Deep Autoencoder for Hyperspectral Unmixing via Global-Local Smoothing

Xia Xu, Xinyu Song, Tao Li, Zhenwei Shi and Bin Pan

Abstract

Hyperspectral unmixing is to decompose the mixed pixels into pure spectral signatures (endmembers) and their proportions (abundances). Recently, deep learning based methods have been applied to enhance the representation ability of unmixing models by extracting joint spatial-spectral characteristics of the hyperspectral data. However, most deep learning based unmixing methods usually conduct global smoothing by convolutions on the whole hyperspectral imagery, which may ignore the variations within the imagery and result in over-smoothing. In this paper, we propose a deep network for hyperspectral unmixing based on a new Global-Local smoothing Autoencoder (GLA). GLA is an unsupervised model which aims at exploring the local homogeneity and the global self-similarity of hyperspectral imagery. The proposed GLA network mainly include two modules: a Local Continuous conditional random field Smoothing (LCS) module and a Global Recurrent Smoothing (GRS) module. In LCS, we propose a conditional random field based smoothing strategy to describe the joint spatial-spectral information within a local homogeneity region, which also reduces the risk of abundance maps boundary blurry. In GRS, we follow the self-similarity assumption for hyperspectral imagery, and develop a recurrent neural network structure to exploit potential long-distance dependency relationships among pixels. The GLA is compared with several state-of-the-art unmixing metheds on both real and synthetic data, and the abundance estimation results indicate that our method is promising. We will publish the code of GLA if the paper has the honor to be accepted.

***Index terms*—** Hyperspectral unmixing, autoencoder, deep learning, global and local.

I. Introduction

HYPERSPECTRAL images usually contain hundreds or even thousands of bands. Different bands can reflect various quality characteristics of objects. Therefore, hyperspectral imagery has inherent advantages on internal structure and chemical composition information of objects. However, due to the mutual restriction between spatial and spectral resolution, a single pixel in hyperspectral imagery is usually composed of multiple materials spectra, which is called mixed pixel. The task of unmixing is to decompose the mixed pixels into endmembers and their abundances. The unmixing methods can be classified into geometrical [1]–[3], statistical [4], Nonnegative Matrix Factorization (NMF) [5], [6], deep learning [7], [8] and sparse regression [9], [10] methods. Note that, this is not a strict classification, and some methods may belong to several categories at the same time, for example, NMF-QMV [11] belongs to both geometric and NMF methods, SSLUEP [12] belongs to both statistical and sparse regression methods. Because the unmixing model is ill-posed, in reference [3], Chia-Hsiang Lin proposed an idea of identifying the maximum volume ellipsoid and mapping such ellipsoid into a Euclidean ball. Furthermore, this method is a suitable approach for the lack of the pure pixels. Moreover, some researches also show that spatial-spectral methods may contribute to approaching the real solutions of the unmixing problem [13], [14]. And literatures [6], [15], [16] attempt to solve the ill-posed problem by incorporating the spatial correlation between pixels to the unmixing model as prior information.

One of the popular approaches to integrate the spatial context information is adding it into the loss function. For example, literature [17] combined spatial information with the sparse unmixing model by adding a total variation term. Different from that, literature [6] introduced the total variation regularizer into the statistical methods and improved the unmixing efficiency. Instead of adding an additional spatial regularization term to the spectral-based method, in literatures [18], [19], the spatial and spectral information were combined in the same term by double reweighted method to reduce the regularization coefficient. In literature [20], the spectral and spatial features were unmixed jointly by a cofactorization model. In order to smooth the context information while preserving the edge sharpening, the discontinuity preserving strategy was introduced [21]. In addition to adding regularization terms to the loss function, there are also some methods which divided the input images spatially before solving. For example, in literatures [22]–[27], the super-pixel segmentation was applied to combine the group spatial information with the unmixing models. Considering that the geometric structure of materials in the real scene was not regular, literatures [22]–[24] divided the whole image into irregular sub-blocks, and unmixed each sub-block by NMF [28]. In literatures [25],

This work was supported by the National Key R&D Program of China under the Grant 2019YFC1510905, the National Natural Science Foundation of China under the Grant 62001251 and 62001252 and the China Postdoctoral Science Foundation under the Grant 2020M670631. *(Corresponding author: Tao Li.)*

Xia Xu and Tao Li (Corresponding author) are with the College of Computer Science, Nankai University, Tianjin 300071, China (e-mail: xuxia@nankai.edu.cn; litao@nankai.edu.cn).

Xinyu Song and Bin Pan are with the School of Statistics and Data Science, Nankai University, Tianjin 300071, China, with the Key Laboratory of Pure Mathematics and Combinatorics, Ministry of Education, China, and also with Science and Technology on Special System Simulation Laboratory, Beijing Simulation Center, China. (e-mail:songxy17@gmail.com; panbin@nankai.edu.cn).

Zhenwei Shi is with Image Processing Center, School of Astronautics, Beihang University, Beijing 100191, China (e-mail: shizhenwei@buaa.edu.cn).

[26], Singular Value Decomposition Subset Select and clustering were used after the super-pixel segmentation. In literature [27], the low-rank property of the super-pixel corresponding abundance matrix was added to the loss function.

In recent years, deep learning methods have been introduced into hyperspectral unmixing tasks, most of which were based on auto-encoder (AE) [29]–[34]. In AE, the multi-layer encoder is usually applied to obtain the abundance matrix, and the softmax activation function is used to meet the abundances sum to one constraint (ASC) and nonnegative constraint (ANC) [35]. Assuming the unmixing model is linear, then a single linear decoding layer will be used to reconstruct the image, and the full connection weight can be considered as the endmember matrix. Aiming at the ill-posedness of the unmixing problem, some prior knowledge is added to the AE model. For example, according to the assumption that abundance matrices of different endmembers are independent, the orthogonal prior information of abundance vectors was applied to AE in [34]. In literature [36], an abundance adaptive smoothing (AAS) term which explored spatial context information was added to the loss function.

To further extract the joint spatial-spectral information, another popular deep learning model, convolution neural network (CNN), is introduced to hyperspectral unmixing [37]–[39]. These methods attempted to discover the spatial correlation in the hyperspectral data and automatically extract the effective relevant information from the imagery. In literature [40], Khajehrayeni and Ghassemian combined CNN and AE, and developed a convolutional auto-encoder (CAE) to extract spatial related features for the unmixing task. In literature [41], the 3D convolution was applied to further enhance the estimation performance of endmembers and abundances.

