{M}_t$ vector in descending order and record the index before the sort $Address$ .
            (m4) If $p/q <= p\_min$
                     $p = p\_min$
                 Else If $p/q > p\_min$
                     $p = ceil(p/q)$
            (m5) Pruning Spectral Library ,
                 $\mathbf{A}_t\left(:, Address(p+1:end)\right) = [\ ].$
            (m6) Synchronised abundance matric
                 $\mathbf{U}_t^k\left(Address(p+1:end),:\right) = [\ ].$
            (m7) Updated pruning frequency $t = t+1.$
    (5) Update iterations: $k = k+1.$
3. Until a certain stopping condition is met.
4. Output: $\mathbf{A}_t$, $\mathbf{U}_t$.
```

D. Parameter Setting

The μ value is updated once after every ten iterations. When the dual residuals $\left\|\mathbf{RU}_t^{k+1} - \mathbf{Z}_t^{k+1}\mathbf{V}_t^{k+1}\right\|_2^2$ (DR) is much larger than the primal residua $\left\|\mathbf{Z}_t^{k+1}\mathbf{V}_t^{k+1} - \mathbf{Z}_t^k\mathbf{V}_t^k\right\|_2^2$ (PR) (e.g., DR > 10PR), the Lagrange multiplier penalty factor should be increased to reduce the deviation value (e.g., $\mu = 2\mu$, $\mathbf{D}_{t+1} = \mathbf{D}_t / 2$). On the contrary, when PR is much larger than PR (e.g. PR>10DR), the Lagrange multiplier penalty factor should be appropriately reduced to increase the deviation (e.g. $\mu = \mu / 2$, $\mathbf{D}_{t+1} = 2\mathbf{D}_t$).

The maximum number of prunings t_max is related to the number of pure substances in the original spectral library L and the minimum number of pure substances p_min , which is determined by the following equation:

$$t\_max \approx round\left(1 + \log_{1-1/q}\frac{p\_min}{L}\right). \quad (28)$$

E. Convergence

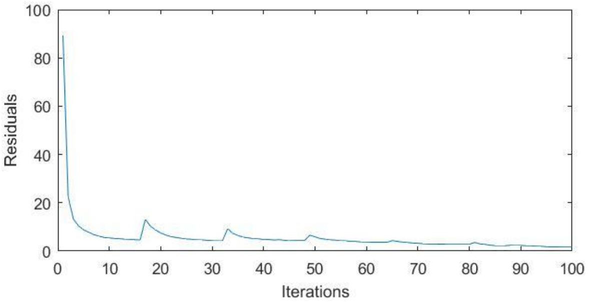


Fig. 2. Plot of iterations versus residuals for PSU_ATV

Fig. 2 is a plot of the $\left\|\mathbf{RU}_t^k - \mathbf{Z}_t^k\mathbf{V}_t^k\right\|_2^2$ as a function of residuals versus iterations for unmixing DC2 using the PSU_WATV at 20 dB. The algorithm outputs a residual value every 5 times for a total of 100 values, where the spectral library is pruned at the 16th, 32nd, 48th, and 80th. Inevitably, the residuals are disturbed at each spectral library pruning, and thus the curve jumps at these locations in the residual plots. However, as the number of iterations increases, the curve jumps are gradually weakened, and the overall trend is reduced, and finally smoothed down. It can be indirectly illustrated from the figure that the method has good convergence and stability.

IV. Experiments

In this section, we conduct two simulated data experiments and one real data experiment to evaluate the effectiveness of our proposed method. We compare our method with several currently popular methods, including SUnSAL, SUnSAL-TV, S^{2}WSU, SU-ATV, LGSU, LSU and ATVLRSU. To ensure fairness, all algorithms are implemented on a PC with an Intel Core i7-10510U CPU and 32GB of RAM, using MATLAB R2016a for computation.

A. Experiments with Simulated Data

We utilise three metrics, the Probability of Success (*ps*) [33]， the Signal Reconstruction Error (SRE) [34] and Root Mean Square Error (RMSE)[26], to evaluate the performance of the algorithms. Let $\mathbf{x}$ represent the real abundance and $\hat{\mathbf{x}}$ is the estimated abundance, the expressions for ps, SRE and RMSE are given in equations (29), (30) and (31).

$$\text{SRE(dB)} = 10 \cdot \lg\left(\text{E}\left(\|\mathbf{x}\|_2^2\right) \Big/ \text{E}\left(\|\mathbf{x}-\hat{\mathbf{x}}\|_2^2\right)\right), \quad (29)$$

where E(•) represents the expected function. The unit of SRE is dB. A larger SRE value indicates smaller errors, demonstrating better unmixing performance.

$$ps = P\left(\left(\|\mathbf{x}-\hat{\mathbf{x}}\|^2 \Big/ \|\mathbf{x}\|^2\right) \le threshold\right), \quad (30)$$

where *ps* denotes the probability that the abundance error rate is less than the *threshold*. A larger value of *ps* indicates a higher probability of correctness, resulting in a better unmixing outcome.

$$\text{RMSE} = \sqrt{\|\mathbf{x}-\hat{\mathbf{x}}\|_2^2 \Big/ (L \times N)}, \quad (31)$$

where RMSE defines the error between the measured value and the true value. The smaller the RMSE value is, the smaller the error is, indicating that the unmixing results are obtained with smaller error and better effect.

In the simulation data experiments, we use the dataset $\mathbf{A} \in \Re^{224*240}$ for our experiments. This dataset consists of 240 randomly selected spectral signatures of different mineral types from the United States Geological Survey (USGS) spectral library [35]. Each spectral signature in the dataset contains 224 bands ranging from 0.4μm to 2.5μm. We conduct two simulation data experiments to evaluate the effectiveness of the proposed method.

1) DC1 is the first hyperspectral dataset, constructed through a linear combination of five different mineral spectra randomly selected from the dataset $\mathbf{A}$ and their corresponding varying abundances.The abundance matrix of the five minerals consists of non-negative elements arranged in a 75×75 pixel format. The abundance map is shown in Fig. 3. By adding Gaussian white noise with SNR of 10 dB, 20 dB, and 30 dB to the noise-free DC1, three different noisy versions of DC1 data are created, and used to simulate the hyperspectral data in the noisy case.
2) DC2 is the second hyperspectral dataset, created from a real scene simulation. It is formed by a linear combination of nine randomly selected spectral signatures of different minerals from the dataset $\mathbf{A}$ and their corresponding abundances. The mineral abundance matrix consists of non-negative elements arranged in a 100×100 pixel format. The abundance map is shown in Fig. 4. Similar to DC1, we create three noisy simulated datasets with SNR of 10 dB, 20 dB, and 30 dB.

(1) Experimental Programme

The PSU-ATV utilizes a spectral library $\mathbf{A}$ containing spectra of 240 different mineral types for unmixing. The number of endmembers in DC1 and DC2 is 5 and 9, respectively. According to equation (28), the maximum pruning iterations for DC1 and DC2 are estimated to be 6 and 5. In order to examine the unmixing effect of spectral pruning numbers, we pruned and unmixed DC1 and DC2 using different pruning numbers and analysed the PSU-ATV by observing the algorithm's unmixing time, SRE, *ps* and RMSE values. To ensure fairness, we set the number of iterations for each pruning to 50, and the number of iterations for unmixing after the final pruning is uniformly set to 200, and the specific settings and unmixing results are shown in TABLE II.

