Inverse Problems and Imaging
Vol. 27, 2027, pp. 83-109 (April 7, 2026)
doi:10.3934/ipi.2026032

TOTAL VARIATION REGULARIZATION BASED DEEP ALTERNATING NEURAL NETWORK FOR BLIND UNMIXING OF HYPERSPECTRAL IMAGES

Hong-Ji Xie✉ and Jie Huang✉*

School of Mathematical Sciences, University of Electronic Science and Technology of China, Chengdu, Sichuan, 611731, China

(Communicated by Bin Dong)

Abstract. In this article, we consider an inverse problem of hyperspectral image unmixing. Although significant progress has been made in the current unmixing studies, there are still issues regarding insufficient utilization of spatial information and high computational complexity. Therefore, we propose a novel unmixing method, named sparse and total variation (TV) based deep alternating neural network (STV-ANet), that incorporates sparse and TV regularization terms within a deep alternating neural network framework. In particular, an innovative alternating algorithm is first developed to solve the unmixing problem based on sparse and TV regularization, which requires only a few auxiliary variables. Then this alternating algorithm is unrolled within a deep neural network, integrating the prior knowledge provided by the regularization terms with the data features extracted by the network. The proposed algorithm improves the interpretability of the deep neural network, while the network liberates the algorithm from the adjustment of hyperparameters. Experiments on both synthetic and real-world datasets demonstrate the superior unmixing performance of our STV-ANet over state-of-the-art methods, particularly in the suppression of noise and the recovery accuracy of abundance maps.

1. **Introduction.** Hyperspectral images (HSIs) contain both spatial information of objects and spectral information of numerous continuous narrow band indexes. Unlike conventional images, the spectral information enables more effective identification of the constituent materials, known as endmembers, and their respective proportions, referred to as abundances, within each pixel [28, 35]. This identification process is termed hyperspectral unmixing (HU). Based on the supposition of the linear spectral mixture model (LMM), HU can be viewed as an ill-posed inverse problem:

$$\mathbf{Y} = \mathbf{AS} + \mathbf{E}, \tag{1}$$

where $\mathbf{Y} \in \mathbb{R}^{L\times N}$ denotes the observed HSI, which contains L bands and N pixels, $\mathbf{A} \in \mathbb{R}^{L\times R}$ denotes the endmember matrix with R materials, $\mathbf{S} \in \mathbb{R}^{R\times N}$ denotes the corresponding abundance matrix, and $\mathbf{E} \in \mathbb{R}^{L\times N}$ represents additive noise.

2020 *Mathematics Subject Classification.* Primary: 94A08, 68U10; Secondary: 49N45.

Key words and phrases. Hyperspectral images, deep network, total variation, sparse, unmixing.

This research is supported in part by NSFC (Grant No. 12371517) and in part by Sichuan Science and Technology Program (Grant No. 2024NSFSC0038).

*Corresponding author: Jie Huang.

Generally, the endmember matrix $\mathbf{A}$ is required to be non-negative to comply with the physical properties of reflectance or absorption. The abundance sum-to-one constraint (ASC) and the abundance non-negative constraint (ANC) are imposed on the abundance matrix $\mathbf{S}$, ensuring the rationality and robustness of the unmixing. Their mathematical expressions are given by

$$\text{ASC: } \mathbf{1}_R^T\mathbf{S} = \mathbf{1}_N^T, \ \text{ANC: } \mathbf{S} \geq 0, \tag{2}$$

where $\mathbf{1}_R$ and $\mathbf{1}_N$ are all-one column vectors with lengths R and N, respectively. Usually, HU can be considered as a blind source separation problem [23], particularly when estimating endmembers and their abundances simultaneously. In this study, we focus on recovering both $\mathbf{A}$ and $\mathbf{S}$ from the known $\mathbf{Y}$, which is framed as a blind unmixing problem.

Non-negative matrix factorization (NMF) is prevalently adopted for addressing this problem, which decomposes an HSI into two matrices representing endmembers and abundances, respectively. Additional priors are introduced into the traditional NMF model to improve the unmixing performance, with the intention of compressing the solution space and enhancing the interpretability of the model [19, 37]. In this case, an optimization problem for HU can be depicted as follows:

$$\min_{\mathbf{A},\mathbf{S}} \mathcal{L}(\mathbf{A},\mathbf{S}) = \frac{1}{2}\|\mathbf{Y} - \mathbf{A}\mathbf{S}\|_F^2 + \lambda f(\mathbf{S}) \quad \text{s.t.} \quad \mathbf{1}_R^T\mathbf{S} = \mathbf{1}_N^T, \ \mathbf{A},\mathbf{S} \geq 0, \tag{3}$$

where $\|\cdot\|_F$ denotes the Frobenius norm, $f(\cdot)$ represents the prior-related regularization term, and λ is a non-negative regularization parameter, balancing data fidelity and regularization. Specifically, the sparse regularization term [7, 19, 32, 34] and the total variation (TV) regularization term [1, 11, 15, 25] are widely acknowledged as effective prior conditions in practice. Although advancements in the NMF provide significant benefits such as enhanced interpretability, it is important to recognize the limitations of many traditional model-based unmixing techniques. These methods often rely heavily on specific models and are challenged by the complexity of large parameter spaces.

Alternatively, learning-based methods, especially deep learning networks, differ from model-based approaches in that they can directly extract features from data. Among learning-based methods, autoencoders are widely used for HU, functioning as data compression models via neural networks. Variants like variational autoencoders (VAEs) [5], stacked autoencoders (SAEs) [24], and convolutional autoencoders (CAEs) [18] enhance efficiency and accuracy. However, they face the "black-box" problem, i.e., the difficulty in understanding the inner workings of a neural network, due to its data-driven nature [2].

As a complement to both model-based and learning-based methods, algorithm unrolling combines their advantages by embedding physical information into the architecture of deep learning networks [6, 14, 16]. In the field of HU, SNMF-Net [31] unfolds the NMF model with the L_p-norm regularization within a deep neural network framework, which significantly improves the unmixing performance. It focuses on sparse unmixing, concentrating on the spectral information of HSIs, while neglecting the spatial information of abundance maps.

In this paper, we propose a highly interpretable unrolling network for HU, named sparse and TV based deep alternating neural network (STV-ANet), which first unrolls an alternating algorithm related to the L_p-norm and a TV regularization for the abundance matrix of HSIs (denoted as TV_H) within a deep alternating neural network based on the NMF. The schematic overview of this network is presented in

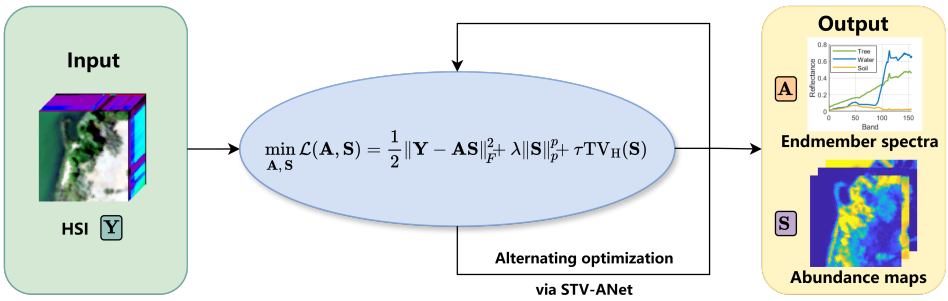


FIGURE 1. Schematic overview of STV-ANet. This image illustrates the process of estimating endmember spectra and abundance maps using NMF based on the L_p-norm and the TV_{H} regularization, through optimization iterations and network learning.

Figure 1. As illustrated in this figure, the HSI is fed into the network, where an NMF model based on the L_p-norm and the TV_{H} is employed to decompose it. More details about TV_{H} can be found in Section 2.3. During the optimization process, STV-ANet alternately updates variables and adaptively learns the parameters, ultimately yielding the desired endmembers and abundances. The key contributions of this study can be encapsulated in the following three points.

1) We devise a novel alternating minimization algorithm to address the unmixing problem involving two regularization terms: sparsity regularization and TV regularization. This alternating approach needs only one auxiliary variable. It updates the variables alternately, which accelerates the iteration process and enhances the efficiency of intermediate variable utilization.

2) We propose a highly interpretable unrolling network STV-ANet for HU. The algorithm's unrolling within the network endows it with multiple advantages. Firstly, the interpretability of all parameters and submodules significantly mitigates the "black-box" issue associated with deep neural networks. Secondly, STV-ANet effectively balances the model's knowledge-driven priors with the network's data-driven priors. Notably, the integration of the TV_{H} regularization improves the spatial information of abundance maps. Lastly, the network optimizes parameters through end-to-end learning, avoiding the need for manual adjustment of model parameters and enhancing data efficiency.

3) The effectiveness of STV-ANet is validated on both synthetic and real-world datasets, demonstrating superior performance compared to other state-of-the-art methods. Specifically, STV-ANet achieves superior unmixing results in accuracy, outperforming leading methods. Moreover, it shows remarkable noise robustness, particularly in the recovery of background details.

The remaining sections are organized as follows. Section 2 introduces prior knowledge, including sparsity, traditional TV, and the proposed TV_{H}. Section 3 presents the HU model with the L_p-norm and the TV_{H} regularization, along with a novel alternating algorithm. Section 4 elaborates on how the proposed algorithm is unrolled into a network and describes the network architecture. Section 5 presents the experimental performance and the implementation details of the proposed STV-ANet. Finally, Section 6 concludes this article.

2. Preliminary knowledge.

2.1. Sparse prior.

Many studies demonstrate that sparse priors enhance the accuracy of unmixing [8, 12, 19, 32, 33]. Typically, sparse priors assume that most pixels in a scene contain only a few endmembers, resulting in an abundance matrix with a large number of zeros. The L_p-norm [19] is a suitable choice for sparsity representation, where the parameter p determines the degree of sparsity, with $0 < p \leq 1$. When $p = 1$, it is the L_1-norm. Commonly, the L_p-norm regularization problem for HU is formulated as follows:

$$\min_{\mathbf{S}} \frac{1}{2}\|\mathbf{Y} - \mathbf{AS}\|_F^2 + \mu_1 \|\mathbf{S}\|_p^p, \tag{4}$$

where μ_1 is a positive regularization parameter, and

$$\|\mathbf{S}\|_p^p = \sum_{i=1}^{R}\sum_{j=1}^{N} |s_{i,j}|^p \tag{5}$$

is a sparse regularized term with $s_{i,j}$ being the (i,j)-th element of $\mathbf{S}$. The problem (4) can be solved by the generalized iterated shrinkage algorithm (GISA) [38]. To elaborate, in each iteration of GISA, a gradient descent is first performed, followed by the application of a generalized soft-thresholding (GST) operator, i.e.,

$$\mathbf{S}^{(k+1)} = \mathcal{G}_{\mu_1,p}\left(\mathbf{S}^{(k)} - t\mathbf{A}^T(\mathbf{AS}^{(k)} - \mathbf{Y})\right), \tag{6}$$

where the GST operator $\mathcal{G}_{\mu_1,p}(\cdot)$, parameterized by μ_1 and p, is defined as follows:

$$\mathcal{G}_{\mu_1,p}(x) = \begin{cases} \operatorname{sgn}(x)(|x| - \mu_1 p|x|^{p-1}), & \text{if } |x| > \Delta_p^{\mathcal{G}}(\mu_1), \\ 0, & \text{otherwise}, \end{cases} \tag{7}$$

where $\operatorname{sgn}(x)$ denotes the sign of x, and $\Delta_p^{\mathcal{G}}(\mu_1)$ is the thresholding value defined as

$$\Delta_p^{\mathcal{G}}(\mu_1) = (2\mu_1(1-p))^{\frac{1}{2-p}} + \mu_1 p\,(2\mu_1(1-p))^{\frac{p-1}{2-p}}. \tag{8}$$