However, recently proposed deep learning based unmixing methods may not take full account on the complexity of real remote sensing scenarios. One of the major challenges for deep learning based unmixing methods is that they usually only conduct unified smoothing on the hyperspectral imagery, which have ignored the variations inside the imagery and may result in over-smoothing. In real scenarios, hyperspectral imagery usually has the characteristics of local homogeneity [17] and the global self-similarity [42]. The local homogeneity refers to the continuous distribution of the spectral values in the local neighborhood. In literature [42], Dong Le et al. proposed the abundance constraints of global and local self-similarity, and defined global self-similarity of pixels as the correlation of all pixels that exists among the whole image. Therefore, different from the definition in literature [43], we refer to self-similarity as similar spectral vectors with similar abundance vectors.

In this paper, we propose a fully unsupervised unmixing method based on a Global-Local smoothing Autoencoder (GLA) for hyperspectral unmixing. The motivation of GLA is to explore the local homogeneity and the global self-similarity of hyperspectral imagery, and describe these characteristics via a powerful deep learning model. The proposed GLA network mainly include two novel components: a Local Continuous conditional random field Smoothing (LCS) module and a Global Recurrent Smoothing (GRS) module.

LCS targets at local smoothing, which tries to exploit the adaptive weights for pixels around the edge or inside a region. Inspired by the local smoothing ability of continuous Conditional Random Field (CRF) [44], [45], we construct the LCS module based on CRF. However, parameters in CRF vary for different tasks, and users have to manually decide the CRF parameters for hyperspectral unmixing. In LCS, we integrate the CRF into the network, and automatically optimize the CRF parameters through the network. Furthermore, different from full connection form in literature [44], we improve it to local connections so as to describe the local joint spatial-spectral information. LCS algorithm can achieve the adaptive smoothing weights adjustment for different positions around a pixel.

GRS tries to refine the abundance map via global smoothing, which is motivated by the global self-similarity of hyperspectral data. In GRS, we construct the global smoothing module via a Recurrent Neural Network (RNN) [46] which incorporate the potential information of all pixels. The GRS module is composed of different directions, each of which conducts convolution and update formula. By this means GRS is able to make full use of the abundance self-similarity between similar spectral vectors.

The contributions of this paper are summarized as follows:

- We propose a new deep learning model, global-local smoothing autoencoder, for hyperspectral unmixing, which can utilize the potential joint spatial-spectral information among pixels.
- A novel LCS module is developed to smooth local homogeneity regions, where the smoothing parameters are adaptively determined and the hyperparameters are optimized by the network.
- A GRS module is designed to extract the global self-similarity information, where a RNN structure is embedded to exploit the pixels relationship from the spectral feature space.

II. Methodology

Depending on the spectral reflection, refraction, and the geometry of the scenario, the unmixing model adopted can be linear or nonlinear. Due to the advantages of simplicity, high efficiency and clear physical meaning, linear unmixing has become a popular research. Therefore, this method is based on linear unmixing model. In this section, we first introduce the linear unmixing model. Then, the overall framework of the model is introduced in section B. In section C and D, the GRS and LCS module are introduced, respectively. Finally, we give the total loss function of the model in section E.

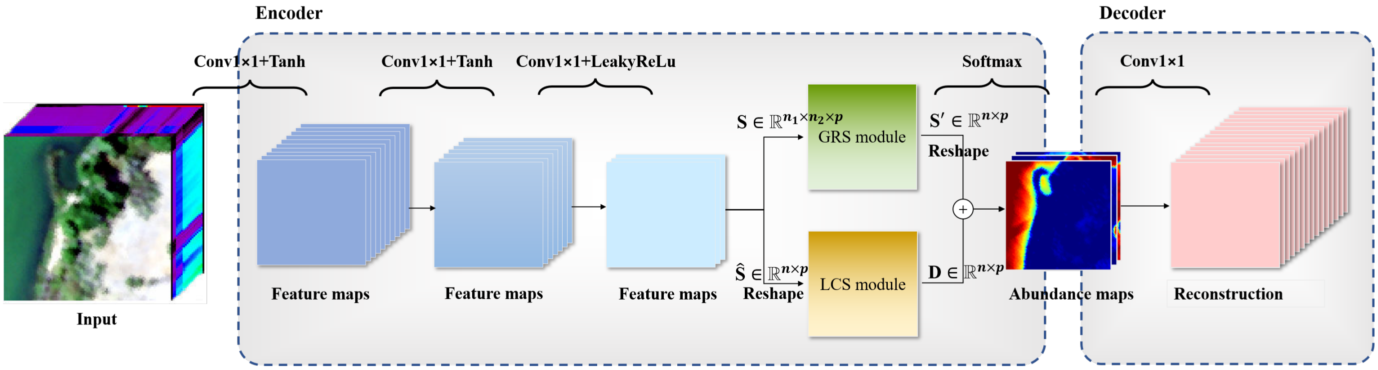


Fig. 1. Network architecture

A. Linear unmixing model

Let $\boldsymbol{x}_i \in \mathbb{R}^{l\times 1}$ denotes the spectral vector of the i-th pixel in the hyperspectral image, then in the linear model, the i-th pixel can be represented as:

$$\boldsymbol{x}_i = \mathbf{M} \times \boldsymbol{h}_i + \boldsymbol{\xi}_i \tag{1}$$

where $\mathbf{M} \in \mathbb{R}^{l\times p}$ is the endmember matrix of the image, $\boldsymbol{\xi}_i \in \mathbb{R}^{l\times 1}$ is the noise vector, $\boldsymbol{h}_i \in \mathbb{R}^{p\times 1}$ is the abundance vector corresponding to the i-th pixel, p is the number of endmembers in the image, and l is the number of bands.

Assuming that there are n pixels in the image, eq. (1) can be rewritten in a matrix form as follows:

$$\mathbf{X} = \mathbf{M} \times \mathbf{H} + \boldsymbol{\Xi} \tag{2}$$

where the image data $\mathbf{X}=[\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_n] \in \mathbb{R}^{l\times n}$, fractional abundance matrix $\mathbf{H}=[\boldsymbol{h}_1, \boldsymbol{h}_2, \ldots, \boldsymbol{h}_n] \in \mathbb{R}^{p\times n}$, noise matrix $\boldsymbol{\Xi}=[\boldsymbol{\xi}_1, \boldsymbol{\xi}_2, \ldots, \boldsymbol{\xi}_n] \in \mathbb{R}^{l\times n}$.