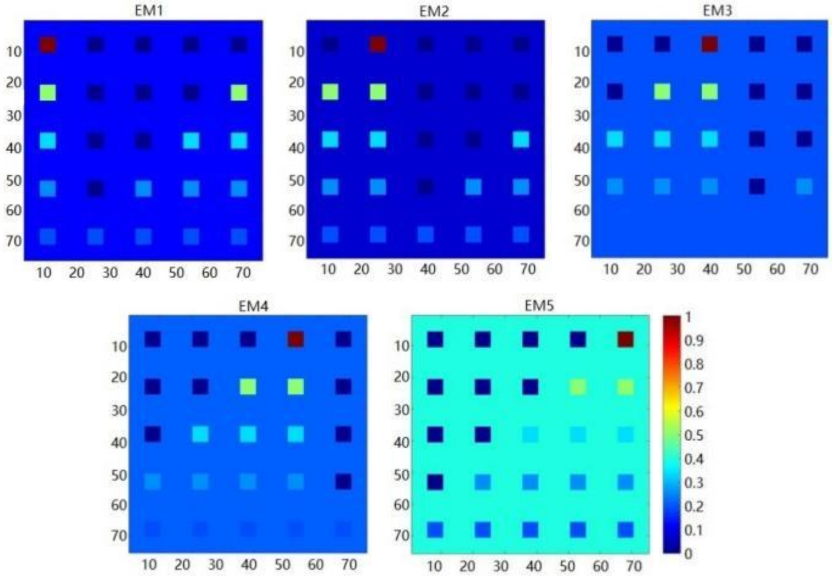


Fig. 3. True abundance map of DC1

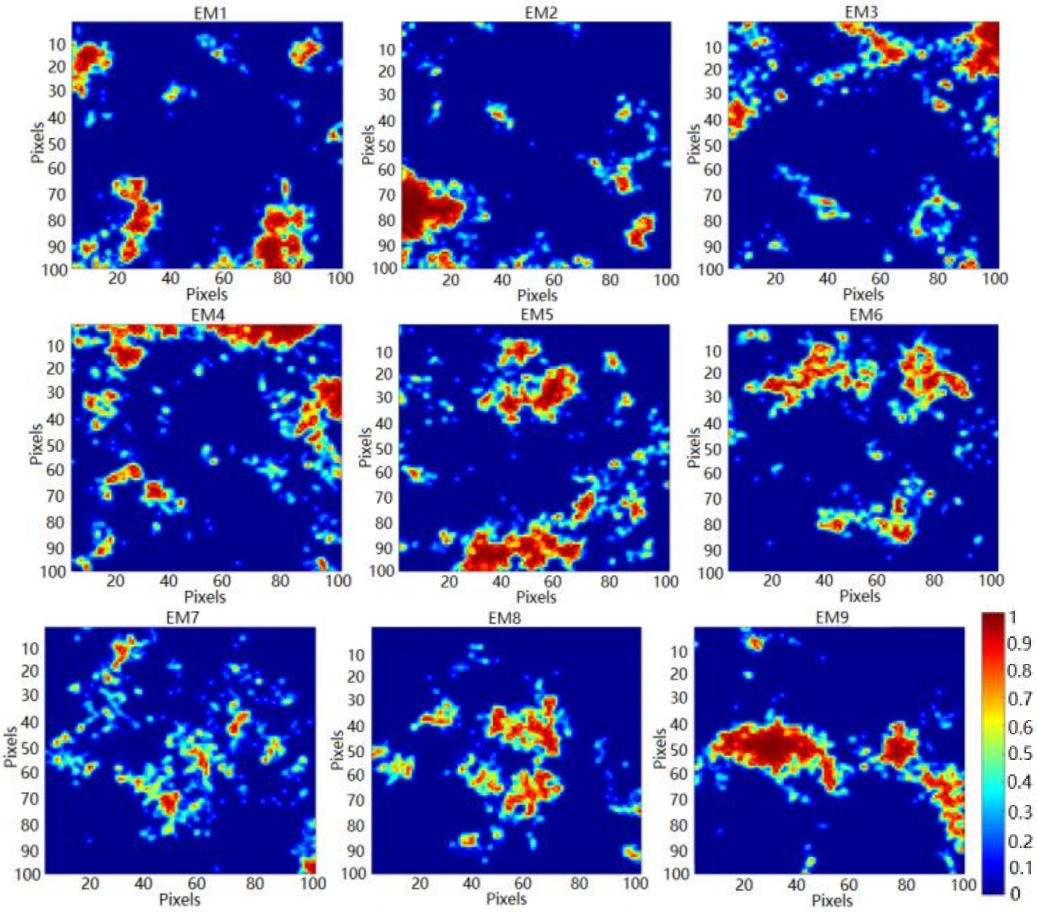
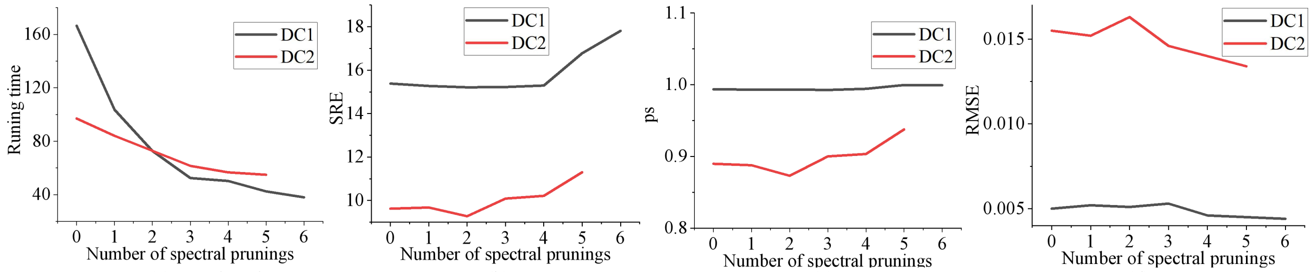


Fig. 4. True abundance map of DC2



(a)Running time

(b)SRE

(c)*ps*

(d)RMSE

Fig. 5. Runtime, SRE, *ps* and RMSE of DC1 and DC2 using the PSU-ATV at 20 dB under different pruning numbers.

In TABLE II, the columns of the iteration number are a string consisting of numbers and the character ”+”. The number of ‘+’ indicates the number of pruning, the number of iterations before pruning is uniformly set to 50, and the number of iterations for unmixing after the last pruning is uniformly set to 200. We observe that as the number of pruning iterations increases, the unmixing speed improves, and the unmixing results become better.