2.2. TV prior.

The TV prior achieves image smoothing by minimizing the TV of an image, while preserving edge features. A general TV-regularized non-convex problem is given by

$$\min_{\mathbf{X}} \frac{1}{2}\|\mathbf{G} - \mathbf{X}\|_F^2 + \mu_2 \mathrm{TV}(\mathbf{X}), \tag{9}$$

here $\mathbf{G} \in \mathbb{R}^{m\times n}$ represents an observed image, $\mu_2 > 0$ is a regularization parameter, and $\mathrm{TV}(\cdot)$ is the anisotropic TV operator [1], which is defined by

$$\begin{aligned} \mathrm{TV}(\mathbf{X}) = &\sum_{i=1}^{m-1}\sum_{j=1}^{n-1} (|x_{i,j} - x_{i+1,j}| + |x_{i,j} - x_{i,j+1}|) \\ &+ \sum_{i=1}^{m-1} |x_{i,n} - x_{i+1,n}| + \sum_{j=1}^{n-1} |x_{m,j} - x_{m,j+1}|, \end{aligned} \tag{10}$$

where $x_{i,j}$ represents the pixel value of $\mathbf{X}$ at position (i,j). The fast gradient projection (FGP) [1] is an efficient method for solving the problem (9). As described in [1], the TV-regularized solution $\mathbf{X}^*$ can be obtained by inputting the observed image $\mathbf{G}$, the regularization parameter μ_2, and the maximum number of inner iterations H_m into the FGP operator, i.e.,

$$\mathbf{X}^* = \mathrm{FGP}(\mathbf{G}, \mu_2, H_m). \tag{11}$$

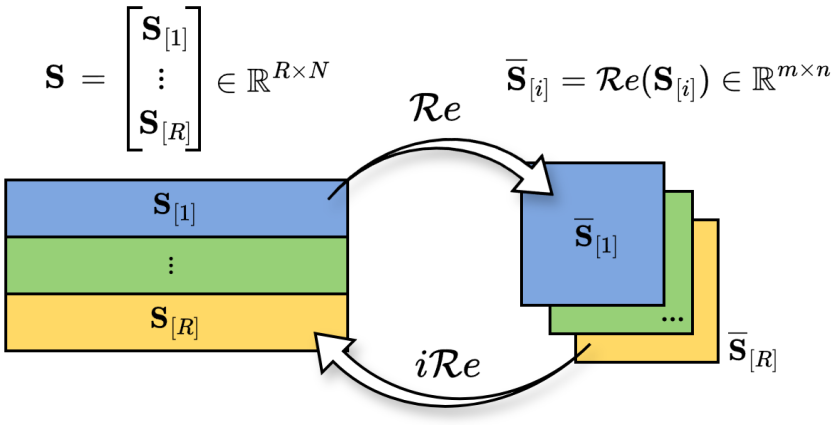


FIGURE 2. Visualization of the operator $\mathcal{R}e$ and its inverse $i\mathcal{R}e$ reshaping the rows of the abundance matrix $\mathbf{S}$.

2.3. **Proposed $\mathrm{TV_H}$ prior for the abundance matrix.** To capitalize on the spatial information in HSIs, scholars introduce the TV prior into HU [3, 8, 15, 27, 30, 33]. In the tensor context [29, 36], the abundance maps can be naturally represented as slices of the abundance tensor along the endmember dimension. However, within the matrix framework, performing TV regularization directly on each abundance map is not straightforward. For example, the authors in [13] use the symbol ϵ to represent the set of horizontal and vertical neighbors in the image and continue to use TV as the notation for the abundance matrix. In contrast, TV-RSNMF [8] reshapes the abundance vector of each pixel to a matrix of image size to compute the TV, denoting it as HTV. We propose a new symbol $\mathrm{TV_H}$, specifically tailored for the abundance matrix. This is motivated by the absence of a well-established mathematical operator to analyze the smoothness of the abundance maps within a matrix framework while the HTV is already employed to denote the high-order TV or the hybrid TV. It is worth noting that this operator is applicable not only to the abundance matrix but also to the entire HSI.

To facilitate subsequent studies, we first provide a systematic introduction to the transformations of the abundances before introducing the operator $\mathrm{TV_H}$. To begin with, we partition the abundance matrix by rows, i.e.,

$$\mathbf{S} = \begin{bmatrix} \mathbf{S}_{[1]} \\ \mathbf{S}_{[2]} \\ \vdots \\ \mathbf{S}_{[R]} \end{bmatrix} \in \mathbb{R}^{R\times N}. \tag{12}$$

Following [35], the operator $\mathcal{R}e$ is used as a reshaping operator that transforms the abundance map of the i-th endmember from an N-element row vector $\mathbf{S}_{[i]}$ to a corresponding $m \times n$ matrix $\overline{\mathbf{S}}_{[i]}$, where we have $N = mn$. Its mathematical expression is

$$\begin{aligned} \mathcal{R}e : \mathbb{R}^N &\to \mathbb{R}^{m\times n} \\ \mathbf{S}_{[i]} &\mapsto \overline{\mathbf{S}}_{[i]}. \end{aligned} \tag{13}$$

At the element level, $\mathbf{S}_{[i]}$ and $\overline{\mathbf{S}}_{[i]}$ have the following correspondences. For any $s_{i,j} \in \mathbf{S}_{[i]}$, there exists a unique $\overline{s}_{a,b} \in \overline{\mathbf{S}}_{[i]}$ such that $\overline{s}_{a,b} = s_{i,j}$, where $i = 1, \ldots, R$, $j = 1, \ldots, N$, $a = 1, \ldots, m$, and $b = 1, \ldots, n$. The relationship between a, b and j is given by $a = \lfloor \frac{j-1}{n} \rfloor + 1$ and $b = j - (a-1)n$. With $s_{i,j}$ and $\overline{s}_{a,b}$ being in a one-to-one correspondence, we define the inverse operator of $\mathcal{R}e$, i.e., $i\mathcal{R}e$ as

$$\begin{aligned} i\mathcal{R}e : \mathbb{R}^{m\times n} &\to \mathbb{R}^{N} \\ \overline{\mathbf{S}}_{[i]} &\mapsto \mathbf{S}_{[i]}. \end{aligned} \tag{14}$$

The various morphologies of the abundance map, as well as the reshaping processes applied to it using the $\mathcal{R}e$ operator and the $i\mathcal{R}e$ operator, are represented diagrammatically in Figure 2.

Clearly, based on the definitions of $\mathcal{R}e$ and $i\mathcal{R}e$, we have

$$\mathbf{S}_{[i]} = i\mathcal{R}e(\overline{\mathbf{S}}_{[i]}) = i\mathcal{R}e(\mathcal{R}e(\mathbf{S}_{[i]})), \tag{15}$$

and therefore for the entire matrix $\mathbf{S}$, we have

$$\mathbf{S} = \begin{bmatrix} i\mathcal{R}e(\mathcal{R}e(\mathbf{S}_{[1]})) \\ i\mathcal{R}e(\mathcal{R}e(\mathbf{S}_{[2]})) \\ \vdots \\ i\mathcal{R}e(\mathcal{R}e(\mathbf{S}_{[R]})) \end{bmatrix}. \tag{16}$$

Subsequently, by employing the $\mathcal{R}e$ operator, we reshape each row of the $R \times N$ abundance matrix $\mathbf{S}$ into an abundance matrix of size $m \times n$. We then apply the TV operator, as defined in (10), to each reshaped matrix. Thereby the TV of the abundance maps is defined as the sum of the TV values of all individual reshaped matrices as

$$\mathrm{TV}_{\mathrm{H}}(\mathbf{S}) = \sum_{i=1}^{R} \mathrm{TV}(\mathcal{R}e(\mathbf{S}_{[i]})). \tag{17}$$

The goal of the abundance matrix denoising problem with the TV_{H} regularization is to recover the denoised abundance matrix $\mathbf{S}$ from the given noisy abundance matrix $\mathbf{D} \in \mathbb{R}^{R\times N}$. This convex optimization problem is defined as

$$\min_{\mathbf{S}} \frac{1}{2}\|\mathbf{D} - \mathbf{S}\|_F^2 + \mu_3 \mathrm{TV}_{\mathrm{H}}(\mathbf{S}). \tag{18}$$

Here, $\|\mathbf{D} - \mathbf{S}\|_F^2$ measures the fidelity between the denoised and noisy abundances, ensuring that the solution remains close to the observed data. The term TV_{H} promotes spatial and spectral smoothness in the abundance matrix while being robust to outliers. The regularization parameter $\mu_3 > 0$ balances the trade-off between data fidelity and regularization strength.

From the perspective of each abundance map, (18) can be reformulated as the following minimization issue

$$\min_{\mathbf{S}} \sum_{i=1}^{R} \left(\frac{1}{2}\|\mathcal{R}e\left(\mathbf{D}_{[i]}\right) - \mathcal{R}e\left(\mathbf{S}_{[i]}\right)\|_F^2 + \mu_3 \mathrm{TV}\left(\mathcal{R}e\left(\mathbf{S}_{[i]}\right)\right) \right), \tag{19}$$

where $\mathbf{D}_{[i]}$ denotes the i-th row of $\mathbf{D}$, for $i = 1, \ldots, R$. For convenience, let $\overline{\mathbf{D}}_{[i]} = \mathcal{R}e\left(\mathbf{D}_{[i]}\right)$, $\overline{\mathbf{S}}_{[i]} = \mathcal{R}e\left(\mathbf{S}_{[i]}\right)$. Accordingly, (19) can be decomposed independently into

$$\min_{\overline{\mathbf{S}}_{[i]}} \frac{1}{2}\|\overline{\mathbf{D}}_{[i]} - \overline{\mathbf{S}}_{[i]}\|_F^2 + \mu_3 \mathrm{TV}\left(\overline{\mathbf{S}}_{[i]}\right), \quad i = 1, \ldots, R. \tag{20}$$

Algorithm 1 $\mathcal{H}_{\mu_3,H_m}(\cdot)$ for the $\mathrm{TV_H}$-regularized problem (18)

Input: Observed abundance matrix $\mathbf{D}$, regularization parameter μ_3, and maximum inner iteration count H_m

1: Apply TV to each of the R rows of $\mathbf{D}$:
2: **for** $i = 1, \ldots, R$ **do**
3: Reshape: $\overline{\mathbf{D}}_{[i]} = \mathcal{R}e\left(\mathbf{D}_{[i]}\right)$
4: $\overline{\mathbf{S}}^*_{[i]} = \mathrm{FGP}\left(\overline{\mathbf{D}}_{[i]}, \mu_3, H_m\right)$
5: **end for**
6: Reshape back: $\mathbf{S}^* = \left[i\mathcal{R}e\left(\overline{\mathbf{S}}^*_{[1]}\right)^T \quad i\mathcal{R}e\left(\overline{\mathbf{S}}^*_{[2]}\right)^T \quad \ldots \quad i\mathcal{R}e\left(\overline{\mathbf{S}}^*_{[R]}\right)^T\right]^T$

Output: $\mathrm{TV_H}$-regularized abundance matrix $\mathbf{S}^*$

They are TV-regularized optimization problems akin to (9), which can be solved by FGP. Thus, for each row of $\mathbf{S}$, it follows that

$$\overline{\mathbf{S}}^*_{[i]} = \mathrm{FGP}\left(\overline{\mathbf{D}}_{[i]}, \mu_3, H_m\right), \quad i = 1, \ldots, R. \tag{21}$$

Then, we use the reshaping operator $i\mathcal{R}e$ to integrate these matrices, obtained from (21), and construct an estimated abundance matrix $\mathbf{S}^*$, which represents an approximate solution to (18), given by

$$\mathbf{S}^* = \begin{bmatrix} i\mathcal{R}e\left(\overline{\mathbf{S}}^*_{[1]}\right) \\ i\mathcal{R}e\left(\overline{\mathbf{S}}^*_{[2]}\right) \\ \vdots \\ i\mathcal{R}e\left(\overline{\mathbf{S}}^*_{[R]}\right) \end{bmatrix}. \tag{22}$$

For the sake of consistency and convenience, we define the process of solving the $\mathrm{TV_H}$-regularized problem as an operator $\mathcal{H}_{\mu_3,H_m}(\cdot)$ and summarize it in Algorithm 1. Thus, the aforementioned process can be simplified to

$$\mathbf{S}^* = \mathcal{H}_{\mu_3,H_m}\left(\mathbf{D}\right). \tag{23}$$

We now analyze the computational complexity of Algorithm 1. The per-iteration complexity of the FGP subroutine scales as $O(mn)$, yielding an aggregate cost of $O(h_i mn)$ for the i-th outer loop, where h_i is the number of inner FGP iterations. After R outer loop, Algorithm 1 accumulates a total complexity of $O(\sum_{i=1}^{R} h_i mn)$, equivalently $O(\overline{h}N)$, with $\overline{h} = \sum_{i=1}^{R} h_i$ and $N = mn$.