Besides, each endmember vector should satisfy the endmember nonnegative constraint (ENC), the abundance vector should satisfy ANC and ASC, that is:

$$\begin{aligned} &\text{ENC} : \mathbf{M} \geqslant 0 \\ &\text{ANC} : \mathbf{H} \geqslant 0 \\ &\text{ASC} : \mathbf{1}_p^T \mathbf{H} = \mathbf{1}_n^T \end{aligned} \tag{3}$$

B. The overall structure

In this section, we describe the overall structure of the proposed GLA (illustrated in Fig. 1).

In the encoder, the first layer is the input layer. The second layer is composed of a 1×1 convolution with $\lfloor \frac{l}{2} \rfloor$ feature maps and the tanh activation function, which limits the output to [-1,1] and makes the model easier to converge. The third layer is composed of a 1×1 convolution with $\lfloor \frac{l}{4} \rfloor$ feature maps and the tanh activation function. And the forth layer is composed of 1×1 convolution with p feature maps and the LeakyRuLu activation function, which makes the output nearly nonnegative.

Next comes the spatial information extraction layer. This layer is composed of the LCS and GRS modules. The local and global features of the input image could be extracted through these two modules, respectively. After fusion of their outputs, in order to satisfy the sum-to-one constraint at each pixel, the softmax function is applied to the output feature maps. The new feature maps can be considered as abundances where the ANC and ASC are enforced at each pixel. The input pixel $\boldsymbol{x}_i$ will get a corresponding abundance vector $\boldsymbol{h}_i$=$f_e(\boldsymbol{x}_i)$, where $f_e(\cdot)$ is the encoding function.

The decoder consists of only one layer of 1×1 convolution with l feature maps without considering bias, i.e. $\widehat{\boldsymbol{x}}_i = f_d(\boldsymbol{h}_i)$ = $\mathbf{W} \times f_e(\boldsymbol{x}_i) = \mathbf{W} \times \boldsymbol{h}_i$, where $\mathbf{W}$ is the weight of the convolution kernel and $f_d(\cdot)$ is the decoding function, $\widehat{\boldsymbol{x}}_i$ is the reconstruction result of the image. Therefore, based on the linear structure of the decoder, the proposed method can be used for linear unmixing model.

Moreover, the proposed GLA does not require real endmembers or a known spectral library, so it is an unsupervised method. And different from treating the vertices of simplex as endmembers by projection, GLA generates endmembers directly through neural network so that it does not need to satisfy the pure-pixel assumption.

C. GRS module

According to the above description of the proposed network, the feature maps obtained from the first three convolutions are passed to GRS and LCS to extract the global and local spatial information, respectively. In this section, we first introduce the proposed GRS.

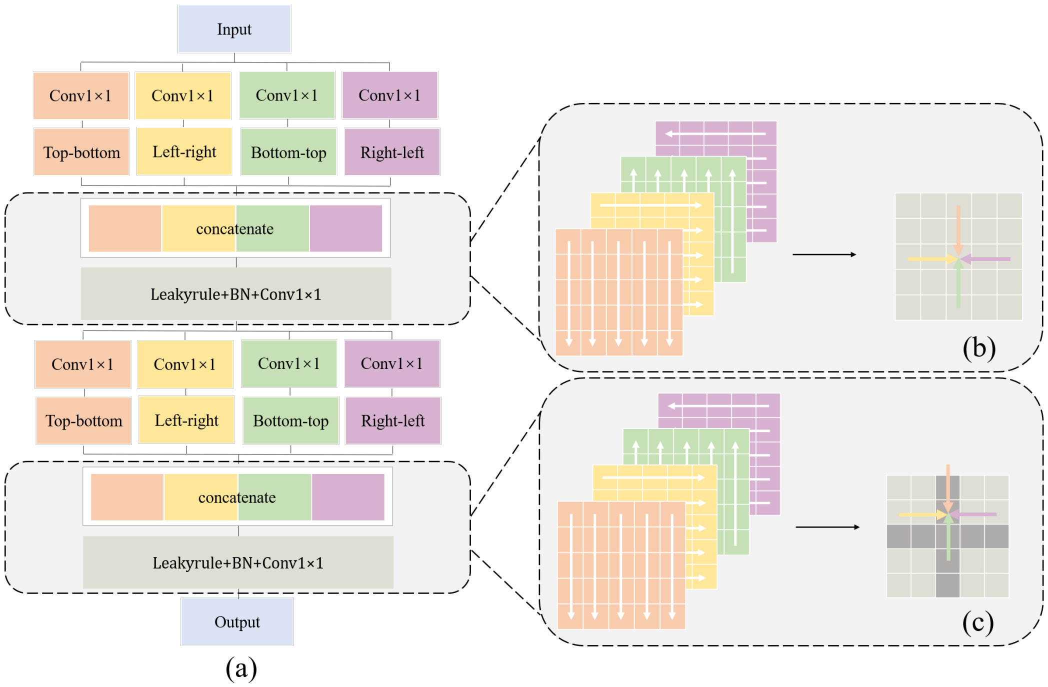


Fig. 2. Structure of GRS module. (a) is the overall structure of GRS module, (b) (c) is the illustration of the smoothing process, the dark gray areas in (c) indicate the extent of smoothing.

Considering the self-similarity of hyperspectral images, the long-distance correlation between pixels is introduced in GRS. In literature [47], the hyperspectral images are regarded as sequential data to take full advantage of the spectral correlation. Therefore, GRS is designed based on RNN, which is used to extract global sequence features. First, GRS integrates the information within the cross range of each pixel, which is obtained by replacement formula in four directions of the image. Second, the operation is repeated to get the global information.

The structure of the GRS module is shown in Fig. 2(a). The first layer is the input layer. The height n_1 and width n_2 of the input feature map for GRS module are consistent with those of hyperspectral image. In the second layer, the input $\mathbf{S} \in \mathbb{R}^{n_1 \times n_2 \times p}$ is convoluted by four different 1×1 convolution kernels. Therefore, every point $\boldsymbol{S}_{uv}$ in $\mathbf{S}$ gets four activation values $\boldsymbol{a}_{uv}^{\text{top}}$, $\boldsymbol{a}_{uv}^{\text{bottom}}$, $\boldsymbol{a}_{uv}^{\text{left}}$, $\boldsymbol{a}_{uv}^{\text{right}}$, where (u, v) is the spatial location. The obtained four feature maps will be used as the inputs of the third layer to extract the information of four directions, respectively.