To facilitate observation, we plot the different pruning iterations versus the running time, SRE, *ps* and RMSE for comparison, as shown in Fig. 5. In Fig. 5(a), it is evident that the time required for unmixing decreases rapidly as the pruning numbers increase. DC1 and DC2 have the shortest running times at the maximum pruning numbers. The maximum pruning numbers are 6 and 5, respectively. In Fig. 5(b), the SRE values show little change or a slight decrease with insufficient pruning iterations. However, with enough pruning iterations, the SRE values significantly improve. This indicates that inadequate pruning iterations do not enhance unmixing performance, while sufficient pruning iterations can effectively increase unmixing accuracy. In Fig. 5(c), the *ps* values for DC1 are relatively high, and there is little change as pruning iterations increase. Conversely, at insufficient pruning, *ps* values for DC2 are low and not very variable, but *ps* values increased significantly after enough spectral library pruning. This suggests that increasing pruning numbers can raise *ps* values when *ps* values are initially low, but when the *ps* value reaches a high level, the *ps* value does not change much with an increase in the pruning numbers. In Fig. 5(d), the RMSE value fluctuates up and down when the pruning iteration is insufficient, but as the pruning iteration increases the RMSE value is significantly reduced (the smaller the RMSE value, the better the unmixing effect). This shows that insufficient pruning iteration cannot improve the unmixing performance, while sufficient pruning iteration can improve the unmixing accuracy.

This experiment demonstrates that for DC1 and DC2, with enough pruning iterations, the PSU-ATV achieves optimal running efficiency and computational accuracy. Therefore, in subsequent experiments, we set the pruning iterations to their maximum values 6 for DC1 and 5 for DC2.

TABLE II.
Unmixing results of DC2 estimated by PSU-ATV in the 20dB under different pruning numbers

	DC1					DC2				
Pruning numbers	Iteration number	Running time (sec)	SRE	*ps*	RMSE	Iteration number	Running time (sec)	SRE	*ps*	RMSE
0	200	166.4472	15.3823	0.9936	0.0050	200	97.0159	9.6231	0.8897	0.0155
1	50+200	103.6268	15.2743	0.9932	0.0052	50+200	84.0850	9.6776	0.8877	0.0152
2	50+50+200	72.7588	15.2120	0.9934	0.0051	50+50+200	72.8832	9.2744	0.8730	0.0163
3	50+50+50+200	52.4766	15.2244	0.9927	0.0053	50+50+50+200	61.5767	10.0852	0.9001	0.0146
4	50+50+50+50+200	50.2239	15.2930	0.9943	0.0046	50+50+50+50+200	56.7302	10.2129	0.9034	0.0140
5	50+50+50+50+50+200	42.4606	16.7784	0.9996	0.0045	50+50+50+50+50+200	54.8842	11.3009	0.9377	0.0134
6	50+50+50+50+50+50+200	38.0384	17.8112	0.9993	0.0044					

TABLE III.
The unmixing results of DC2 estimated by PSU-ATV at 20 dB under different iteration numbers

Iteration numbers before pruning	DC1					DC2				
	Iteration numbers	Running time (sec)	SRE	*ps*	RMSE	Iteration numbers	Running time (sec)	SRE	*ps*	RMSE
100	100*6+200	75.1005	17.8112	0.9993	0.0044	100*5+200	98.5671	11.3009	0.9377	0.0134
90	90*6+200	69.2317	17.8112	0.9993	0.0044	90*5+200	87.8088	11.3009	0.9377	0.0134
80	80*6+200	61.2970	17.8112	0.9993	0.0044	80*5+200	78.6829	11.3009	0.9377	0.0134
70	70*6+200	54.0475	17.8112	0.9993	0.0044	70*5+200	69.0338	11.3009	0.9377	0.0134
60	60*6+200	47.3170	17.8112	0.9993	0.0044	60*5+200	62.2848	11.3009	0.9377	0.0134
50	50*6+200	38.0384	17.8112	0.9993	0.0044	50*5+200	54.8842	11.3009	0.9377	0.0134
40	40*6+200	28.5416	17.8112	0.9993	0.0044	40*5+200	42.8064	11.3009	0.9377	0.0134
30	30*6+200	25.2175	17.8112	0.9993	0.0044	30*5+200	31.9655	11.3009	0.9377	0.0134
20	20*6+200	18.2001	14.9159	0.9767	0.0063	20*5+200	26.4917	11.3009	0.9377	0.0134
10	10*6+200	11.6944	11.2047	0.9643	0.0097	10*5+200	13.7126	11.3009	0.9377	0.0134
5	5*6+200	8.1693	-1.3676	0.0268	0.0397	5*5+200	10.9011	9.9344	0.8654	0.0196

TABLE IV.
Unmixing results of DC1 and DC2 under different SNR

Algorithms		DC1			DC2		
		10db	20db	30db	10db	20db	30db
SUnSAL (2011)	SRE	0.1891	3.0382	6.153	1.6613	4.195	8.3632
	ps	0	0.109	0.819	0.2927	0.5559	0.7921
	RMSE	0.0364	0.0253	0.0171	0.0407	0.0304	0.0185
SUnSAL-TV (2012)	SRE	5.8212	9.9123	16.677	3.8530	6.2273	11.4819
	ps	0.8100	0.973	0.996	0.5147	0.6463	0.9505
	RMSE	0.0177	0.0128	0.0075	0.0317	0.0239	0.0129
S^{2}WSU (2018)	SRE	2.4093	5.6022	16.247	3.2175	7.4758	*20.2115*
	ps	0.137	0.7621	**1**	0.4790	0.7837	*0.9995*
	RMSE	0.0247	0.0182	0.0062	0.0385	0.0326	*0.005*
SU-ATV (2021)	SRE	8.5248	16.0722	*22.9589*	4.0348	9.7841	15.6295
	ps	0.8868	0.9929	**1**	0.5236	0.8896	0.9946
	RMSE	0.0128	0.0058	*0.0034*	0.0292	0.0163	0.0088
LGSU (2022)	SRE	0.7619	5.3886	11.619	1.7391	6.9651	17.5631
	ps	0.0630	0.7740	0.9977	0.3307	0.7453	0.9958
	RMSE	0.0378	0.0197	0.0087	0.0399	0.0217	0.0079
LSU (2023)	SRE	-0.24	9.41	17.832	2.16	9.000	19.3353
	ps	0.25	0.9636	**1**	0.3699	0.7823	*0.9998*
	RMSE	0.0353	0.0175	0.0044	0.0430	0.0254	*0.0044*
ATVLRSU (2025)	SRE	*9.5410*	*16.3607*	19.9839	*5.6688*	*10.8889*	14.3619
	ps	*0.9627*	*0.9982*	**1**	*0.6194*	*0.9334*	0.9929
	RMSE	*0.0115*	*0.0052*	0.0037	*0.0256*	*0.014*	0.0094
Ours	SRE	**11.74**	**17.8112**	**24.7507**	**6.3709**	**11.3009**	**22.141**
	ps	**0.9877**	**0.9993**	**1**	**0.6624**	**0.9377**	**0.9999**
	RMSE	**0.009**	**0.0044**	**0.0033**	**0.0268**	**0.0134**	**0.0043**

TABLE V.