3. **Proposed L_p-$\mathrm{TV_H}$ HU model and algorithm.** We introduce the sparse prior (5) and the $\mathrm{TV_H}$ prior (17) in the form of regularization to the problem (3). Thus, under the NMF framework, the HU model with the L_p-norm and the $\mathrm{TV_H}$ regularization (hereafter the L_p-$\mathrm{TV_H}$ HU model) can be formulated as

$$\min_{\mathbf{A},\mathbf{S}} \mathcal{L}(\mathbf{A},\mathbf{S}) = \frac{1}{2}\|\mathbf{Y}-\mathbf{A}\mathbf{S}\|_F^2 + \lambda\|\mathbf{S}\|_p^p + \tau\mathrm{TV_H}(\mathbf{S}) \quad \text{s.t.} \quad \mathbf{1}_R^T\mathbf{S} = \mathbf{1}_N^T,\ \mathbf{A},\mathbf{S} \geq 0, \tag{24}$$

where the positive regularization parameters λ and τ govern the sparsity and smoothness of the abundance matrix, respectively, $\|\cdot\|_p$ indicates the L_p-norm and $\mathrm{TV_H}(\cdot)$ signifies the TV regularization for each abundance map.

We solve the L_p-$\mathrm{TV_H}$ HU model (24) within the framework of alternating minimization, which iteratively updates multiple variables to progressively converge

towards a near-optimal solution. To begin, we introduce an auxiliary variable to the model, which allows us to decompose the complex optimization problem into several more manageable subproblems. In this way, (24) can be rewritten into the following form

$$\begin{aligned}\min_{\mathbf{A},\mathbf{S},\mathbf{S}_1} \mathcal{L}_1(\mathbf{A},\mathbf{S},\mathbf{S}_1) = \frac{1}{2}\|\mathbf{Y}-\mathbf{A}\mathbf{S}\|_F^2 + \frac{\alpha}{2}\|\mathbf{S}-\mathbf{S}_1\|_F^2 + \lambda\|\mathbf{S}\|_p^p + \tau \mathrm{TV}_{\mathrm{H}}(\mathbf{S}_1) \\ \text{s.t.} \quad \mathbf{1}_R^T\mathbf{S} = \mathbf{1}_N^T, \mathbf{1}_R^T\mathbf{S}_1 = \mathbf{1}_N^T, \mathbf{A},\mathbf{S},\mathbf{S}_1 \geq 0\end{aligned} \tag{25}$$

where the auxiliary variable $\mathbf{S}_1 \in \mathbb{R}^{R\times N}$, and $\alpha > 0$ measures the similarity between $\mathbf{S}$ and $\mathbf{S}_1$. Here, we impose the same constraints on $\mathbf{S}_1$ as those on $\mathbf{S}$ to further regulate the abundance matrix. The alternating minimization approach for updating (25) proceeds with the following steps:

$$\begin{cases} \mathbf{A}^{(k+1)} = \underset{\mathbf{A}\geq 0}{\arg\min}\, \mathcal{L}_1\left(\mathbf{A}, \mathbf{S}^{(k)}, \mathbf{S}_1^{(k)}\right), \\ \mathbf{S}^{(k+1)} = \underset{\mathbf{S}\geq 0, \mathbf{1}_R^T\mathbf{S}=\mathbf{1}_N^T}{\arg\min}\, \mathcal{L}_1\left(\mathbf{A}^{(k+1)}, \mathbf{S}, \mathbf{S}_1^{(k)}\right), \\ \mathbf{S}_1^{(k+1)} = \underset{\mathbf{S}_1\geq 0, \mathbf{1}_R^T\mathbf{S}_1=\mathbf{1}_N^T}{\arg\min}\, \mathcal{L}_1\left(\mathbf{A}^{(k+1)}, \mathbf{S}^{(k+1)}, \mathbf{S}_1\right). \end{cases} \tag{26}$$

We address these subproblems sequentially.

• **A-subproblem.** After dropping fixed variables, the $\mathbf{A}$-subproblem becomes

$$\min_{\mathbf{A}} \frac{1}{2}\|\mathbf{Y}-\mathbf{A}\mathbf{S}^{(k)}\|_F^2 \quad \text{s.t.} \quad \mathbf{A}\geq 0. \tag{27}$$

It is a convex optimization problem with a non-negativity constraint. The projected gradient (PG) [4] is a straightforward and efficient iterative approach for this problem. Drawing on the PG approach, we perform one gradient descent step and one projection step in each iteration to compute $\mathbf{A}^{(k+1)}$. First, the gradient descent is employed to update the variable as

$$\mathbf{A}^{(k+\frac{1}{2})} = \mathbf{A}^{(k)} - t_1(\mathbf{A}^{(k)}\mathbf{S}^{(k)} - \mathbf{Y})(\mathbf{S}^{(k)})^T, \tag{28}$$

where $\mathbf{A}^{(k+\frac{1}{2})}$ denotes the intermediate variable obtained by the gradient step, and $t_1 > 0$ is a step size. We then project $\mathbf{A}^{(k+\frac{1}{2})}$ onto the feasible set to ensure the non-negativity and obtain

$$\mathbf{A}^{(k+1)} = \mathcal{P}\left(\mathbf{A}^{(k+\frac{1}{2})}\right), \tag{29}$$

where $\mathcal{P}(\cdot) = \max(\cdot, 0)$ is a projection operator that enforces the non-negativity constraint.

• **S-subproblem.** Proceeding to the $\mathbf{S}$-subproblem, we encounter the following formulation

$$\min_{\mathbf{S}} \frac{1}{2}\|\mathbf{Y}-\mathbf{A}^{(k+1)}\mathbf{S}\|_F^2 + \frac{\alpha}{2}\|\mathbf{S}-\mathbf{S}_1^{(k)}\|_F^2 + \lambda\|\mathbf{S}\|_p^p \quad \text{s.t.} \quad \mathbf{1}_R^T\mathbf{S} = \mathbf{1}_N^T, \mathbf{S}\geq 0. \tag{30}$$

The difference between the objective function of $\mathbf{S}$-subproblem and the L_p-norm non-convex sparse problem (4) is that here are two terms related to the Frobenius norm in (30). Due to the presence of gradients in both the Frobenius norm terms, we can directly apply the gradient descent method to get the intermediate update $\mathbf{S}^{(k+\frac{1}{2})}$ as

$$\mathbf{S}^{(k+\frac{1}{2})} = \mathbf{S}^{(k)} - t_2\left((\mathbf{A}^{(k+1)})^T(\mathbf{A}^{(k+1)}\mathbf{S}^{(k)} - \mathbf{Y}) + \alpha(\mathbf{S}^{(k)} - \mathbf{S}_1^{(k)})\right), \tag{31}$$

Algorithm 2 L_p-$\mathrm{TV_H}$ alternating minimization algorithm

Input: Noisy HSI $\mathbf{Y}$, parameters: λ, p, α, τ, t_1, t_2, maximum number of outer iterations K_m and maximum number of inner iterations H_m for $\mathrm{TV_H}$

Initialization: $\mathbf{A}^{(0)}$, $\mathbf{S}^{(0)} = \mathbf{S}_1^{(0)}$

while $k = 0, \ldots, K_m$ **do**

  Update $\mathbf{A}^{(k+1)} = \mathcal{P}\left(\mathbf{A}^{(k)} - t_1(\mathbf{A}^{(k)}\mathbf{S}^{(k)} - \mathbf{Y})(\mathbf{S}^{(k)})^T\right)$

  Calculate $\mathbf{S}^{(k+\frac{1}{2})} = \mathbf{S}^{(k)} - t_2\left((\mathbf{A}^{(k+1)})^T(\mathbf{A}^{(k+1)}\mathbf{S}^{(k)} - \mathbf{Y}) + \alpha(\mathbf{S}^{(k)} - \mathbf{S}_1^{(k)})\right)$

  Update $\mathbf{S}^{(k+1)} = \mathcal{N}\left(\mathcal{P}\left(\mathcal{G}_{\lambda,p}\left(\mathbf{S}^{(k+\frac{1}{2})}\right)\right)\right)$

  Update $\mathbf{S}_1^{(k+1)} = \mathcal{N}\left(\mathcal{P}\left(\mathcal{H}_{\frac{\tau}{\alpha},H_m}\left(\mathbf{S}^{(k+1)}\right)\right)\right)$

end while

Output: $\hat{\mathbf{A}} = \mathbf{A}^{(K_m)}$, $\hat{\mathbf{S}} = \mathbf{S}^{(K_m)}$

where $t_2 > 0$ is a step size. Then we utilize the GST operator to ensure the sparsity [38], and impose the non-negativity projection operator and the normalization function to satisfy the constraint conditions ANC and ASC respectively. To elaborate, the GST operator $\mathcal{G}_{\lambda,p}(\cdot)$ and the non-negativity projection operator $\mathcal{P}(\cdot)$ adhere to the definitions presented in (7) and (29), respectively. Furthermore, the normalization function $\mathcal{N}(\cdot)$ is characterized by

$$\mathcal{N}(s_{i,j}) = \frac{s_{i,j}}{\sum_{i=1}^{R} s_{i,j}}, \tag{32}$$

which is applied to normalize the columns of the intermediate solution after each iteration. These steps lead to an approximate solution of the problem (30) as follows:

$$\mathbf{S}^{(k+1)} = \mathcal{N}\left(\mathcal{P}\left(\mathcal{G}_{\lambda,p}\left(\mathbf{S}^{(k+\frac{1}{2})}\right)\right)\right). \tag{33}$$

• **$\mathbf{S}_1$-subproblem.** Lastly, the $\mathbf{S}_1$-subproblem in (26) presents itself in the form of

$$\min_{\mathbf{S}_1} \frac{\alpha}{2}\|\mathbf{S}^{(k+1)} - \mathbf{S}_1\|_F^2 + \tau \mathrm{TV_H}(\mathbf{S}_1) \quad \text{s.t.} \quad \mathbf{1}_R^T\mathbf{S}_1 = \mathbf{1}_N^T, \mathbf{S}_1 \geq 0. \tag{34}$$

By comparing (18) and (34), we can use Algorithm 1 to obtain an approximate solution to the unconstrained version of the problem (34), which is given by

$$\mathbf{S}_1^{(k+\frac{1}{2})} = \mathcal{H}_{\frac{\tau}{\alpha},H_m}\left(\mathbf{S}^{(k+1)}\right). \tag{35}$$

Ultimately, the constraint conditions are applied to the $\mathrm{TV_H}$-regularized matrix $\mathbf{S}_1^{(k+\frac{1}{2})}$ via projection operators, yielding the following update formula for $\mathbf{S}_1^{(k+1)}$:

$$\mathbf{S}_1^{(k+1)} = \mathcal{N}\left(\mathcal{P}\left(\mathbf{S}_1^{(k+\frac{1}{2})}\right)\right). \tag{36}$$

Let K_m be the maximum number of outer iterations. The process terminates after K_m iterations. We summarize the alternating minimization approach in Algorithm 2 for solving the L_p-$\mathrm{TV_H}$ HU problem (24).

The computational complexity of Algorithm 2 is analyzed as follows. The per-iteration cost comprises two components: gradient descent updates and projection operators. The gradient descent updates for $\mathbf{A}$ and $\mathbf{S}$ incur $O(LRN)$ complexity. Since $\mathcal{P}$, $\mathcal{N}$, and $\mathcal{G}_{\lambda,p}$ operate element-wise, their complexities are $O(LR)$, $O(RN)$, and $O(RN)$, respectively. The complexity of the $\mathcal{H}_{\frac{\tau}{\alpha},H_m}$ operator has been derived

in Section 2.3 as $O(\overline{h}N)$. Thus, the total per-iteration cost is $O\big((LR+\overline{h})N\big)$. With K_m outer iterations, the overall time complexity is $O\big((LR+\overline{h})K_mN\big)$.

4. **STV-ANet.** In the field of HU, model-based methods are interpretable but inflexible due to complex tuning, while learning-based methods are adaptive but lack explainability. Algorithm unfolding improves interpretability by converting models into networks, but existing networks (e.g., SNMF-Net) ignore spatial information in HSIs. We propose a novel unrolling network STV-ANet for blind HU, which unfolds an alternating minimization algorithm with the L_p-norm and the $\mathrm{TV_H}$ regularization into an NMF-based network. STV-ANet enhances sparsity and spatial characteristics of HSIs and enables automatic parameter learning. In addition, by unrolling the L_p-norm and the $\mathrm{TV_H}$ regularization terms into the network, it suppresses noise and thereby mitigates overfitting.