In the third layer, the information in the four directions will be extracted through the different replacement formulas. For each direction, the activation value of each pixel is replaced by the linear combination of the current activation value and the activation value of the previous pixel. For example, from top to bottom, the replacement formula is:

$$\boldsymbol{a}_{uv}^{\text{top}} = max\{\mathbf{W}^{\text{top}}\boldsymbol{a}_{u-1,v}^{\text{top}} + \boldsymbol{a}_{uv}^{\text{top}}, \mathbf{0}\} \tag{4}$$

where $\mathbf{W}^{\text{top}} \in \mathbb{R}^{p\times p}$ is the weight matrix in the top-down replacement formula.

In the forth layer, the feature map of $4p$ channels is obtained by concatenating the feature maps of four directions. The fifth layer is a 1×1 convolution to reduce the channels of the feature map from $4p$ to p. At the training process, we use batch normalization to reduce the internal covariance shift in the hidden layer and use LeakyReLu activation function to avoid the problem of neuron death.

After the top-bottom and bottom-top replacement formulas of the third layer, each pixel can obtain the information of its column. Similarly, the information of the row in which a pixel is located can be obtained by the left-right and right-left replacement formulas. As shown in Fig. 2(b), in the feature map obtained by the first five layers, each pixel can extract the information within its four directions. Then the output of fifth layer is smoothed again from the sixth layer to the ninth layer by repeating the process from the second layer to the fifth layer. But in this time, as shown in Fig. 2(c), this operation smooths the pixels already containing the cross information so that each pixel in the output of GRS module can obtain the information of all pixels. Finally, in order to integrate with the output of LCS module, the output of GRS module is reshaped as $\mathbf{S}' \in \mathbb{R}^{p\times n}$.

After GRS module, each pixel in $\mathbf{S}'$ can obtain the information of all pixels, i.e. the i-th pixel $\boldsymbol{S}_i'$=$g_i(\boldsymbol{x}_1, \boldsymbol{x}_2, ..., \boldsymbol{x}_n)$, where $g_i(\cdot)$ represents the relationship between $\boldsymbol{S}_i'$ and all pixels.

The pseudocode of GRS is shown in **Algorithm 1**.

Algorithm 1: The pseudocode of GRS

```
Input: feature map $\mathbf{S}$.
Output: feature map for extracting global information $\mathbf{S}'$.
Initialization:
channels of the input $p$, direc={top, bottom, left, right}, number of times to extract cross information $T_G$.
global smoothing process:
Set $C_0(\mathbf{S}) = \mathbf{S}$
while $t < T_G$ do
    for $\eta$ in direc do
        Calculate the convolutional layer $C_t^{\eta}(\mathbf{S}) = conv(C_t(\mathbf{S}), p, p)$
    Concatenate the outputs of previous layer to get $C_t(\mathbf{S}) \leftarrow [C_t^{\text{top}}(\mathbf{S}), C_t^{\text{bottom}}(\mathbf{S}), C_t^{\text{left}}(\mathbf{S}), C_t^{\text{right}}(\mathbf{S})]$
    $C_{t+1}(\mathbf{S}) = conv(\text{LeakyReLu}(\text{BN}(C_t(\mathbf{S}))), 4p, p)$
    $t = t + 1$
Reshape the output of last convolution layer $C_{T_G}(\mathbf{S})$ as $\mathbf{S}' \in \mathbb{R}^{p \times n}$
Return $\mathbf{S}'$.
```

D. LCS module

In this paper, we propose a LCS module to extract local features. It has the following three characteristics. First, in LCS, the fully connected CRF is transformed into local connections to extract the local information. Second, an adaptive weight method based on CRF is proposed to prevent the edge of abundance maps from blurring. Third, a larger local neighborhood is applied in LCS to obtain more precise context information.

When CRF is applied to the unmixing task, the conditional probability formula $P(\mathbf{D}|\hat{\mathbf{S}})$ is usually defined as follows:

$$P(\mathbf{D}|\hat{\mathbf{S}}) = \frac{1}{Z(\hat{\mathbf{S}})} \exp(-\Omega(\mathbf{D}, \hat{\mathbf{S}})) \tag{5}$$

where the normalization term $Z(\hat{\mathbf{S}})$ and the term in exponent $\Omega(\mathbf{D}, \hat{\mathbf{S}})$ are defined as:

$$Z(\hat{\mathbf{S}}) = \int \exp(-\Omega(\mathbf{D}, \hat{\mathbf{S}}))\, \mathrm{d}\mathbf{D} \tag{6}$$

$$\Omega(\mathbf{D}, \hat{\mathbf{S}}) = \sum_{i=1}^{n} \Phi(\boldsymbol{d}_i, \hat{\boldsymbol{s}}_i) + \sum_{i,j} \varphi(\boldsymbol{d}_i, \boldsymbol{d}_j) \tag{7}$$

$\hat{\mathbf{S}}$=$[\hat{\boldsymbol{s}}_1,\ \hat{\boldsymbol{s}}_2, \cdots,\ \hat{\boldsymbol{s}}_n] \in \mathbb{R}^{p \times n}$ is the reshaped feature map and $\hat{\boldsymbol{s}}_i$ is the feature corresponds to the abundance vector $\boldsymbol{h}_i$, $\mathbf{D}$=$[\boldsymbol{d}_1, \boldsymbol{d}_2, \cdots, \boldsymbol{d}_n]$$\in \mathbb{R}^{p \times n}$ is the feature map adjusted by LCS and $\boldsymbol{d}_i$ is the feature that contains local spatial information corresponds to $\boldsymbol{h}_i$, function Φ is to limit that abundance of each pixel does not change much before and after adjustment, function φ is to limit that similar abundance remains similar after adjustment. In general, we define:

$$\Phi(\boldsymbol{d}_i, \hat{\boldsymbol{s}}_i) = ||\boldsymbol{d}_i - \hat{\boldsymbol{s}}_i||^2 \tag{8}$$

$$\varphi(\boldsymbol{d}_i, \boldsymbol{d}_j) = \exp\left(-\frac{||\hat{\boldsymbol{s}}_i - \hat{\boldsymbol{s}}_j||^2}{2\theta}\right) ||\boldsymbol{d}_i - \boldsymbol{d}_j||^2 \tag{9}$$

where θ is a hyperparameter.