Running Time (in seconds) using different algorithms on DC1 and DC2 under 200 Iterations

	SUnSAL	SUnSAL-TV	S^{2}WSU	SU-ATV	LGSU	LSU	ATVLRSU	Ours
DC1	12.8676	130.3231	44.5532	76.1009	125.6412	114.3873	96.1818	**38.0384**
DC2	19.2772	229.4359	66.7235	98.0697	199.7575	196.3757	247.272	**56.2314**

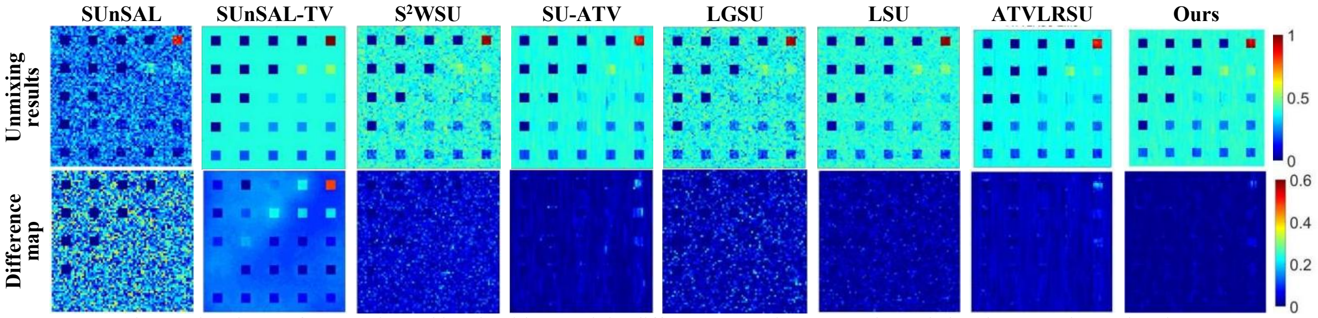


Fig. 6. Abundance of DC1 estimated by algorithms and the difference maps between estimated abundances and true abundances.

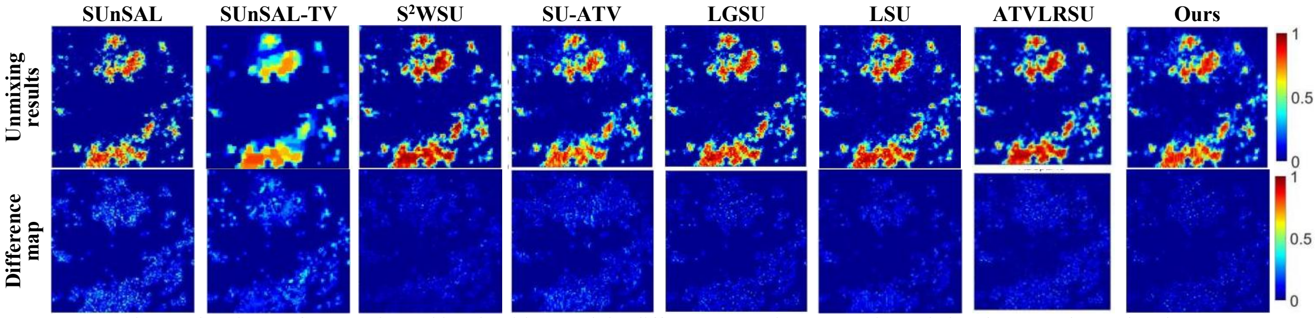


Fig. 7. Abundance of DC2 estimated by algorithms and the difference maps between estimated abundances and true abundances.

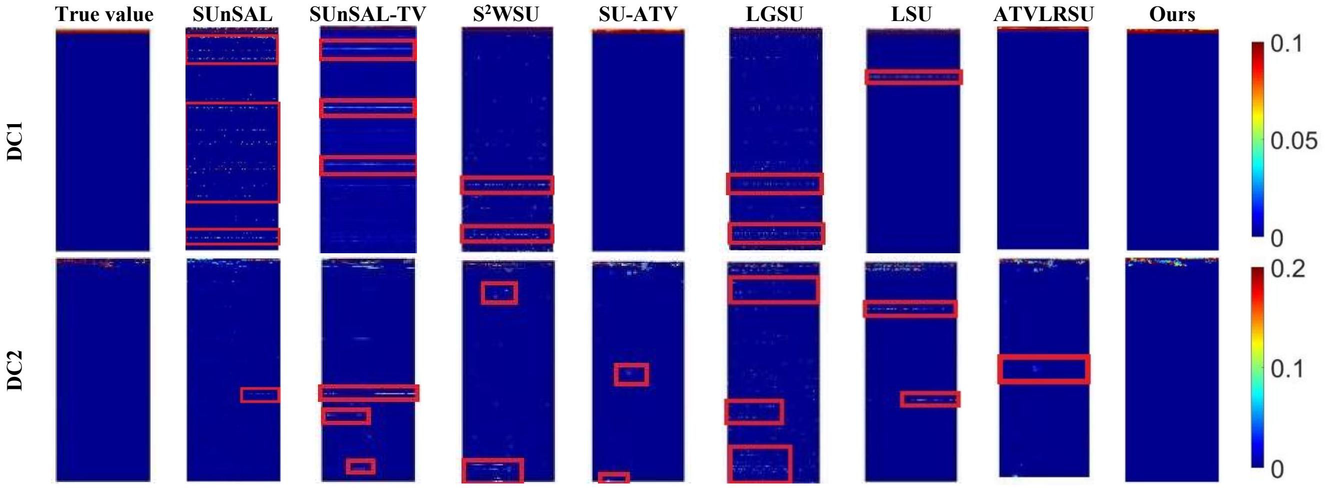


Fig. 8. Abundance of DC1 and DC2 estimated by each algorithm at 20 dB.

In order to detect the relationship between the iteration numbers before pruning and the unmixing results, we based on the above experiments (DC1 pruning 6 times, DC2 pruning 5 times), set the number of iterations before each pruning to 5 or 10 to 100 with a 10 interval, and set the number of iterations after pruning unmixing is uniformly set to 200. The results of the experiment are shown in TABLE III.

In TABLE III, the columns of the iteration numbers are strings consisting of numbers, ”*” and ”+”. The number before ”*” indicates the iteration numbers before pruning (set to 5 or a number in the range of 10-100 at intervals of 10). The number after ”*” indicates the pruning numbers. The number after the ”+” indicates the iteration numbers of unmixing after the last pruning (set uniformly to 200).

From the results in TABLE III, we observe that for both DC1 and DC2 experiments, when the iteration numbers before pruning are between 30 and 100, the unmixing results are close to the estimated optimal values. This suggests that the iteration numbers before pruning do not significantly affect the unmixing results. However, when the iteration numbers are reduced to 10 or 20, the unmixing results for the DC1 experiment show serious deviations. In particular, when the iteration times are 5, there is a huge error in the unmixing results of the two experimental data. This occurs due to insufficient iterations leading to learning errors. Through this experiment, we understand that too few iterations before pruning can significantly disrupt the unmixing results. To avoid such errors, we set the iteration numbers before each pruning to 50. This choice ensures that the unmixing results remain accurate while greatly enhancing unmixing efficiency.