4.1. **Algorithm unrolling.** In this section, we unroll Algorithm 2 into a deep neural network. The updating formulas for $\mathbf{A}$, $\mathbf{S}$, and $\mathbf{S}_1$ are integrated into the forward propagation network. Leveraging the learning properties of neural networks, we employ iterative formulas as the learners of the network. Within the update formulas, various operators associated with the constraint conditions act as activation functions. Specifically, the non-negative projection operator $\mathcal{P}(\cdot)$ serves as the rectified linear unit (ReLU) function, the normalization operator $\mathcal{N}(\cdot)$ acts as the normalization function, and the GST operator $\mathcal{G}_{\lambda,p}(\cdot)$ functions as the GST activation function. Thus, the entire forward propagation process for the $k+1$-th layer is as follows:

$$\mathbf{A}^{(k+1)} = \mathcal{P}\left(\mathbf{A}^{(k)} - t_1(\mathbf{A}^{(k)}\mathbf{S}^{(k)} - \mathbf{Y})(\mathbf{S}^{(k)})^T\right), \tag{37}$$

$$\mathbf{S}^{(k+1)} = \mathcal{N}\left(\mathcal{P}\left(\mathcal{G}_{\lambda,p}\left(\mathbf{S}^{(k)} - t_2(\mathbf{W}^{(k+1)}(\mathbf{A}^{(k+1)}\mathbf{S}^{(k)} - \mathbf{Y}) + \alpha(\mathbf{S}^{(k)} - \mathbf{S}_1^{(k)})\right)\right)\right), \tag{38}$$

$$\mathbf{S}_1^{(k+1)} = \mathcal{N}\left(\mathcal{P}\left(\mathcal{H}_{\frac{\tau}{\alpha},H_m}\left(\mathbf{S}^{(k+1)}\right)\right)\right). \tag{39}$$

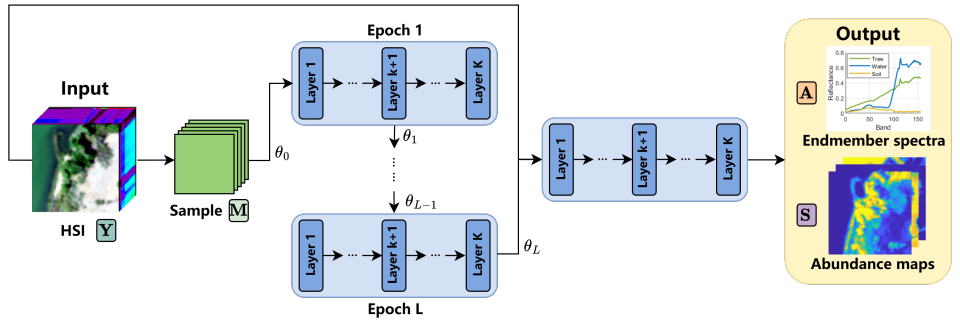
Following [31], we replace $(\mathbf{A}^{(k+1)})^T$ with a learnable weight matrix $\mathbf{W}^{(k+1)} \in \mathbb{R}^{R\times L}$ in (38), to enhance the dynamic optimization capability of the network.

The loss function for STV-ANet is the Euclidean distance between the reconstructed HSI $\hat{\mathbf{Y}}$ and the input HSI $\mathbf{Y}$. Motivated by optimizing the utilization efficiency of internal network resources [26], the output of each layer is incorporated into the loss function. Consequently, for a given training sample set $\mathbf{M} = \{\mathbf{m}_1, \mathbf{m}_2, \ldots, \mathbf{m}_{N_{\text{sample}}}\}$, where $\mathbf{m}_i$, for $i = 1, \ldots, N_{\text{sample}}$, is a row vector with length L, the loss function for the network is defined as

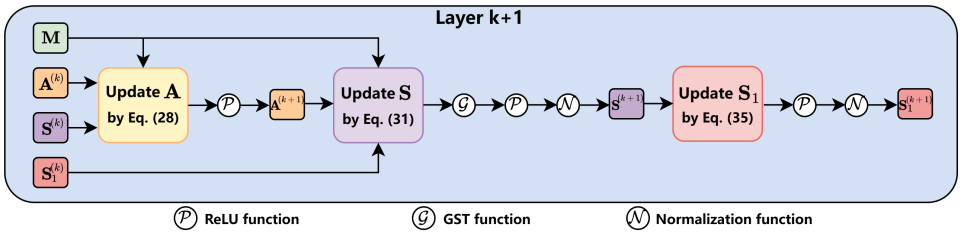
$$\mathcal{L}_\theta = \frac{1}{2K}\sum_{k=0}^{K-1}\left\|\mathbf{M} - \mathbf{A}^{(k+1)}\mathbf{S}^{(k+1)}\right\|_F^2, \tag{40}$$

where $\theta = \{\lambda, p, \alpha, \tau, t_1, t_2, \mathbf{W}\}$ is the set of learnable parameters, K indicates the number of layers in the network (which corresponds to K_m in the model), and $\mathbf{S}^{(k+1)} \in \mathbb{R}^{R\times N_{\text{sample}}}$ represents the abundance matrix for the sample pixels in $\mathbf{M}$ during training.

Backward propagation network is employed to learn the parameters. Since the steps to generate $\mathbf{A}^{(k+\frac{1}{2})}$, $\mathbf{S}^{(k+\frac{1}{2})}$ and $\mathbf{S}_1^{(k+\frac{1}{2})}$, as shown in (28), (31), and (35), are all differentiable, the partial derivatives of the loss function $\mathcal{L}_\theta$ with respect to each parameter in the set θ can be computed through backpropagation across the network.



(a)



(b)

FIGURE 3. Flowcharts of STV-ANet. (a) Network architecture overview. (b) Update rules for forward propagation at layer $k+1$.

4.2. **Network architecture.** The comprehensive update approach of STV-ANet, which encompasses the entire training process and the intricate details of forward propagation, is meticulously depicted in Figure 3. The training process of the network involves the following steps:

Step 1. *Network initialization.* There are two components needed to be addressed: data initialization and parameter initialization. Data initialization refers to extracting initial matrices, including an endmember matrix $\mathbf{A}$ and a corresponding abundance matrix $\mathbf{S}$, from the input HSI $\mathbf{Y}$, and setting the initial value of $\mathbf{S}_1$ to be identical to $\mathbf{S}$. Here, we employ the vertex component analysis (VCA) and fully constrained least squares (FCLS) to achieve this, while other suitable methods can also be utilized. Parameter initialization assigns initial values to the parameters within the parameter set θ to facilitate subsequent network optimization. For further details on the parameter settings, see Section 5.4.

Step 2. *Sample selection.* Randomly extracting some spectra vectors from the input HSI yields the training set $\mathbf{M} = \{\mathbf{m}_1, \mathbf{m}_2, \ldots, \mathbf{m}_{N_{\text{sample}}}\}$, as well as the corresponding parts of the initial matrices $\mathbf{A}$ and $\mathbf{S}$, i.e., $\mathbf{A}^{(0)}$ and $\mathbf{S}^{(0)}$.

Step 3. *Model education.* $\mathbf{A}^{(0)}$, $\mathbf{S}^{(0)}$ and the training set $\mathbf{M}$ are fed into the algorithm-based forward propagation network to generate $\mathbf{A}^{(k+1)}$ and $\mathbf{S}^{(k+1)}$, for $k = 0, \ldots, K-1$.

Step 4. *Parameter optimization.* The stochastic gradient descent algorithm is utilized in the backward propagation network to compute the gradients for updating the model parameters. As the network learns, the parameter set θ is optimized in

each epoch. Repeat steps 3 and 4 until the preset maximum number of network layers is reached.

Step 5. *Results acquisition.* Inputting the initial matrices $\mathbf{A}$, $\mathbf{S}$ and the observed HSI $\mathbf{Y}$ into the trained network results in the estimated matrices $\hat{\mathbf{A}}$ and $\hat{\mathbf{S}}$, and the reconstructed HSI $\hat{\mathbf{Y}} = \hat{\mathbf{A}}\hat{\mathbf{S}}$.

In this architecture, the input HSI is first decomposed into endmembers and abundances, which are then alternately updated through an iterative process and subsequently reconstructed back into an HSI. We note that this procedural flow generally bears certain similar structure to encoder-decoder networks. However, as an unrolling-based HU network, STV-ANet offers greater interpretability compared to conventional encoder-decoder architectures.

5. Experiments.

We implemented the STV-ANet using PyTorch 2.2.1+cu118 within the Python 3.8.19 environment. All experiments were performed on a Windows 10 operating system, leveraging the computational power of an NVIDIA GeForce RTX 4070 GPU.

Several HU experiments, conducted on synthetic and real-world HSIs, showcase the effectiveness of our STV-ANet. The comparative analysis included model-based techniques such as VCA [17], FCLS [9], and TV-RSNMF [8], as well as learning-based approaches like uDAS [20], UnDIP [22], MiSiCNet [21], and the unfolding algorithm SNMF-Net [31]. All the comparative methodologies were rigorously applied with their respective optimal parameter settings to ensure a fair assessment. Notably, VCA and FCLS serve as the initialization for all methods under comparison, except MiSiCNet, which employs SiVM [10] for endmember initialization.

For quantitative analysis, the spectral angle distance (SAD) is employed as a metric to assess the accuracy of endmembers, and the related mathematical formulation is given as

$$\mathrm{SAD}_i = \arccos\left(\frac{(\mathbf{A}^{[i]})^T\hat{\mathbf{A}}^{[i]}}{\|\mathbf{A}^{[i]}\|_2\|\hat{\mathbf{A}}^{[i]}\|_2}\right), \tag{41}$$

where $\hat{\mathbf{A}}^{[i]}$ is the i-th column of $\hat{\mathbf{A}}$, and $\mathbf{A}^{[i]}$ is the reference one, for $i = 1, \ldots, R$. Moreover, the root mean square error (RMSE) is utilized to assess the precision of abundance, which is defined as

$$\mathrm{RMSE}_i = \frac{1}{\sqrt{N}}\|\mathbf{S}_{[i]} - \hat{\mathbf{S}}_{[i]}\|_F, \tag{42}$$

where $\hat{\mathbf{S}}_{[i]}$ and $\mathbf{S}_{[i]}$ are the i-th rows of estimated and ground-truth abundance matrices, respectively. The smaller the values of SAD and RMSE are, the better the unmixing effect is.

Additionally, we introduce the zero-mean Gaussian noise into both synthetic and real-world HSI datasets, with the noise level quantified by the signal-to-noise ratio (SNR) as

$$\mathrm{SNR} = 10\log_{10}\frac{E\left[y^T y\right]}{E\left[e^T e\right]}, \tag{43}$$

where y denotes the observed HSI at the pixel level, e represents the noise within y, and $E[\cdot]$ denotes the expectation operator.

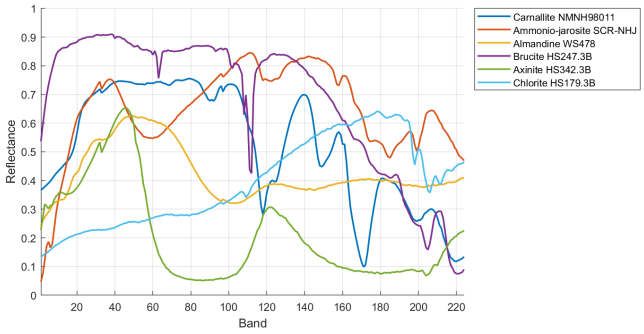
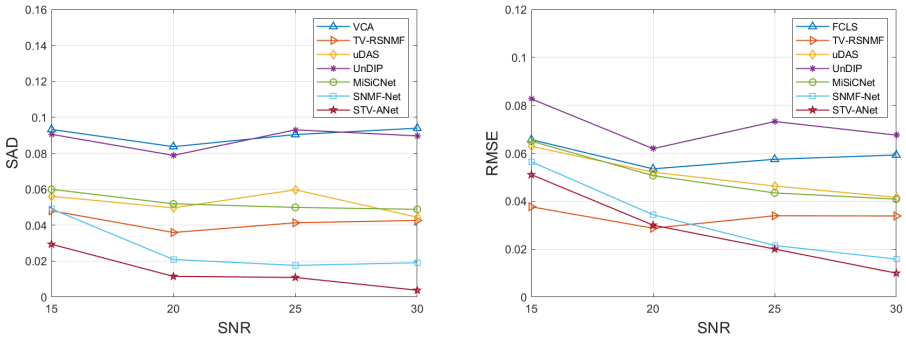


FIGURE 4. Spectral signatures of six endmembers in the synthetic dataset.



(a) SAD (b) RMSE

FIGURE 5. Noise robustness analysis of various algorithms. The performance of STV-ANet and comparative algorithms in terms of (a) SAD and (b) RMSE metrics across different noise levels.