Define k_{ij}=$\exp(-\frac{||\hat{\boldsymbol{s}}_i - \hat{\boldsymbol{s}}_j||^2}{2\theta})$, then we have $\varphi(\boldsymbol{d}_i, \boldsymbol{d}_j)$=$k_{ij}\ ||\boldsymbol{d}_i - \boldsymbol{d}_j||^2$. Suppose that the approximate distribution of $P(\mathbf{D}|\hat{\mathbf{S}})$ is $Q(\mathbf{D}|\hat{\mathbf{S}})$ =$\prod_{i=1}^{n} Q(\boldsymbol{d}_i|\hat{\mathbf{S}})$, then the KL divergence of $P(\mathbf{D}|\hat{\mathbf{S}})$ and $Q(\mathbf{D}|\hat{\mathbf{S}})$ is

$$\begin{aligned}
&\text{KL}\left(P(\mathbf{D}|\hat{\mathbf{S}}) || Q(\mathbf{D}|\hat{\mathbf{S}})\right) \\
&= E_{P(\mathbf{D}|\hat{\mathbf{S}})}\left(\log P(\mathbf{D}|\hat{\mathbf{S}}) - \log Q(\mathbf{D}|\hat{\mathbf{S}})\right) \\
&= E_{\boldsymbol{d}_i}\left(E_{\mathbf{D}\setminus\boldsymbol{d}_i}(\log P(\mathbf{D}|\hat{\mathbf{S}}))\right) - E\left(\sum_{j=1}^{n} \log Q(\boldsymbol{d}_j|\hat{\mathbf{S}})\right) \\
&= E_{\boldsymbol{d}_i}\left(E_{\mathbf{D}\setminus\boldsymbol{d}_i}(\log P(\mathbf{D}|\hat{\mathbf{S}}))\right) - E\left(\log Q(\boldsymbol{d}_i|\hat{\mathbf{S}})\right) + \text{const}
\end{aligned} \tag{10}$$

where $E(\cdot)$ is the expectation function, $\mathbf{D}\backslash\boldsymbol{d}_i$ denotes the i-th column $\boldsymbol{d}_i$ of matrix $\mathbf{D}$ is removed, const is a constant independent of $\mathbf{D}$ and $\hat{\mathbf{S}}$. By minimizing $\text{KL}\left(P(\mathbf{D}|\hat{\mathbf{S}})\|Q(\mathbf{D}|\hat{\mathbf{S}})\right)$, we can get the result:

$$\log Q(\boldsymbol{d}_i|\hat{\mathbf{S}}) = E_{\mathbf{D}\backslash\boldsymbol{d}_i}\left(\log P(\mathbf{D}|\hat{\mathbf{S}})\right) + \text{const} \tag{11}$$

If Eq.(5), Eq.(7), Eq.(8) and Eq.(9) is brought into Eq.(11), then

$$\begin{aligned}\log Q(\boldsymbol{d}_i|\hat{\mathbf{S}}) = &-\left(1+2\sum_{j\neq i}k_{ij}\right)||\boldsymbol{d}_i||^2 \\ &+2\boldsymbol{d}_i^T\left(\hat{\boldsymbol{s}}_i+2\sum_{j\neq i}k_{ij}E\boldsymbol{d}_j\right)+\text{const}\end{aligned} \tag{12}$$

The iterative formula can be obtained by maximizing the likelihood function $Q(\boldsymbol{d}_i|\hat{\mathbf{S}})$. And in LCS module, we use locally connected continuous CRF to extract homogeneous information:

$$\begin{aligned}\boldsymbol{d}_i^t &= \frac{\hat{\boldsymbol{s}}_i+2\sum_{j\neq i,j\in\mathcal{N}(i)}k_{ij}E\boldsymbol{d}_j}{1+2\sum_{j\neq i,j\in\mathcal{N}(i)}k_{ij}} \\ &= \frac{\hat{\boldsymbol{s}}_i+2\sum_{j\neq i,j\in\mathcal{N}(i)}k_{ij}\boldsymbol{d}_j^{t-1}}{1+2\sum_{j\neq i,j\in\mathcal{N}(i)}k_{ij}}\end{aligned} \tag{13}$$

where $\mathcal{N}(i)$ is the neighborhood position set of the i-th pixel and $\boldsymbol{d}_i^t$ is the t-th iteration of the adjusted vector corresponding to the i-th pixel.

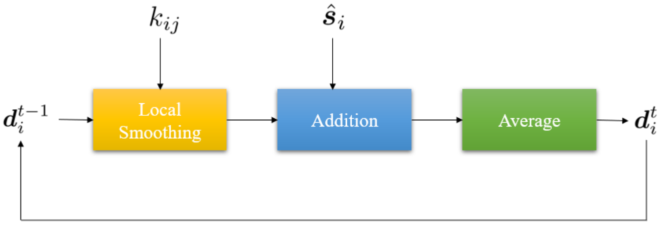


Fig. 3. Structure of LCS module

According to Eq. (13), the process of updating $\boldsymbol{d}_i^t$ is shown in Fig. 3.

- Local smoothing, which computes the weighted sum of neighboring pixels, e.g., $\sum_{j\neq i,j\in\mathcal{N}(i)}k_{ij}\boldsymbol{d}_j^{t-1}$ for $\boldsymbol{d}_i^t$.
- Addition, which combines the pixel information with its neighborhood, i.e., $\hat{\boldsymbol{s}}_i+2\sum_{j\neq i,j\in\mathcal{N}(i)}k_{ij}\boldsymbol{d}_j^{t-1}$.
- Average, which divides $1+2\sum_{j\neq i,j\in\mathcal{N}(i)}k_{ij}$ to obtain the weighted average results.