In summary, we define the experimental schemes as follows:

DC1 experimental scheme: pruning number is 6, iteration number before pruning is 50, and iteration number for unmixing after pruning is 200.

DC2 experimental scheme: pruning number is 5, iteration number before pruning is 50, and iteration number for unmixing after pruning is 200.

(2) Experimental Contents

In the unmixing experiments for DC1 and DC2 using different methods, we set the iteration numbers for each algorithm to 200 for fairness (the PSU-ATV uses the experimental protocol established in the previous section). TABLE IV presents the unmixing results for the two simulated datasets. In the table, we use bold to mark the best values and italic to mark the second best values. From this we draw the following conclusions:

1) These methods(SUnSAL-TV, SU-ATV, ATVLRSU and PSU-ATV), which make use of spatial TV regularization term, are better than several other methods in terms of unmixing results. Moreover, our proposed PSU-ATV utilizes both the characteristics of the abundance space and the self-learning capability of the spectral library, leading to further improvements in unmixing accuracy.
2) Under high noise conditions (e.g., 10 dB), several algorithms (SUnSAL, S²WSU, LGSU, LSU) exhibit poor unmixing results due to severe noise interference, and even significant unmixing errors occurring (e.g. SUnSAL, LGSU, and LSU). However, our proposed PSU-ATV method demonstrates excellent robustness, attributed to its strong interference suppression capabilities from the ATV term and the denoising processes implemented during pruning, far surpassing other algorithms.
3) In contrast to the layered sparse regression unmixing

method (LSU), which does not account for the impact of noise on the spectral library, resulting in poor unmixing performance under high noise conditions. Our proposed algorithm exhibits strong resistance to noise interference. It maintains excellent unmixing performance even in high noise environments.

Figures 6 and 7 display the abundance estimation results for two simulated datasets using several algorithms at 20 dB, as well as the difference map between the unmixing estimated abundances and the true abundances. The figures are composed of colors such as red, yellow, green, and blue, and these different colors represent varying weights (red has the highest weight of 1 and blue has the lowest weight of 0). In Fig. 6, the unmixing results for DC1 show that algorithms exhibit considerable spatial error points, such as SUnSAL, S2WSU, LGSU, and LSU. In contrast, SUnSAL-TV, SU-ATV, ATVLRSU and PSU-ATV demonstrate better spatial unmixing performance. However, analysing these three better unmixing results from the difference maps, it can be found that SUnSAL-TV has the largest deviation, followed by SU-ATV and ATVLRSU, and the proposed PSU-ATV yields the best performance. In Fig. 7, from the unmixing results of DC2, it can be seen that several algorithms are good except for the SUnSAL-TV. However, the difference maps illustrate that the proposed method has the smallest error.

Fig. 8 shows the unmixed abundance estimates for the two simulated data at 20 dB, where the areas with obvious errors are circled in red box. From the figure, it is evident that methods such as SUnSAL, SUnSAL-TV, S^{2}WSU, and LGSU exhibit a high number of anomalous abundances, indicating poor unmixing performance. In contrast, SU-ATV, ATVLRSU and the proposed method PSU-ATV show less abundance error. However, in the unmixing results in the DC2, SU-ATV and ATVLRSU exhibit some anomalous abundance values, while the proposed method continues to perform well. This demonstrates that the proposed method possesses strong robustness across different hyperspectral datasets.

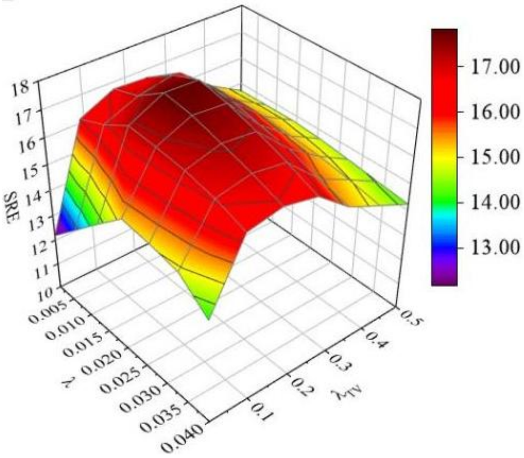


Fig. 9. SRE as a function of parameters λ and λ_{TV} for DC1 estimated by PSU_ATV at 20 dB

To test the stability of the algorithm, this experiment uses the DC1 data as a reference. Create a 3D plot with SRE near the regularisation parameters λ and λ_{TV} of the optimal unmixing result of PSU_ATV, and analyze the influence of the parameters on the outcome. The definition of λ and λ_{TV} , please refer to model (9). Fig. 9 shows the relationship among the parameters λ , λ_{TV} and SRE for DC1 at 20 dB. From the Fig. 9, it can be observed that the plot is centred on the optimal regularisation parameter, λ and λ_{TV} the SRE data shows a slow decreasing trend as λ and λ_{TV} are varied. This indicates that the unmixing results are stable and robust.

TABLE V presents the running time of unmixing using eight different algorithms under 200 iterations. To ensure fairness, all algorithms are run on a PC equipped with an Intel Core i7-10510U CPU and 32 GB of RAM using Matlab R2016a. In terms of time consumption, SUnSAL is the fastest algorithm, followed by the proposed algorithm PSU-ATV and S^{2}WSU. Other algorithms are significantly slower. However, the PSU-ATV greatly improves unmixing accuracy (see TABLE IV). Considering both computational efficiency and algorithm accuracy, the PSU-ATV algorithm, which sacrifices a bit of time for a substantial increase in unmixing accuracy, is the best option.

TABLE VI.
Ablation experiments of DC1 and DC2 using PSU-ATV unmixing at 20 dB

	λ	λ_{TV}	B	Pruning	SRE	*ps*	RMSE
DC1	√	×	×	×	3.0382	0.109	0.0253
	√	√	×	×	9.9123	0.973	0.0128
	√	√	√	×	16.0722	0.9929	0.0058
	√	√	√	√	17.8112	0.9993	**0.0044**
DC2	√	×	×	×	4.195	0.5559	0.0304
	√	√	×	×	6.2273	0.6463	0.0239
	√	√	√	×	9.7841	0.8896	0.0163
	√	√	√	√	11.3009	0.9377	**0.0134**

To evaluate the effectiveness of the various components (λ, λ_{TV} , B and spectral library pruning) of PSU-ATV. We conducted an ablation experiment on the PSU-ATV using DC1 and DC2, and the results are shown in TABLE VI. From TABLE VI, it is evident that adding one or more elements significantly improves the unmixing accuracy of the algorithms. Moreover, the algorithm that combines all four elements yields the most favorable results for both the DC1 and DC2 . This indicates that each element in the algorithm is important and effective.

B. Experiments with Real Data

Since there is no real abundance value in the real data experiment, in order to evaluate the effectiveness of the real data hyperspectral unmixing, we redefine the Signal Reconstruction Error of real data (SRE$_{\mathrm{RD}}$) and the Root Mean Square Error of real data (RMSE$_{\mathrm{RD}}$). A larger SRE$_{\mathrm{RD}}$ value indicates smaller errors, demonstrating better unmixing accuracy. Conversely, a smaller RMSE$_{\mathrm{RD}}$ value signifies smaller errors, indicating higher precision in the unmixing results.