5.1. **Synthetic dataset.** The synthetic dataset consists of 64×64 pixels and 224 bands with wavelengths between 0.38 and 2.5 μm. Six spectral signatures (Carnallite, Ammonio-jarosite, Almandine, Brucite, Axinite, and Chlonte) were selected from the United States Geological Survey (USGS) digital spectral library as endmembers [30]. The signatures of the endmembers are shown in Figure 4.

The performance of STV-ANet in different levels of white Gaussian noise was benchmarked against various cutting-edge algorithms to demonstrate its superiority. Figure 5 illustrates the performance of all compared algorithms on SAD and RMSE at SNR levels of 15, 20, 25, and 30 dB. It is evident that STV-ANet performs exceptionally well in terms of SAD, outperforming other algorithms across all noise levels. In terms of RMSE, it exhibits a slight superiority over SNMF-Net, showing a trend of improvement as the SNR increases.

Regarding endmember curves and abundance maps, our algorithm shows significant improvements. Figure 6 presents the spectral curves of six endmembers within the synthetic dataset with SNR = 25 dB. Notably, the curve derived from the

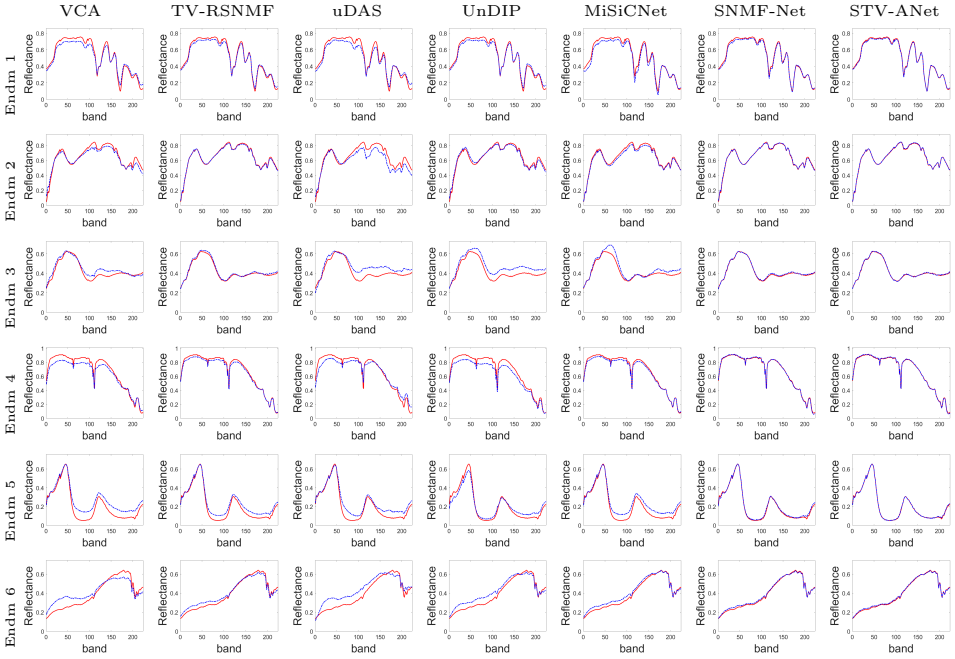


FIGURE 6. Endmember spectral curves comparison for the synthetic dataset with SNR = 25 dB. The blue curves delineate the endmembers extracted by various HU algorithms, whereas the red curves correspond to the ground-truth endmembers.

STV-ANet closely aligns with the ground-truth curve, indicating a superior fit compared to other algorithms. This suggests that the STV-ANet effectively captures the spectral signatures of the endmembers, which is critical for accurate HU. Figure 7 illustrates the STV-ANet's effectiveness in discerning abundances, particularly in areas where they are otherwise indistinguishable from the background. While our TV_{H} regularization reduced the background noise compared to SNMF-Net, the figure still reveals its presence, which may affect the overall accuracy of the abundance maps. We note that this likely stems from the sparsity regularization's tolerance of minor residual values. Further improvements may be achieved through more aggressive denoising regularization.

5.2. **Real-world datasets.** To enhance the credibility and practicality of our study and test the robustness of the proposed network, we conducted HU experiments on two real-world datasets: Samson and Jasper Ridge.

5.2.1. *Samson dataset.* The Samson dataset contains 95×95 pixels and 156 bands ranging from 0.401 to 0.889 μm. There are three endmembers in this dataset: water, soil, and tree.

Table 1 shows the quantization performance of all compared algorithms with respect to SAD, RMSE, and runtime. From Table 1, we see that STV-ANet demonstrates the best performance in terms of mean RMSE and the second-best in mean SAD. This table also records the average running time of all comparison algorithms over five runs. The increased computational cost of STV-ANet, compared with

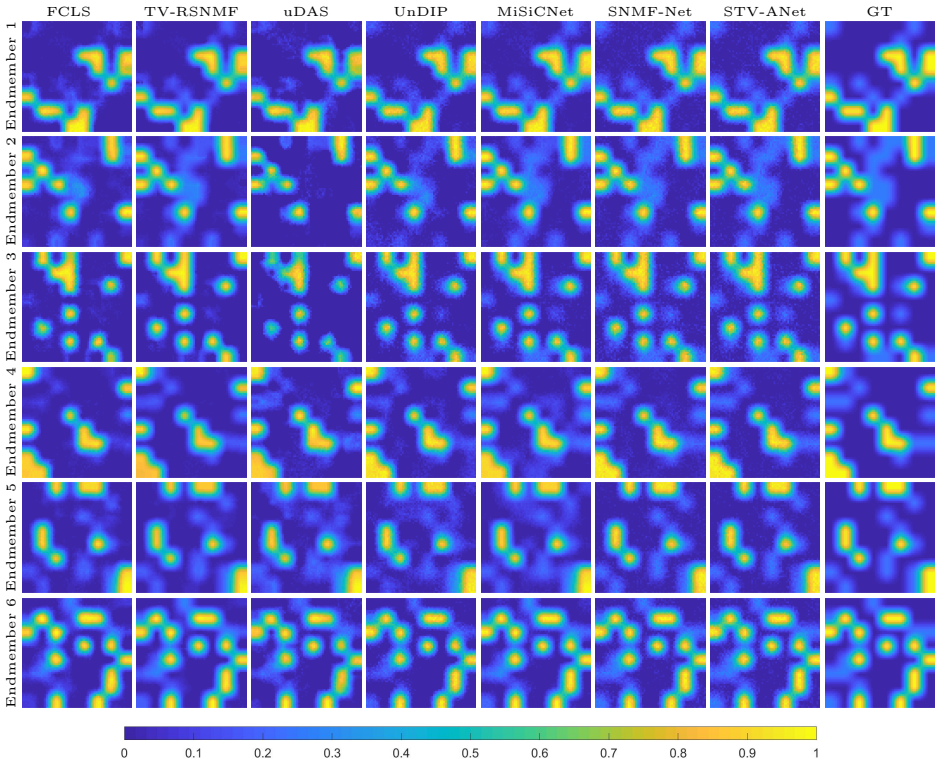


FIGURE 7. Abundance maps estimated by different algorithms for the synthetic dataset with SNR = 25 dB. Here, GT stands for the ground-true abundance map.

traditional and certain deep learning approaches, primarily arises from the inner-loop iterations necessitated by the unrolled TV_H regularization. In comparison, its speed advantage over UnDIP likely arises from the latter involving more iterations that each require complete backpropagation. Figures 8 and 9 present the specific details of the HU results for the Samson dataset in terms of endmember curves and abundance maps. Inspecting Figure 8, our algorithm's endmember curves closely match the true endmembers across almost the entire spectral range, particularly for the water endmember. As observed in Figure 9, the abundance estimates from all algorithms align well with the ground truth at large. However, subtle chromatic discrepancies persist in the abundance maps of nearly all soil endmembers. These deviations likely stem from suboptimal handling of extreme values in the abundance matrices across different algorithms, manifesting as the slightly underestimated abundance values in the yellow regions for the soil endmember in the abundance maps. In practice, these deviations do not alter soil spatial distribution patterns and have limited impact on higher-level applications such as land-use classification, especially when RMSE values are small. An enhanced sparsity regularization may alleviate this issue. Nevertheless, the color discrepancies in our abundance maps are relatively small. This result shows that STV-ANet demonstrates superior performance on this real-world dataset.

TABLE 1. Quantitative results for the Samson dataset. The optimal results are highlighted in bold, whereas the suboptimal results are indicated with underline.

Algorithms		VCA	TV-RSNMF	uDAS	UnDIP	MiSiCNet	SNMF-Net	STV-ANet
SAD	Soil	2.44	2.41	**1.78**	3.99	3.13	<u>2.34</u>	2.35
	Tree	1.15	0.90	<u>0.47</u>	1.46	**0.23**	0.81	0.81
	Water	8.05	7.75	16.47	19.90	5.58	**1.79**	<u>1.93</u>
	Mean	3.88	3.69	6.24	8.45	2.98	**1.65**	<u>1.70</u>
Algorithms		FCLS	TV-RSNMF	uDAS	UnDIP	MiSiCNet	SNMF-Net	STV-ANet
RMSE	Soil	3.34	2.87	4.68	4.86	3.51	<u>2.85</u>	**2.77**
	Tree	2.68	**2.42**	2.92	3.91	3.38	2.49	<u>2.45</u>
	Water	1.26	0.94	9.84	2.13	0.92	<u>0.82</u>	**0.77**
	Mean	2.43	2.08	5.81	3.63	2.60	<u>2.05</u>	**1.99**
Time(s)		**6.08**	24.26	<u>10.06</u>	359.72	47.42	27.88	103.32

[1] All values of SAD and RMSE are multiplied by 100.

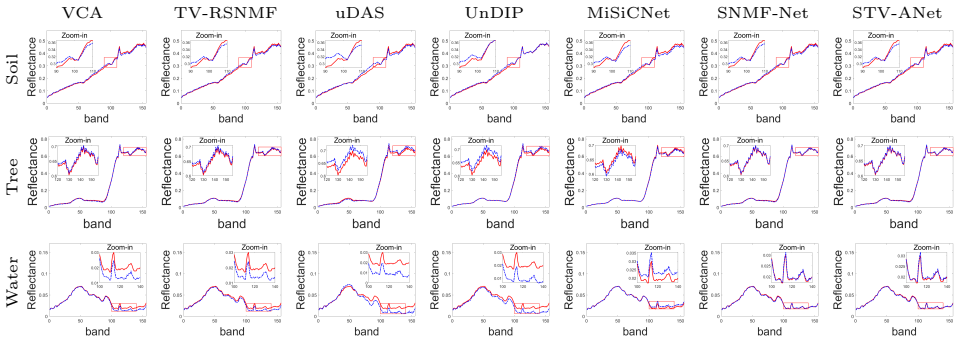


FIGURE 8. Endmember spectral curves comparison for the Samson dataset. The blue curves delineate the endmembers extracted by various HU algorithms, whereas the red curves correspond to the ground-truth endmembers.

5.2.2. *Jasper Ridge dataset.* The Jasper Ridge dataset has 100×100 pixels with 198 bands, covering wavelength from 0.38 to 2.5 μm. It contains four endmembers including tree, water, soil, and road.

Table 2 presents the experimental results of SAD, RMSE and runtime for all algorithms under consideration. Our STV-ANet performs best on the water endmember in both SAD and RMSE. Moreover, the smallest mean SAD and the second smallest mean RMSE show that our algorithm is more dominant in comprehensive strength. The runtime patterns of various algorithms on this dataset are consistent with those observed on the Samson dataset. These results demonstrate that our approach achieves a favorable balance between efficacy and efficiency. Figures 10 and 11 display the endmember curves and abundance maps obtained from the the Jasper Ridge dataset, respectively. In Figure 10, the proposed algorithm still demonstrates superior performance in the water endmember relative to other algorithms, and it accurately captures the nuances of all the endmember curves. Figure 11 shows that

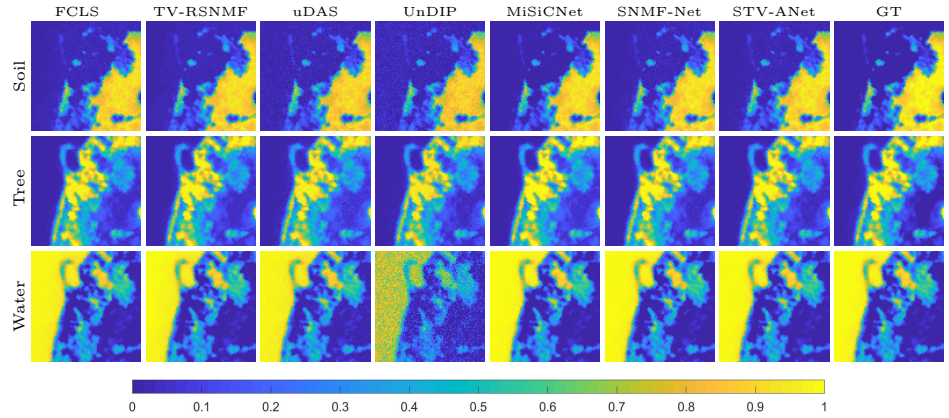


FIGURE 9. Abundance maps estimated by different algorithms for the Samson dataset.