In order to express Eq. (13) by a matrix form, we define the smoothing matrix $\mathbf{K}$, where the element of the i-th row and j-th column is k_{ij}. The initialization matrix $\mathbf{K}$ is defined as:

$$\mathbf{K}_{\text{init}} = \begin{bmatrix} 0 & \frac{1}{3} & 0 & & & & \\ \frac{1}{2} & 0 & & \ddots & & & \\ \vdots & \frac{1}{3} & \ddots & \frac{1}{4} & \ddots & \frac{1}{3} & \\ \frac{1}{2} & \vdots & & 0 & & \vdots & \frac{1}{2} \\ & \frac{1}{3} & \ddots & \frac{1}{4} & \ddots & \frac{1}{3} & \vdots \\ & & & \ddots & & 0 & \frac{1}{2} \\ & & & & 0 & \frac{1}{3} & 0 \end{bmatrix}_{n\times n} \tag{14}$$

where $\vdots$ denotes the middle $w-1$ zeros and w is the width of the input image. Then the iterative formula is:

$$\mathbf{D}^t = \left[\boldsymbol{d}_1^t, \boldsymbol{d}_2^t, \cdots, \boldsymbol{d}_n^t\right] = \frac{\hat{\mathbf{S}}+2\mathbf{K}\mathbf{D}^{t-1}}{\mathbf{1}_n+2\mathbf{K}\mathbf{1}_n} \tag{15}$$

where $\mathbf{1}_n$=$[1,1,\ldots,1]^T$$\in$$\mathbb{R}^n$, the fraction represents the element-wise division. In order to maintain the local smoothness of LCS module, the loss of LCS module is $||\mathbf{K}\odot\mathbf{G}||_{\text{F}}^2$, where $\odot$ is the Hadamard product, $\mathbf{G}$=$\mathbf{K}_{\text{init}}$>$\mathbf{0}$, ">" is the element-wise Boolean operation. The pseudocode of LCS is shown in **Algorithm 2**.

Algorithm 2: The pseudocode of LCS

```
Input: feature map $\hat{\mathbf{S}}$.
Output: feature map for extracting local information $\mathbf{D}^{T_L}$.
Initialization:
pixel number $n$, maximum iteration number $T_L$, smoothing factor $\mathbf{K}^0 = \mathbf{K}_{\text{init}}$, $\mathbf{D}^0 = [\boldsymbol{d}_1^0, \boldsymbol{d}_2^0, \cdots, \boldsymbol{d}_n^0]$.
local smoothing process:
while $t < T_L$ do
    $t = t + 1$;
    for $i = 1, ..., n$ do
        Set $\boldsymbol{d}_i^t = \frac{\hat{\boldsymbol{s}}_i + 2\sum_{j \neq i, j \in \mathcal{N}(i)} k_{ij}^{t-1} \boldsymbol{d}_j^{t-1}}{1 + 2\sum_{j \neq i, j \in \mathcal{N}(i)} k_{ij}}$
    Set $\mathbf{D}^t = [\boldsymbol{d}_1^t, \boldsymbol{d}_2^t, \cdots, \boldsymbol{d}_n^t]$
Return $\mathbf{D}^{T_L} = [\boldsymbol{d}_1^{T_L}, \boldsymbol{d}_2^{T_L}, \cdots, \boldsymbol{d}_n^{T_L}]$.
```

E. Loss function

Based on the previous description of each module, the total loss L_{total} in GLA can be written as the following three parts

$$L_{\text{total}} = L_{\text{rec}} + L_{\text{sparse}} + L_{\text{spatial}} \tag{16}$$

It is proved in [48] that the spectral angle distance (SAD) could get better estimation results when used as reconstruction loss function. So the first term L_{rec} is defined as:

$$L_{\text{rec}} = \frac{1}{n}\sum_{i=1}^{n} \arccos\left(\frac{\langle \boldsymbol{x}_i, \hat{\boldsymbol{x}}_i \rangle}{\|\boldsymbol{x}_i\|_2 \|\hat{\boldsymbol{x}}_i\|_2}\right) \tag{17}$$

In the second term, we use $l_{1/2}$-regularization to enforce the sparsity constraint. Then

$$L_{\text{sparse}} = \alpha \|\mathbf{H}\|_{1/2} = \alpha \sum_{i=1}^{n}\sum_{r=1}^{p} |h_{ri}|^{1/2} \tag{18}$$

where α is a hyperparameter.

And for the third term, firstly, based on the assumption of image self-similarity, the abundance vector of i-th pixel $\boldsymbol{h}_i$ should be a function of the spectral vector of all pixels, i.e.

$$\boldsymbol{h}_i = f_i(\boldsymbol{x}_1, \boldsymbol{x}_2, ..., \boldsymbol{x}_n) \tag{19}$$

where $f_i(\cdot)$ represents the relationship between $\boldsymbol{h}_i$ and all spectra $\mathbf{X}$. After GRS module, the i-th feature vector $\boldsymbol{S}_i^{'}$ of the output $\mathbf{S}^{'}$ can be written as $g_i(\boldsymbol{x}_1, \boldsymbol{x}_2, ..., \boldsymbol{x}_n)$, so we can define

$$f_i(\boldsymbol{x}_1, \boldsymbol{x}_2, ..., \boldsymbol{x}_n) \approx \text{softmax}(\boldsymbol{S}_i^{'} + \boldsymbol{d}_i^{T_L}) \tag{20}$$

Secondly, due to the local homogeneous smoothing of LCS module, we assume the i-th abundance $\boldsymbol{h}_i$ can be approximately represented as

$$\boldsymbol{h}_i = \sum_{j \in \mathcal{N}(i)} b_{ij} \boldsymbol{h}_j \tag{21}$$

where b_{ij} is the coefficient corresponding to $\boldsymbol{h}_j$.

In order to satisfy self-similarity and homogeneity simultaneously, Eq.(19), Eq.(20) and Eq.(21) are combined to obtain the spatial loss function

$$\begin{aligned} L_{\text{spatial}} &= \beta \|\mathbf{H} - \mathbf{H}\mathbf{B}\|_{\text{F}}^2 + \gamma(||\mathbf{K} \odot \mathbf{G}||_{\text{F}}^2 + ||\mathbf{B} \odot \mathbf{G}||_{\text{F}}^2) \\ &+ \tau \|\mathbf{1}_n^T (\mathbf{B} \odot \mathbf{G}) - \mathbf{1}_n^T\|_{\text{F}}^2 \end{aligned} \tag{22}$$

where $\mathbf{H}$ is the fractional abundance matrix, the i-th row and j-th column in $\mathbf{B}$ is b_{ij}, smoothing matrix $\mathbf{K}$ is defined in LCS module, $\mathbf{G}$=$\mathbf{K}_{\text{init}}$>$\mathbf{0}$, ">" is the element-wise Boolean operation. $\mathbf{B}$ is initialized like $\mathbf{K}_{\text{init}}$, β, γ, τ are the hyperparameters, the last term in L_{spatial} enforces the sum of the coefficients to be one.