$$\mathrm{SRE}_{\mathrm{RD}}=10\cdot\lg\left(\mathrm{E}\left(\left\|\mathbf{Y}\right\|_2^2\right)\Big/\mathrm{E}\left(\left\|\mathbf{Y}-\mathbf{A}\hat{\mathbf{X}}\right\|_2^2\right)\right). \quad (32)$$

$$\mathrm{RMSE}_{\mathrm{RD}}=\sqrt{\left\|\mathbf{Y}-\mathbf{A}\hat{\mathbf{X}}\right\|_2^2\Big/\left(M\times N\right)}. \quad (33)$$

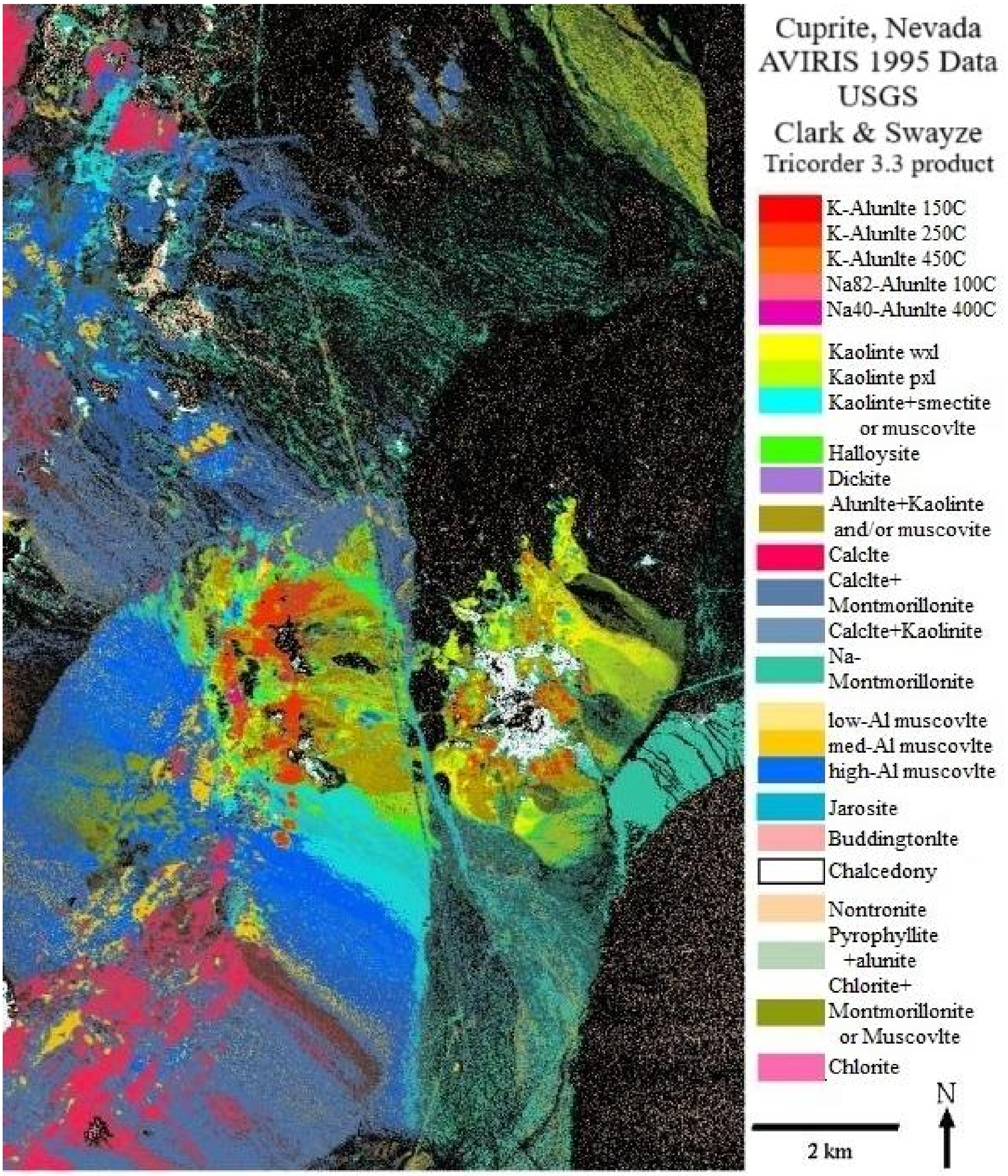


Fig. 10. Distribution of individual minerals in the Cuprite data for Nevada

In this section we utilize the Cuprite mine data acquired by the Airborne Visible Infrared Imaging Spectrometer (AVIRIS) [36] in Nevada in 1995 as real data for our experiments. Fig. 10 presents the mineral distribution map of the Cuprite mining area, generated using the Tricorder 3.3 software [37] for the mineral analysis. The real hyperspectral data consists of 188 bands (after removing 36 noisy bands from the original 224 bands), with wavelengths ranging from 370 nm to 2480 nm . The spatial and spectral resolution of the spectral data are 20 m and 10 nm, respectively. The experimental data is generated by intercepting a 250×191 pixel sized region in the real data. We use the SUnSAL, SUnSAL-TV, S^2WSU, SU-ATV, LGSU, LSU, ATVLRSU and the proposed method PSU-ATV to unmix the real data. And then analyze and compare the results to verify the feasibility of the proposed method.

Based on the references, the estimated number of endmembers for the real data is 12, with the pruning progress q set to 2 and the spectral library consisting of 224 bands. According to equation (28), the maximum iteration number for the PSU-ATV is calculated to be 5. Based on the experience of the simulation experiments, the pruning number is set to 5, the learning and iteration number before pruning is set to 50, and the iteration number for unmixing after pruning is set to 200 on the real data experiment.

Fig. 11 presents the optimal abundance maps obtained by eight different methods (SUnSAL, SUnSAL-TV, S^2WSU, SU-ATV, LGSU, LSU, ATVLRSU and PSU-ATV) through the adjustment of regularization parameters. The best regularization parameters for these methods are as follows: SUnSAL (λ = 0.005), SUnSAL-TV (λ = 0.02, λ_{TV} = 0.015), S^2WSU (λ = 0.41), SU-ATV (λ = 0.03, λ_{TV} = 0.09), LGSU (λ_1 = 0.01, λ_2 = 0.005), LSU (λ = 0.01), ATVLRSU (λ = 0.005, λ_{TV} = 0.15, λ_{WT} = 0.8) and PSU-ATV (λ = 0.01,

$\lambda_{TV} = 0.01, r = 0.8$).