TABLE 2. Quantitative results for the Jasper Ridge dataset. The optimal results are highlighted in bold, whereas the suboptimal results are indicated with underline.

Algorithms		VCA	TV-RSNMF	uDAS	UnDIP	MiSiCNet	SNMF-Net	STV-ANet
SAD	Tree	2.21	2.03	2.34	3.75	**0.42**	1.98	<u>1.86</u>
	Water	9.11	6.89	29.84	20.65	6.09	<u>3.63</u>	**3.00**
	Soil	<u>0.73</u>	**0.72**	0.92	1.66	7.50	0.80	0.74
	Road	1.13	**0.91**	<u>1.04</u>	1.85	2.19	1.07	1.06
	Mean	3.29	2.63	8.54	6.98	4.05	<u>1.87</u>	**1.67**
Algorithms		FCLS	TV-RSNMF	uDAS	UnDIP	MiSiCNet	SNMF-Net	STV-ANet
RMSE	Tree	2.90	<u>2.40</u>	3.63	2.75	2.60	2.59	**2.33**
	Water	1.27	<u>0.88</u>	7.78	2.01	0.93	0.89	**0.68**
	Soil	2.98	**2.38**	5.06	5.15	8.96	2.87	<u>2.83</u>
	Road	2.00	**1.53**	3.78	5.02	10.21	1.89	<u>1.80</u>
	Mean	2.29	**1.80**	5.06	3.73	5.67	2.06	<u>1.91</u>
Time(s)		**12.02**	33.11	<u>17.20</u>	495.03	59.76	30.67	180.91

[1] All values of SAD and RMSE are multiplied by 100.

most of the estimated abundance maps are close to the ground truth. However, our algorithm provides better detail processing for the background (blue areas) of each endmember, demonstrating the effectiveness of the TV_H regularization term in noise removal.

5.3. **SORT metric and supplementary results.** In this subsection, we introduce a sorting metric, named as SORT, which facilitates the comparison of unmixing results and guides the selection of network parameters. During the experiments, we observed that SAD and RMSE are not always aligned. There are cases where SAD is very small but RMSE is relative large, or vice versa, which are not the outcomes we desire. To address these, we propose SORT to balance SAD and RMSE, which

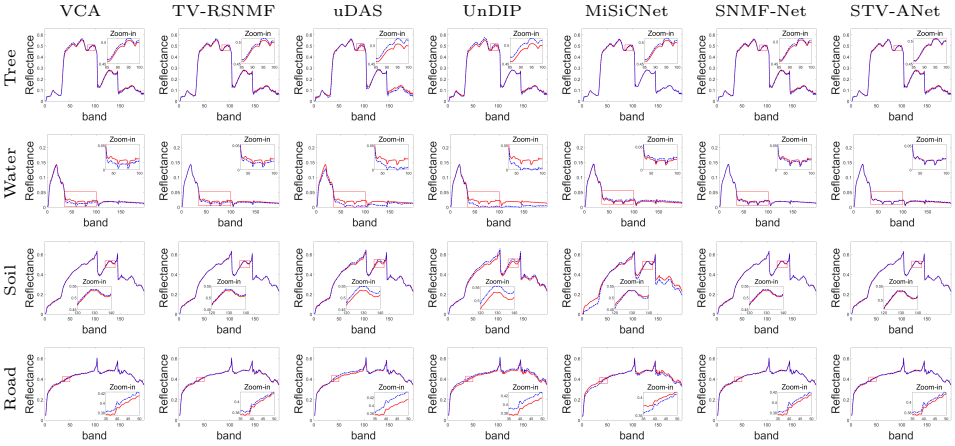


FIGURE 10. Endmember spectral curves comparison for the Jasper Ridge dataset. The blue curves delineate the endmembers extracted by various HU algorithms, whereas the red curves correspond to the ground-truth endmembers.

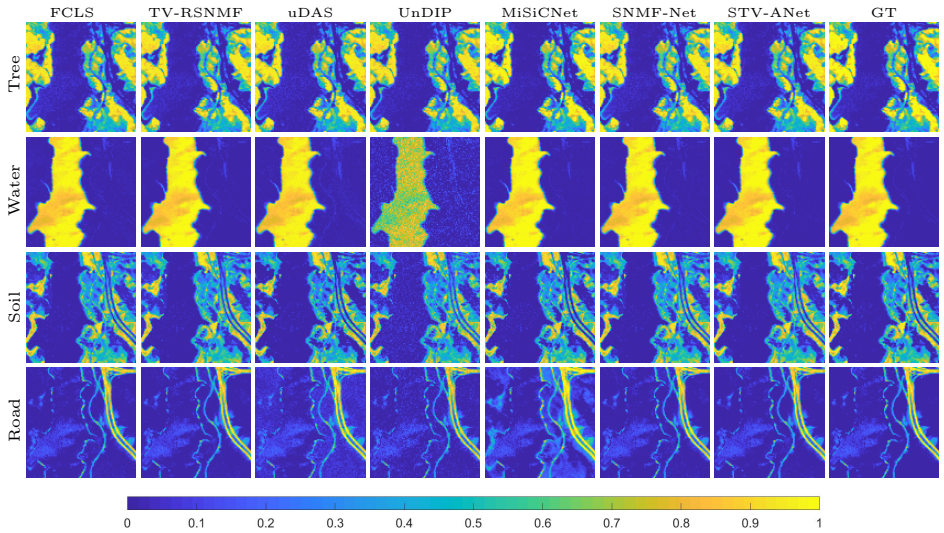


FIGURE 11. Abundance maps estimated by different algorithms for the Jasper Ridge dataset.

is defined as

$$\mathrm{SORT} = \frac{\mathrm{SAD}_{\mathrm{norm}} + \mathrm{RMSE}_{\mathrm{norm}}}{2}, \tag{44}$$

where $\mathrm{SAD}_{\mathrm{norm}}$ and $\mathrm{RMSE}_{\mathrm{norm}}$ represent the normalized values of SAD and RMSE for each HU result relative to all compared results, respectively. These results can be from the same method with different configurations or different methods, but should be based on the same dataset. Taking the case of different methods as an

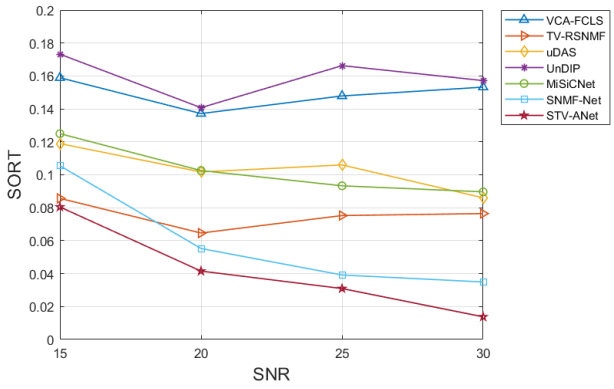


FIGURE 12. The performance of STV-ANet and comparative algorithms, evaluated using the SORT metric, across different noise levels for the synthetic dataset.

example, $\mathrm{SAD}_{\mathrm{norm}}$ and $\mathrm{RMSE}_{\mathrm{norm}}$ can be calculated by

$$\mathrm{SAD}_{\mathrm{norm}} = \frac{\mathrm{SAD} - \mathrm{SAD}_{\mathrm{min}}}{\mathrm{SAD}_{\mathrm{max}} - \mathrm{SAD}_{\mathrm{min}}}, \tag{45}$$

$$\mathrm{RMSE}_{\mathrm{norm}} = \frac{\mathrm{RMSE} - \mathrm{RMSE}_{\mathrm{min}}}{\mathrm{RMSE}_{\mathrm{max}} - \mathrm{RMSE}_{\mathrm{min}}}, \tag{46}$$

where SAD represents the SAD value obtained by each method, $\mathrm{SAD}_{\mathrm{min}}$ is the minimum value, and $\mathrm{SAD}_{\mathrm{max}}$ is the maximum value among all compared methods. The same applies to RMSE.

The benefits of normalization include eliminating the influence of different scales, which makes SAD and RMSE comparable. We consider SAD and RMSE to be equally important, and thus assign both of them a weight of 0.5, making SORT the average value of $\mathrm{SAD}_{\mathrm{norm}}$ and $\mathrm{RMSE}_{\mathrm{norm}}$.

When comparing the unmixing results of different algorithms using SORT, Figure 12, Table 3, and Table 4 are provided as supplementary experimental results for the three datasets. Figure 12 shows the performance of various comparative algorithms on the SORT metric across different SNR levels for the synthetic dataset. The bottom line in the figure corresponds to STV-ANet, indicating that it achieves the best overall performance. Table 3 presents the normalized SAD and RMSE values for each algorithm on the Samson dataset, along with their mean value, SORT. As shown in this table, the proposed STV-ANet has the smallest SORT value, indicating that it performs the best when considering both SAD and RMSE. Table 4 is for the Jasper Ridge dataset. In this table, STV-ANet ranks highly among the top performers in both normalized metrics, which makes it the best performer in SORT. While TV-RSNMF outperforms in terms of RMSE, its performance in SAD is not prominent enough to surpass SNMF-Net and STV-ANet.

5.4. **Sensitivity analysis.** In this subsection, we primarily explore the factors that affect network performance, including analysis and initialization of network parameters, the ordering of optimization updates, and systematic ablation studies.

TABLE 3. Normalized SAD, RMSE, and SORT values for each algorithm on the Samson dataset. The optimal results are highlighted in bold, whereas the suboptimal results are indicated with underline.

Algorithms	V&F	TV-RSNMF	uDAS	UnDIP	MiSiCNet	SNMF-Net	STV-ANet
SAD_{norm}	32.86	30.00	67.51	100.00	19.60	**0.00**	<u>0.72</u>
$RMSE_{norm}$	11.26	2.06	100.00	42.86	15.86	<u>1.38</u>	**0.00**
SORT	22.06	16.02	83.76	71.43	17.73	<u>0.69</u>	**0.36**

[1] All values in this table are multiplied by 100.
[2] V&F means VCA and FCLS.

TABLE 4. Normalized SAD, RMSE, and SORT values for each algorithm on the Jasper Ridge dataset. The optimal results are highlighted in bold, whereas the suboptimal results are indicated with underline.

Algorithms	V&F	TV-RSNMF	uDAS	UnDIP	MiSiCNet	SNMF-Net	STV-ANet
SAD_{norm}	23.70	14.10	100.00	77.31	34.69	<u>2.98</u>	**0.00**
$RMSE_{norm}$	12.63	**0.00**	84.15	49.88	100.00	6.79	<u>2.90</u>
SORT	18.17	7.05	92.07	63.59	67.34	<u>4.88</u>	**1.45**

[1] All values in this table are multiplied by 100.
[2] V&F means VCA and FCLS.

TABLE 5. Experimental setup of parameters.

Datasets			Synthetic dataset (SNR)				Samson dataset	Jasper Ridge dataset
			15 dB	20 dB	25 dB	30 dB		
Parameters	Turning	K	30	30	30	30	9	15
		lr_1	0.19	0.51	0.69	1.00	0.05	0.02
		lr_2	10^{-6}	10^{-6}	10^{-6}	10^{-6}	10^{-8}	10^{-7}
	Learnable		$\alpha = 10^4$, $\tau = 0.5$, t_1 by (47), t_2 by (48), $\lambda = 0.01t_2$, $p = 0.5$, $\mathbf{W}$ by (49)					
	Fixed		$H_m = 2$, $N_{\text{sample}} = 500$, $L = 500$					

• *Parameter analysis and initialization.* We maintain a consistent end-to-end network architecture across all experiments, while optimizing hyperparameters separately for each dataset and noise level. All parameter settings for the datasets are consolidated in Table 5. In our experiments, we categorize the parameters into three distinct groups:

1) Tuning parameters. We first introduce the tuning parameters, including the number of network layers K, as well as the learning rates lr_1 and lr_2 for abundances and endmembers, respectively. These parameters have a significant impact on the HU results but lack a discernible pattern. Consequently, a search over their values is required to achieve optimal outcomes.

When the number of network layers is small, the network cannot learn from the data adequately. However, as the number of layers increases, so does the computational load. Hence, we set the maximum value of K to 30. Without considering computational power, a higher number of layers might yield superior performance, especially for the synthetic dataset. We also provide experimental sets for the two learning rates to serve as references for subsequent experiments. Typically, lr_1 is set to a larger value to ensure fast convergence, while lr_2 is set to a smaller value to enable fine-tuning and maintain stability. Specifically, lr_1 is chosen from the interval $(0, 2.5]$, while lr_2 is selected from the set $\{10^{-8}, 10^{-7}, 10^{-6}\}$.

2) Learnable parameters. We next introduce the learnable parameters, which are optimized by the network during training rather than manually tuned. This category includes the regularization parameters α, τ and λ, the L_p-norm parameter p, the step sizes t_1 and t_2, and the network weight matrix $\mathbf{W}$. These parameters are iteratively updated through the network learning process but require initial settings before the learning process begins. Since they are trained by network subsequently, their initial values usually have a minimal effect on the results.

We first pay attention to the regularization parameters α and τ. For the synthetic dataset with SNR = 25 dB, while keeping all other settings constant, we vary α over $\{1, 10, ..., 10^6\}$ and τ over $\{10^{-4}, 10^{-3}, 0.01, 0.1, 0.5, 1\}$. In Figure 13, all metrics remain relatively stable when α ranges from 1 to 10^4, followed by a sharp increase when α exceeds 10^4. In contrast, The impact of τ variations on the outcomes is not readily apparent. To make an informed selection of τ, we fix α at 10^4. Figure 14 presents the SAD and RMSE results for different values of τ under this setting. For aesthetic reasons, the case for $\tau = 1$ is omitted from this figure. The comparison of the line shapes between normalized SAD and RMSE in Figure 15 with their original counterparts in Figure 14 reveals that the normalization does not alter the ranking of the results. As depicted in Figure 15, SORT effectively balances SAD and RMSE, achieving optimal performance at $\alpha = 10^4$ and $\tau = 0.5$, which also is the best result across the entire grid.

The initialization settings for the parameters t_1, t_2, λ, p and $\mathbf{W}$ follow reference [31] and are briefly described below. We adopt an adaptive initialization scheme for the step sizes,

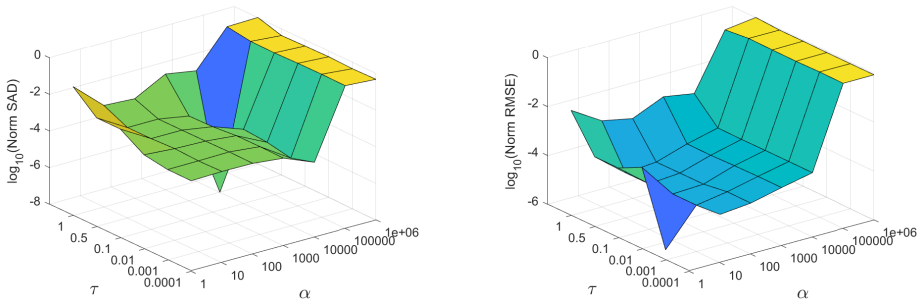
$$t_1 = \frac{1}{\|\mathbf{S}^{(0)}(\mathbf{S}^{(0)})^T\|_2 + \epsilon}, \tag{47}$$

$$t_2 = \frac{1}{\|(\mathbf{A}^{(0)})^T\mathbf{A}^{(0)}\|_2 + \epsilon}, \tag{48}$$

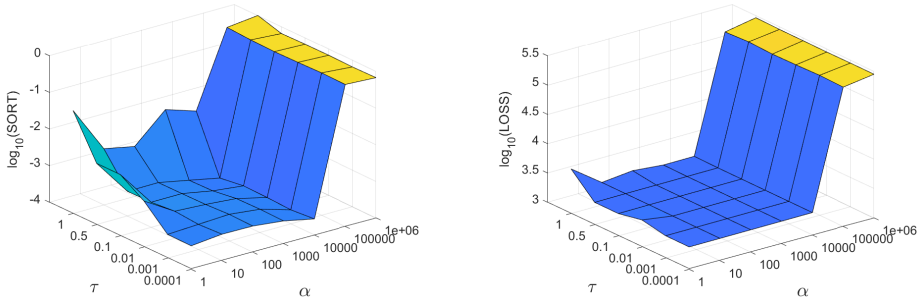
where ϵ is a small positive constant set to 0.01. Furthermore, λ is initialized to $0.01t_2$ and p to 0.5. We also conduct sensitivity analysis for the parameters λ and p on the synthetic dataset at SNR = 25 dB, while fixing all other parameters at their optimal values. Specifically, we varied the initialization values of λ over $\{10^{-7}, 10^{-6}, \ldots, 10^{-3}\}$ and p over $\{0.1, 0.2, \ldots, 1.0\}$, and compared these with our parameter setting. As illustrated in Figure 16, both λ and p achieve optimal SORT performance precisely at our parameter setting. Finally, we precompute $\mathbf{W}$ by solving the following problem:

$$\arg \min_{\mathbf{W} \in \mathbb{R}^{R \times L}} \quad \|\mathbf{W}^T\mathbf{D}^*\|_F^2 \quad \text{s.t.} \quad \mathbf{W}^T\mathbf{D}^* = \mathbf{I}, \tag{49}$$

where $\mathbf{D}^*$ is a low-coherence dictionary (see [31] for more details).

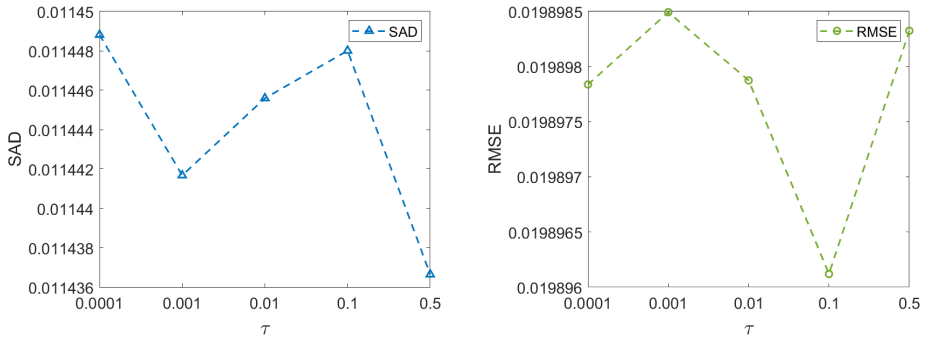


(a) SAD (b) RMSE



(c) SORT (d) LOSS

FIGURE 13. Initialization strategy for the parameters α and τ. This figure illustrates the variation of four metrics — (a) SAD, (b) RMSE, (c) SORT, and (d) LOSS — across a grid defined by the parameters α and τ, using the synthetic data at an SNR of 25 dB.



(a) SAD (b) RMSE

FIGURE 14. Performance in terms of SAD and RMSE for varying τ with α fixed at 10^4.

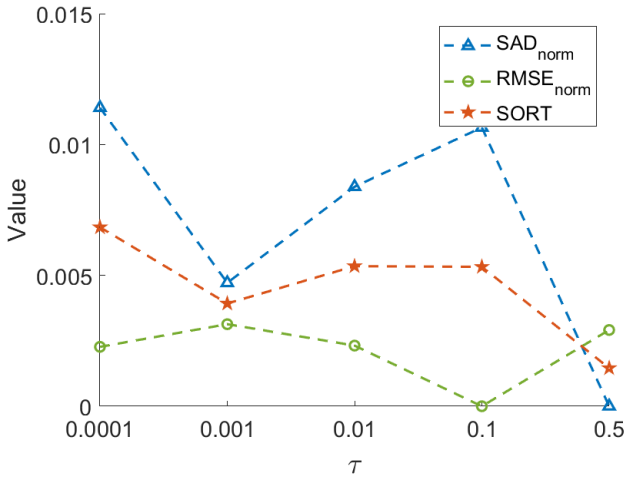
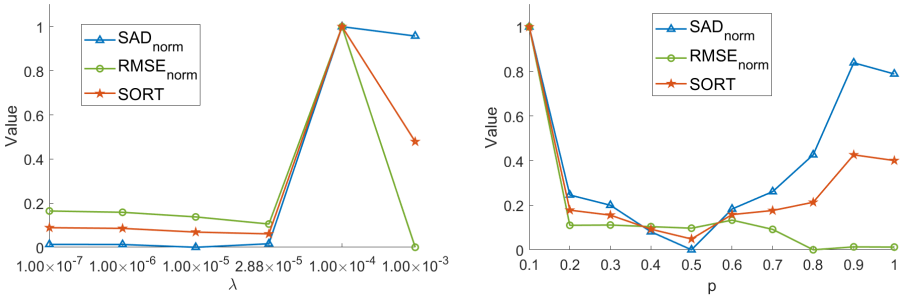


FIGURE 15. Evaluation of τ using the normalized SAD and RMSE values and the composite SORT indicator.



(a) λ (b) p

FIGURE 16. Sensitivity analysis of the initialization values for λ and p. Note that the proposed parameter setting is $\lambda = 0.01t_2 = 2.88 \times 10^{-5}$ and $p = 0.5$, where t_2 is calculated by (48).

3) Fixed parameters. In our experiments, several parameters are kept fixed, including the maximum inner iteration count H_m, the sample shape $(m, n)_{sample}$ for the $\mathrm{TV_H}$ regularization and the epoch number L for the network. However, considering their potential influence on the results, we also incorporate them into our analysis.

In our experiments, we set $H_m = 2$, $(m, n)_{\mathrm{sample}} = \left(\frac{N_{\mathrm{sample}}}{10}, 10\right)$, where $N_{\mathrm{sample}} = 500$, and $L = 500$. The specific shape of the training sample was picked for convenience. Through these experiments, we observed that the shape of the training sample does not affect the results, indicating that it can be any factor pair of N_{sample}.

These tests were conducted on the synthetic data with an SNR of 25 dB. It should be noted that the inner iterations terminate when either the maximum

TABLE 6. Memory consumption under varying maximum inner iteration counts and precision levels.

	H_m	CPU(MB)	GPU(MB)	Avg Iter	Max Iter	SAD	RMSE
$\epsilon_{in} = 10^{-4}$	1	10741.02	893.04	1.00	1	1.0481	1.9421
	2	16114.16	1351.25	1.50	2	1.0475	1.9426
	3	17951.64	1507.95	1.67	3	1.0475	1.9425
	100	21926.52	2408.92	2.00	3	1.0475	1.9425
	H_m	CPU(MB)	GPU(MB)	Avg Iter	Max Iter	SAD	RMSE
$\epsilon_{in} = 10^{-32}$	1	16167.85	4793.97	1.00	1	1.0481	1.9421
	2	24071.79	7166.37	1.50	2	1.0475	1.9426
	3	27069.80	8068.59	1.69	3	1.0473	1.9427
	100	30299.77	8999.64	1.88	16	1.0483	1.9429

[1] CPU and GPU represent the average memory usage per execution of the FGP module.
[2] Avg Iter represents the average number of iterations under the corresponding H_m.
[3] Max Iter indicates the maximum number of iterations reached in the experiments.
[4] All values of SAD and RMSE are multiplied by 100.

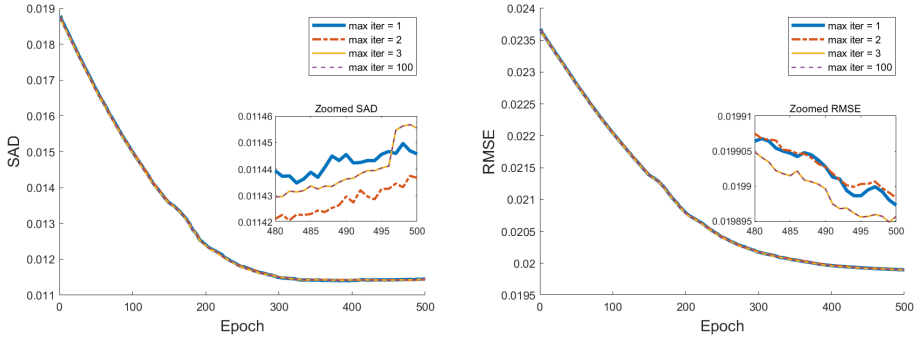
TABLE 7. Study on the sequence of the model updates in algorithmic performance.