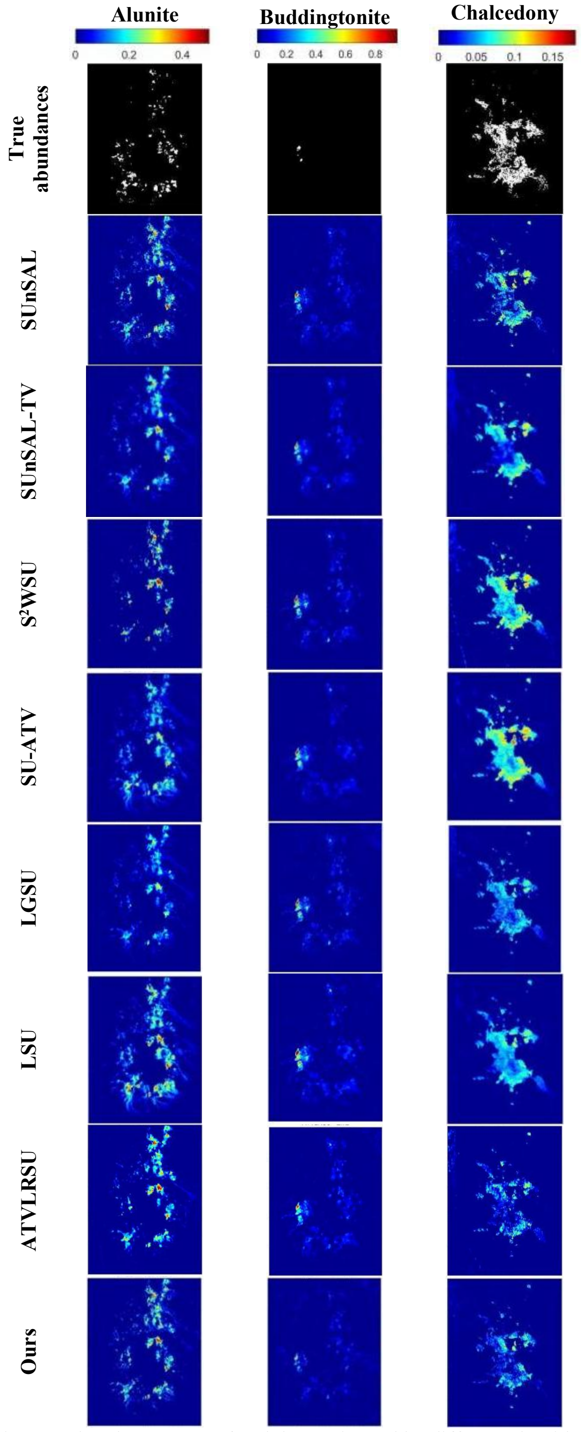


Fig. 11. Abundance maps of real data estimated by different algorithms.

In Fig. 11, Due to the large number of endmembers, only three endmembers with large distinctions for comparison: Alunite, Buddingtonite and Chalcedony. Abundance values are indicated using a color gradient from red to blue, representing values from high to low. The figure shows that the abundance estimated by the eight algorithms is relatively close to the true values. However, SUnSAL, SUnSAL-TV, and LSU exhibit significant discrepancies in the abundance of Alunite. S^{2}WSU and SU-ATV yield overly smooth estimates for Chalcedony, and SUnSAL, LGSU, ATVLRSU and LSU show larger errors for Buddingtonite, while the proposed method PSU-ATV performs well across all three endmembers.

TABLE VII
Unmixing results of different methods on real data

Algorithms	RMSE$_{RD}$	SRE$_{RD}$
SUnSAL	0.0216	19.344
SUnSAL-TV	0.0138	28.337
S^{2}WSU	0.0062	30.132
SU-ATV	0.0054	35.315
LGSU	0.0066	31.764
LSU	0.0056	34.316
ATVLRSU	0.0052	36.431
Ours	0.0046	39.365

Table VII presents the quantitative comparison results using real hyperspectral data. It is evident that the proposed algorithm achieves the best values in both RMSE$_{RD}$ and SRE$_{RD}$, demonstrating its significant advantage over multiple other algorithms.

V. Conclusion

This paper proposes a sparse unmixing algorithm based on spectral library pruning and adaptive total variation constraint. The method employs a novel error-resistant pruning technique to streamline the endmember spectral library, enhancing computational efficiency of unmixing. Additionally, adaptive total variation regularization is applied to effectively denoise hyperspectral data, improving unmixing accuracy. Experimental results on both simulated and real datasets demonstrate that the proposed method significantly outperforms other state-of-the-art algorithms in terms of both computational efficiency and unmixing precision, particularly under low signal-to-noise ratio conditions. Future work will focus on model optimization to further enhance unmixing accuracy.

References

[1] A. K. Mahlein, M. T. Kuska, and J. Behmann, “Hyperspectral sensors and imaging technologies in phytopathology: state of the art,” Annual Review of Phytopathology, vol. 56, pp 535-558, 2018.

[2] A. Plaza, P. Martinez, and R. Perez, “A new approach to mixed pixel classification of hyperspectral imagery based on extended morphological profiles,” Pattern Recognition, vol. 37, no. 6, pp 1097-1116, 2004.

[3] X. R. Feng, H. C. Li, and R Wang, “Hyperspectral unmixing based on nonnegative matrix factorization: A comprehensive review,” IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing, vol. 15, pp 4414-4436, 2022.

[4] X. Tao, M. E. Paoletti, and L. Han, “A new deep convolutional network for effective hyperspectral unmixing,” IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing, vol. 15, pp 6999-7012, 2022.

[5] Y. Gao, B. Pan, and X Xu, “A Reversible Generative Network for Hyperspectral Unmixing with Spectral Variability,” IEEE Transactions on Geoscience and Remote Sensing, vol. 62, pp 5519115, 2024.

[6] N. Keshava, and J. F. Mustard, “Spectral unmixing,” IEEE Signal Processing Magazine, vol. 19, pp 44-57, 2002.

[7] Y. Y. Chan, X. P. Li, and J Mai, “Sparse Unmixing in the Presence of Mixed Noise Using ℓ0-norm Constraint and Log-Cosh Loss,” IEEE Transactions on Geoscience and Remote Sensing, vol. 62, pp 5527419, 2024.

[8] D. Bertsimas, J. Pauphilet, and B. Van Parys, “Sparse regression,” Statistical Science, vol. 34, no. 4, pp 555-578, 2020.

[9] M. D. Iordache, J. M. Bioucas-Dias, and A. Plaza, “Sparse unmixing of hyperspectral data,” IEEE Transactions on Geoscience and Remote Sensing, vol. 49, no. 6, pp 2014-2039, 2011.

[10] A. D. Frankel, C. S. Mueller, and C. S. Barnhardx, “USGS national seismic hazard maps,” Earthquake Spectra, vol. 16, pp 1-19, 2000.

[11] J. C. Gao, J. Y. Shi and F. Zhu. “Robust Sparse Unmixing via Continuous Mixed Norm to Address Mixed Noise,” IEEE Geoscience and Remote Sensing Letters, vol. 22, pp 5502705, 2025.

[12] L. Sun, Z. Wu, and L. Xiao, “A novel ℓ1/2 sparse regression method for hyperspectral unmixing,” International Journal of Remote Sensing,vol. 34, no. 20, pp 6983-7001, 2013.