Update Order	SAD	RMSE	SAD_{norm}	$\text{RMSE}_{\text{norm}}$	SORT
$\mathbf{S}$, $\mathbf{S}_1$, $\mathbf{A}$	**1.05**	1.94	**0.00**	38.14	**19.07**
$\mathbf{A}$, $\mathbf{S}$, $\mathbf{S}_1$	1.13	**1.94**	100.00	**0.00**	50.00
$\mathbf{S}_1$, $\mathbf{A}$, $\mathbf{S}$	1.06	1.94	15.08	100.00	57.54

[1] All values in this table are multiplied by 100.

inner iteration count is reached or the prescribed precision is satisfied. We evaluated memory consumption across comparative experiments using iteration count $H_m \in \{1, 2, 3, 100\}$ and precision threshold $\epsilon_{in} \in \{10^{-4}, 10^{-32}\}$ (with the latter approximating zero). The experimental results are presented in Table 6. We see from Table 6 that, although CPU and GPU expenditure increases with higher precision, the consequent impact on SAD and RMSE metrics is marginal. Therefore, a precision of 10^{-4} is sufficient. For the maximum inner iteration count H_m, we set $H_m = 1, 2, 3, 100$, while keeping the other conditions unchanged. The results show that when $H_m \geq 3$, STV-ANet exhibits the same behavior in terms of both SAD and RMSE, as evidenced by the coinciding lines for $H_m = 3$ and $H_m = 100$ in Figure 17. This indicates that the value of H_m has little impact on the final results. Figure 17a shows that when $H_m = 2$, SAD performs the best, whereas in Figure 17b, it is when $H_m \geq 3$. Furthermore, observing that the average iteration counts in Table 6 do not exceed 2, we then choose $H_m = 2$ in STV-ANet to achieve better SAD performance and reduce computational load. For the convergence analysis, Figure 17 shows that after 300 epochs, the SAD tends to plateau, while the RMSE continues to show a slight downward trend.

• *Update order.* The sequence in which we update $\mathbf{A}$, $\mathbf{S}$, and $\mathbf{S}_1$ is crucial due to their interdependencies. Table 7 compares the SAD, RMSE, and SORT metrics across three update orders. Notably, the order $\mathbf{S}$, $\mathbf{S}_1$, $\mathbf{A}$ yields the optimal result in terms of SORT, which we adopt for our iterative approach.



(a) SAD (b) RMSE

FIGURE 17. The influence of the maximum number of the internal loops on the SAD and RMSE at each epoch (counting both inner and outer loops). Here, max iter corresponds to the maximum inner iteration count H_m, and the zoomed windows display the results of the last 20 iterations.

• *Ablation study.* The performance comparison of SNMF-Net and STV-ANet highlights that incorporating the $\mathrm{TV_H}$ regularization consistently improves the efficiency of HU, regardless of whether the datasets are synthetic (as shown in Figure 5) or real-world (as detailed in Tables 1 and 2). Additionally, the results in the SORT metric, presented in Figure 12 and Tables 3 and 4, further confirm the advantages of introducing the TV regularization. This demonstrates that the inclusion of the $\mathrm{TV_H}$-based regularization not only enhances the overall performance but also ensures robustness across diverse experimental settings.

6. **Conclusions.** In this article, we introduce a novel alternating minimization algorithm for tackling the unmixing problem that incorporates the L_p-norm and the $\mathrm{TV_H}$ regularization terms and extend this approach to adaptive learning within an NMF deep neural network, termed as STV-ANet. Specifically, in terms of the model, for the L_p-$\mathrm{TV_H}$ HU problem, a novel algorithm based on the gradient descent, the projection and the thresholding operators is employed to solve three subproblems iteratively. By using only one auxiliary variable, this algorithm enhances computational efficiency. On the network front, the proposed algorithm is unrolled into the network via backpropagation, offering the advantage of high interpretability and the ability to leverage both model priors and data features. The learning of parameters through the network enables the model to automatically adjust its parameters to adapt to the data, thereby optimizing performance. Extensive experiments conducted on both synthetic and real-world datasets demonstrate that the proposed STV-ANet achieves superior performance in HU, outperforming several state-of-the-art algorithms. We also note that as the numbers of network layers and internal iterations increase, the computational efficiency of our approach may decrease, which is an area that requires future optimization and improvement.

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

REFERENCES

[1] A. Beck and M. Teboulle, Fast gradient-based algorithms for constrained total variation image denoising and deblurring problems, *IEEE Trans. Image Process.*, **18** (2009), 2419-2434.

[2] J. S. Bhatt and M. V. Joshi, Deep learning in hyperspectral unmixing: A review, *Proceedings of the IEEE International Geoscience and Remote Sensing Symposium*, Virtual Symposium, 2020, 2189-2192.

[3] Y.-B. Cai and J. Huang, A subspace-based total variation regularization for sparse hyperspectral image unmixing, *Int. J. Remote Sens.*, **46** (2025), 5673-5694.

[4] P. H. Calamai and J. J. Moré, Projected gradient methods for linearly constrained problems, *Math. Program.*, **39** (1987), 93-116.

[5] S. Chitnis, K. Mantripragada and F. Z. Qureshi, SpACNN-LDVAE: Spatial attention convolutional latent Dirichlet variational autoencoder for hyperspectral pixel unmixing, *Proceedings of the IEEE International Geoscience and Remote Sensing Symposium*, Athens, Greece, 2024, 7714-7719.

[6] C. Cui, X. Wang, S. Wang, L. Zhang and Y. Zhong, Unrolling nonnegative matrix factorization with group sparsity for blind hyperspectral unmixing, *IEEE Trans. Geosci. Remote Sens.*, **61** (2023), 5516712.

[7] Z. Guo and S. Osher, Template matching via l_1 minimization and its application to hyperspectral data, *Inverse Probl. Imaging*, **5** (2011), 19-35.

[8] W. He, H. Zhang and L. Zhang, Total variation regularized reweighted sparse nonnegative matrix factorization for hyperspectral unmixing, *IEEE Trans. Geosci. Remote Sens.*, **55** (2017), 3909-3921.

[9] 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.*, **39** (2001), 529-545.

[10] R. Heylen, D. Burazerovic and P. Scheunders, Fully constrained least squares spectral unmixing by simplex projection, *IEEE Trans. Geosci. Remote Sens.*, **49** (2011), 4112-4122.

[11] J. Huang, K.-H. Chen, J.-J. Wang and W. Yan, Hyperspectral image denoising via weighted double sparsity total variation and low-rank representation, *Inverse Probl. Imaging*, **18** (2024), 1142-1170.

[12] J. Huang, T.-Z. Huang, L.-J. Deng and X.-L. Zhao, Joint-sparse-blocks and low-rank representation for hyperspectral unmixing, *IEEE Trans. Geosci. Remote Sens.*, **57** (2019), 2419-2438.

[13] M.-D. Iordache, J. M. Bioucas-Dias and A. Plaza, Total variation spatial regularization for sparse hyperspectral unmixing, *IEEE Trans. Geosci. Remote Sens.*, **50** (2012), 4484-4502.

[14] Y. Jiang, Q. Ding, Y. G. Wang, P. Liò and X. Zhang, Vision graph U-Net: Geometric learning enhanced encoder for medical image segmentation and restoration, *Inverse Probl. Imaging*, **18** (2024), 672-689.

[15] Y. M. Jung, T. Jeong and S. Yun, Non-convex TV denoising corrupted by impulse noise, *Inverse Probl. Imaging*, **11** (2017), 689-702.

[16] V. Monga, Y. Li and Y. C. Eldar, Algorithm unrolling: Interpretable, efficient deep learning for signal and image processing, *IEEE Signal Proc. Mag.*, **38** (2021), 18-44.

[17] J. M. P. Nascimento and J. M. B. Dias, Vertex component analysis: A fast algorithm to unmix hyperspectral data, *IEEE Trans. Geosci. Remote Sens.*, **43** (2005), 898-910.

[18] B. Palsson, M. O. Ulfarsson and J. R. Sveinsson, Convolutional autoencoder for spectral-spatial hyperspectral unmixing, *IEEE Trans. Geosci. Remote Sens.*, **59** (2021), 535-549.

[19] Y. Qian, S. Jia, J. Zhou and A. Robles-Kelly, Hyperspectral unmixing via $L_{1/2}$ sparsity-constrained nonnegative matrix factorization, *IEEE Trans. Geosci. Remote Sens.*, **49** (2011), 4282-4297.

[20] Y. Qu and H. Qi, uDAS: An untied denoising autoencoder with sparsity for spectral unmixing, *IEEE Trans. Geosci. Remote Sens.*, **57** (2019), 1698-1712.

[21] B. Rasti, B. Koirala, P. Scheunders and J. Chanussot, MiSiCNet: Minimum simplex convolutional network for deep hyperspectral unmixing, *IEEE Trans. Geosci. Remote Sens.*, **60** (2022), 5522815.

[22] B. Rasti, B. Koirala, P. Scheunders and P. Ghamisi, UnDIP: Hyperspectral unmixing using deep image prior, *IEEE Trans. Geosci. Remote Sens.*, **60** (2022), 5504615.

[23] S. Shu, T.-Z. Huang and J. Huang, Dual-channel enhanced decoder network for blind hyperspectral unmixing, *IEEE Geosci. Remote Sens. Lett.*, **21** (2024), 5500405.

[24] Y. Su, J. Li, A. Plaza, A. Marinoni, P. Gamba and S. Chakravortty, DAEN: Deep autoencoder networks for hyperspectral unmixing, *IEEE Trans. Geosci. Remote Sens.*, **57** (2019), 4309-4321.
[25] Y. Sun and Y. Sun, An optimized TV regularization algorithm for image reconstruction of co-planar array capacitive sensor, *IEEE Trans. Instrum. Meas.*, **74** (2025), 5008009.
[26] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke and A. Rabinovich, Going deeper with convolutions, *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, Boston, MA, USA, 2015, 1-9.
[27] J.-J. Wang, T.-Z. Huang, J. Huang, H.-X. Dou, L.-J. Deng and X.-L. Zhao, Row-sparsity spectral unmixing via total variation, *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.*, **12** (2019), 5009-5022.
[28] R. E. Warren and S. J. Osher, Hyperspectral unmixing by the alternating direction method of multipliers, *Inverse Probl. Imaging*, **9** (2015), 917-933.
[29] L. Wu, J. Huang and Z.-Y. Zhu, Tucker tensor decomposition with rank estimation for sparse hyperspectral unmixing, *Int. J. Remote Sens.*, **45** (2024), 3992-4022.
[30] F. Xiong, Y. Qian, J. Zhou and Y. Y. Tang, Hyperspectral unmixing via nonnegative tensor factorization, *IEEE Trans. Geosci. Remote Sens.*, **57** (2019), 2341-2357.
[31] F. Xiong, J. Zhou, S. Tao, J. Lu and Y. Qian, SNMF-Net: Learning a deep alternating neural network for hyperspectral unmixing, *IEEE Trans. Geosci. Remote Sens.*, **60** (2022), 5510816.
[32] Z. Yang, G. Zhou, S. Xie, S. Ding, J.-M. Yang and J. Zhang, Blind spectral unmixing based on sparse nonnegative matrix factorization, *IEEE Trans. Image Process.*, **20** (2011), 1112-1125.
[33] S. Zhang, J. Li, H.-C. Li, C. Deng and A. Plaza, Spectral-spatial weighted sparse regression for hyperspectral image unmixing, *IEEE Trans. Geosci. Remote Sens.*, **56** (2018), 3265-3276.
[34] H. Zheng, Y. Lou, G. Tian and C. Wang, A scale-invariant relaxation in low-rank tensor recovery with an application to tensor completion, *SIAM J. Imaging Sci.*, **17** (2024), 756-783.
[35] Z.-Y. Zhu, T.-Z. Huang and J. Huang, Projection subspace based low-rank representation for sparse hyperspectral unmixing, *Appl. Math. Model.*, **125** (2024), 463-481.
[36] Z.-Y. Zhu, T.-Z. Huang, J. Huang and L. Wu, Tensor singular value decomposition and low-rank representation for hyperspectral image unmixing, *Signal Process.*, **230** (2025), 109799.
[37] L. Zhuang, C.-H. Lin, M. A. T. Figueiredo and J. M. Bioucas-Dias, Regularization parameter selection in minimum volume hyperspectral unmixing, *IEEE Trans. Geosci. Remote Sens.*, **57** (2019), 9858-9877.
[38] W. Zuo, D. Meng, L. Zhang, X. Feng and D. Zhang, A generalized iterated shrinkage algorithm for non-convex sparse coding, *Proceedings of the IEEE International Conference on Computer Vision*, Sydney, Australia, 2013, 217-224.

Received April 8, 2025; revised February 10, 2026; early access April 7, 2026.