[13] M. D. Iordache, J. M. Bioucas-Dias, and A. Plaza, “Collaborative sparse regression for hyperspectral unmixing,” IEEE Transactions on Geoscience and Remote Sensing, vol. 52, pp 341-354, 2013.

[14] X. Zhao, M. Song, and T. Yang, “Hyperspectral Unmixing Based on Chaotic Sequence Optimization of ℓp-norm,” IEEE Geoscience and Remote Sensing Letters, vol. 21, pp 5507705, 2024.

[15] M. D. Iordache, J. M. Bioucas-Dias, and A. Plaza, “Total Variation Spatial Regularization for Sparse Hyperspectral Unmixing,” IEEE Transactions on Geoscience and Remote Sensing, Vol. 50, no. 11, pp 4484–4502, 2012.

[16] L. Sun, W. Ge, and Y. Chen, “Hyperspectral unmixing employing L1−L2 sparsity and total variation regularization,” International Journal of Remote Sensing, vol. 39, no. 19, pp 6037-6060, 2018.

[17] R. Wang, H. C. Li, and A. Pizurica, “Hyperspectral unmixing using double reweighted sparse regression and total variation,” IEEE Geoscience and Remote Sensing Letters, vol. 14, no. 7, pp 1146-1150, 2017.

[18] J. Sigurdsson, M. O. Ulfarsson, and J. R. Sveinsson, “Blind Hyperspectral Unmixing Using Total Variation and Lq Sparse Regularization,” IEEE Transactions on Geoscience and Remote Sensing, vol. 54, no. 11, pp 6371-6384. 2016.

[19] D. Zhang, T. Wang, and S. Yang, “Spectral reweighting and spectral similarity weighting for sparse hyperspectral unmixing,” IEEE Geoscience and Remote Sensing Letters, vol. 19, pp 19: 1-5, 2022.

[20] C. Deng, Y. Chen, and S. Zhang, “Robust dual spatial weighted sparse unmixing for remotely sensed hyperspectral imagery,” Remote Sensing, vol. 15, no. 16, pp 4056, 2023.

[21] L. Qi, J. Li, and Y Wang, “Spectral–spatial-weighted multiview collaborative sparse unmixing for hyperspectral images,” IEEE Transactions on Geoscience and Remote Sensing, vol. 58, no. 12, pp 8766-8779, 2020.

[22] C. Xu, “Adaptive Total Variation Regularized for Hyperspectral Unmixing,” Proceedings of IEEE International Conference on Progress in Informatics and Computing, 17-19 December, 2021, Shanghai, China, pp 170-173.

[23] Z. F. Pang, H. L. Zhang, and L. S. Zhang, “Image denoising based on the adaptive weighted TVp regularization,” Signal Processing, vol. 167, pp 107325, 2020.

[24] M. Ma, C. Xu, and J. Zhang, “Hyperspectral sparse unmixing based on a novel adaptive total variation regularization,” Infrared Physics & Technology, vol. 127, pp 104362, 2022.

[25] X. Shen, H. Liu, and X. Zhang, “Superpixel-guided local sparsity prior for hyperspectral sparse regression unmixing,” IEEE Geoscience and Remote Sensing Letters, vol. 19, 19: 1-5, 2022.

[26] X. Shen, L. Chen, and H. Liu, “Efficient hyperspectral sparse regression unmixing with multilayers,” IEEE Transactions on Geoscience and Remote Sensing, vol. 61, pp 5522614, 2023.

[27] K. Wang, L. Zhong, and J Zheng, “Robust Multiscale Spectral-Spatial Regularized Sparse Unmixing for Hyperspectral Imagery,” IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing, vol. 17, pp 1269-1285, 2023.

[28] X. Zou, M. Xu, and S Liu, “Superpixel-based Graph Laplacian Regularized and Weighted Robust Sparse Unmixing,” IEEE Transactions on Geoscience and Remote Sensing, vol. 62, pp 5526415, 2024.

[29] T. Ince, and N. Dobigeon, “Spatial-Spectral Multiscale Sparse Unmixing for Hyperspectral Images,” IEEE Geoscience and Remote Sensing Letters, vol. 20, pp 5511605, 2023.

[30] R. Ren, J. L. Xiao, and J. Huang, “Endmember Distinguished Low-Rank and Sparse Representation for Hyperspectral Unmixing,” Proceedings of IEEE International Geoscience and Remote Sensing Symposium, 07-12 July, 2024, Athens, Greece, pp 7701-7705.

[31] Z. Y. Xiong, H. Long, and Y. F. Zhang, “A neighborhood weighted-based method for the detection of outliers,” Applied Intelligence, vol. 53, no. 9, pp 9897-9915, 2023.

[32] E. Ghadimi, A. Teixeira, and I. Shames, “Optimal parameter selection for the alternating direction method of multipliers (ADMM): Quadratic problems,” IEEE Transactions on Automatic Control, vol. 60, no. 3, pp 644-658, 2014.

[33] R. Gaizman, and Y. Y. Zeevi, “Image unmixing success estimation in spatially varying systems,” Proceedings of IEEE Global Conference on Signal and Information Processing, 14-16 December, Orlando, FL, USA, 2015, pp 1047-1051.

[34] T. Fang, F. Zhu, and J. Chen, “Hyperspecral Unmixing based on Multilinear Mixing Model using Convolutional Autoencoders,” IEEE Transactions on Geoscience and Remote Sensing, vol. 62, pp5507316, 2024.

[35] W. M. Davis, “The topographic maps of the United States Geological Survey,” Science, vol. 534, pp 225-227, 1893.

[36] R. O. Green, M. L. Eastwood, and C. M. Sarture, “Imaging spectroscopy and the airborne visible/infrared imaging spectrometer (AVIRIS),” Remote Sensing of Environment, vol. 65, no. 3, pp 227-248, 1998.

[37] R. N. Clark, G. A. Swayze, and K. E. Livo, “Imaging spectroscopy: Earth and planetary remote sensing with the USGS Tetracorder and expert systems,” Journal of Geophysical Research: Planets, vol. 108, pp E12, 2003.

[38] C. Xu, “Adaptive Total Variation Regularization for Weighted Low-Rank Tensor Sparse Hyperspectral Unmixing,” IAENG International Journal of Applied Mathematics, vol. 54, no. 11, pp 2404-2417, 2024 .

[39] H. Yang, N. Xu, and Y. Hao, “Integrating Pure Pixel Identification into Nonnegative Matrix Factorization for Endmember Extraction,” IAENG International Journal of Computer Science, vol. 46, no. 2, pp 291-299, 2019.

[40] C. Xu, Y. Guo, and F. Li, “Sparse Unmixing of Hyperspectral Images Based on Adaptive Total Variation and Low-Rank Constraints,” Spectroscopy and Spectral Analysis, vol. 45, no. 4, pp 1071-1081, 2025.

Chenguang Xu is a PhD candidate of Intelligent Electromechanical Equipment Innovation Research Institute, East China Jiaotong University, Nanchang, China and an associate professor of School of Information Engineering, Nanchang Institute of Technology, Nanchang, China.