III. Experiments and discussion

We use MSE criterion to measure the quality of abundance estimations, and the proposed method GLA is compared with the geometry-based, NMF-based and deep-learning-based methods. The comparison methods include: NMF-$L_{1/2}$ [49], NMF-anls [50], NMF-QMV [11], MVES [51], HyperCSI [52], HISUN [3], CNNAEU [39], AAS [36].

$$\mathrm{MSE} = \frac{1}{n}\sum_{i=1}^{n}\left\|\boldsymbol{h}_i - \widetilde{\boldsymbol{h}}_i\right\|^2 \tag{23}$$

where $\widetilde{\boldsymbol{h}}_i$ is the reference abundance fraction of the i-th pixel, $\boldsymbol{h}_i$ is the estimated abundance fraction of the i-th pixel.

The numbers of endmember p is obtained by HySime algorithm before network training. [53] The weight of the convolution kernel in decoder is initialized by the endmember matrix from VCA [54], and the initial iteration $\mathbf{D}^0$ in LCS module is defined by abundance matrix from FCLS [55]. Futhermore, using Adam optimizer and attenuation learning rate to ensure the stability of training results. Setting the network parameters T_L=10, T_G=2 and α=1e-5, β=1e-6, γ=1e-5, τ=5, the learning rates of encoder and decoder are 1e-2 and 1e-3, respectively, and the learning rate is reduced by 90% every 200 iterations.

A. Real and synthetic data

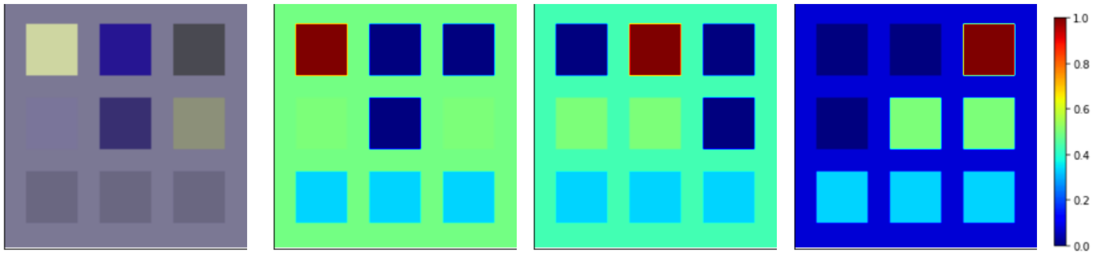
The experiments were carried out on three real data sets and three synthetic data sets. The description of each dataset is as follows:

1) real data:

- Samson is a common real data set, with 95×95 pixels, and each pixel has 156 channels covering the wavelengths from 401 nm to 889 nm. The image has three endmembers: soil, tree and water.
- Jasper Ridge is a real image contains 100×100 pixels, each pixel has 224 channels ranging from 380 nm to 2500 nm. Affected by water vapor and atmosphere, 198 channels are left after removing channels 1-3, 108-112, 154-166 and 220-224. The image has four endmembers: road, soil, tree and water.
- Cuprite is a real hyperspectral image containing 150×150 pixels, each pixel has 224 channels ranging from 400 nm to 2500 nm. After removing the noisy channels and water absorption channels, 183 channels are remained. The estimated number of endmembers is nine, and the reference endmembers in USGS library are Chalcedony, Montmorillonite, Buddingtonite, Muscovite, Hematite, Kaolinite, Alunite, Pyrope and Andradite.

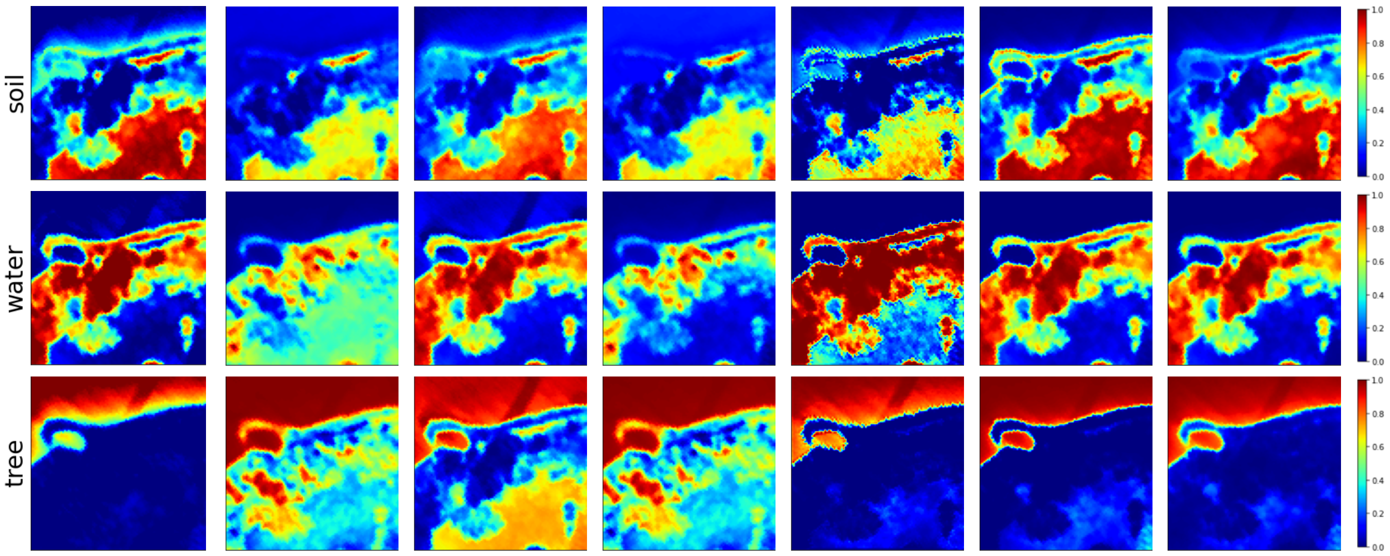
2) synthetic data:

- Data1 generation process:
  Given the number of endmembers z contained in the image and the corresponding spectral vector, the 100×100 image is divided into 100 square regions with 10 sides. Each region contains only one pure spectrum, and the mixed pixels are generated by using the 11×11 spatial filter. Pixels with abundances greater than 0.8 are removed and replaced with a mixture of two endmembers. Then 30db Gaussian noises are added. The spectral vector of pure materials is selected from the United States Geological Survey (USGS) library splib07a. When z is set as 3-10, the generated data sets are named data1-edm3, data1-edm4,..., data1-edm10 accordingly.
  10 endmembers with 224 channels are selected, which are: Acmite, Actinolite, Elbaite, Grass_dry, Pyroxmangite, Serpentine, Halloysite, Dumortierite, Opal, Quartz.
- Data2 generation process:
  Given the number of endmembers and endmember matrix in advance, the matrix with the same size of the image is divided into several small square regions according to the specified interval, and the abundance matrix is generated by assigning values to the matrix according to certain rules, as shown in Fig.4. After the image is generated, the 30db Gaussian white



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

Fig. 4. data2-edm3 (a) simulated image (b)abundance of endmember1 (c) abundance of endmember2 (d) abundance of endmember3



(a) reference (b) NMF-$L_{1/2}$ (c) NMF-anls (d) NMF-QMV (e) CNNAEU (f) AAS (g) GLA

Fig. 5. The abundance estimation of Samson data set is compared with six methods. The first column is the reference abundance.

noises are added to get the synthetic image data. Similarly, suppose that the synthetic data contains 3-10 endmembers, and the generated data is named data2-edm3, data2-edm4, ..., data2-edm10.

- Data3 generation process:
  In Data3, three randomly selected endmembers are used to generate 100 × 100 × 224 simulation data according to the Dirichlet distribution, and add 20db, 30db and 40db noise, respectively. If the abundance of a single endmember in a pixel is greater than 0.7, the pixel is replaced by a mixture of two endmembers (the abundance of both endmembers in the pixel is 0.5).

B. *Experimental results of abundance matrix*

TABLE I: Mean MSE and standard deviation of Samson and Jasper Ridge data sets. The best result is bold.

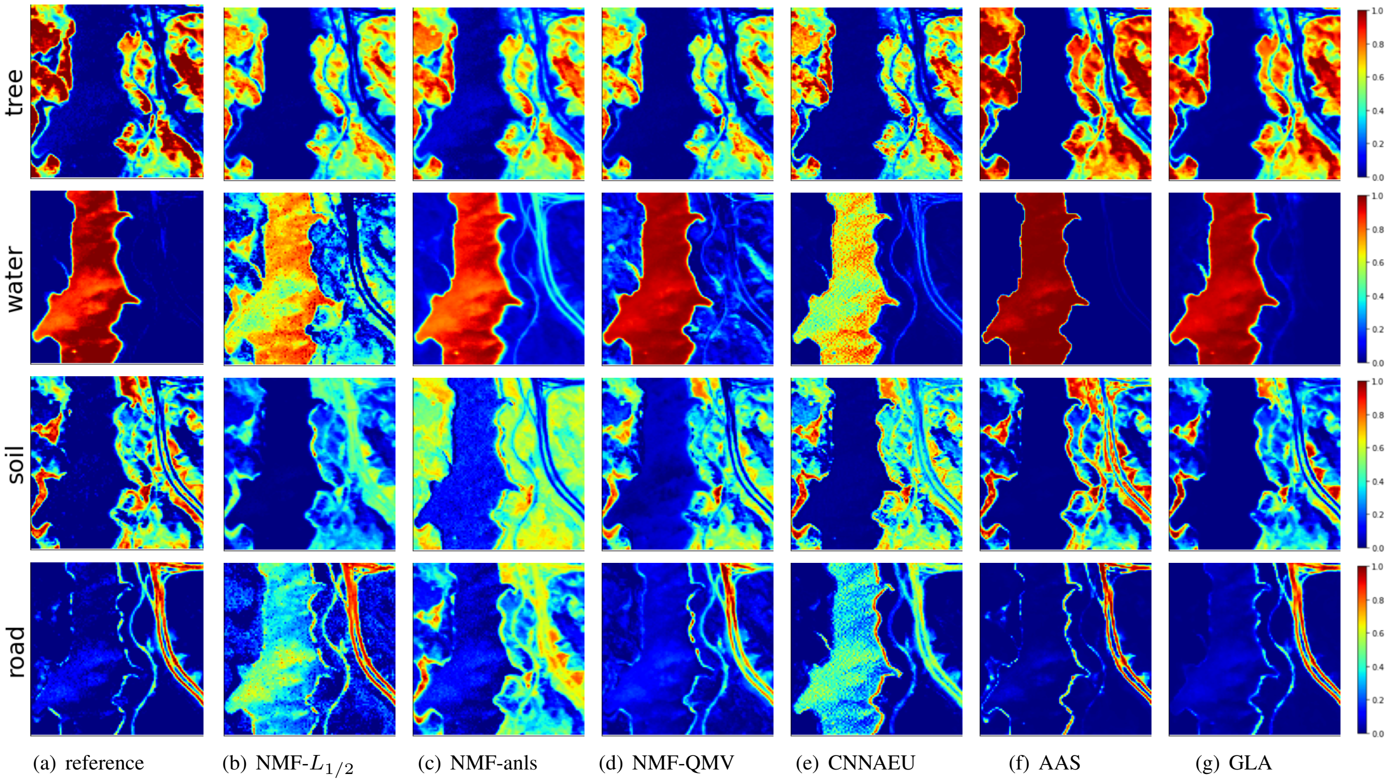
	NMF-$L_{1/2}$		NMF-anls		NMF-QMV		CNNAEU		AAS		GLA	
	mean	std	mean	std	mean	std	mean	std	mean	std	mean	std
Samson	0.3460	0.0433	0.2791	0.0269	0.2253	0	0.0613	0.0338	0.0360	0.0011	**0.0279**	0.0019
Jasper Ridge	0.3510	0.0263	0.1590	0.0289	0.0635	0	0.1187	0.0172	0.0678	0.0157	**0.0386**	0.0028

TABLE II: Endmember and abundance results of Samson and Jasper Ridge data set. The best result is bold.

	MVES		HyperCSI		HISUN		GLA	
	edm	abd	edm	abd	edm	abd	edm	abd
Samson	0.1831±0	0.3230±0	0.1564±0	0.2002±0	0.1915±0	0.2789±0	**0.0795**±0.0056	**0.0279**±0.0019
Jasper Ridge	0.2431±0	0.2945±0	0.2461±0	0.1036±0	0.2227±0	0.1387±0	**0.1037**±0.0135	**0.0386**±0.0028

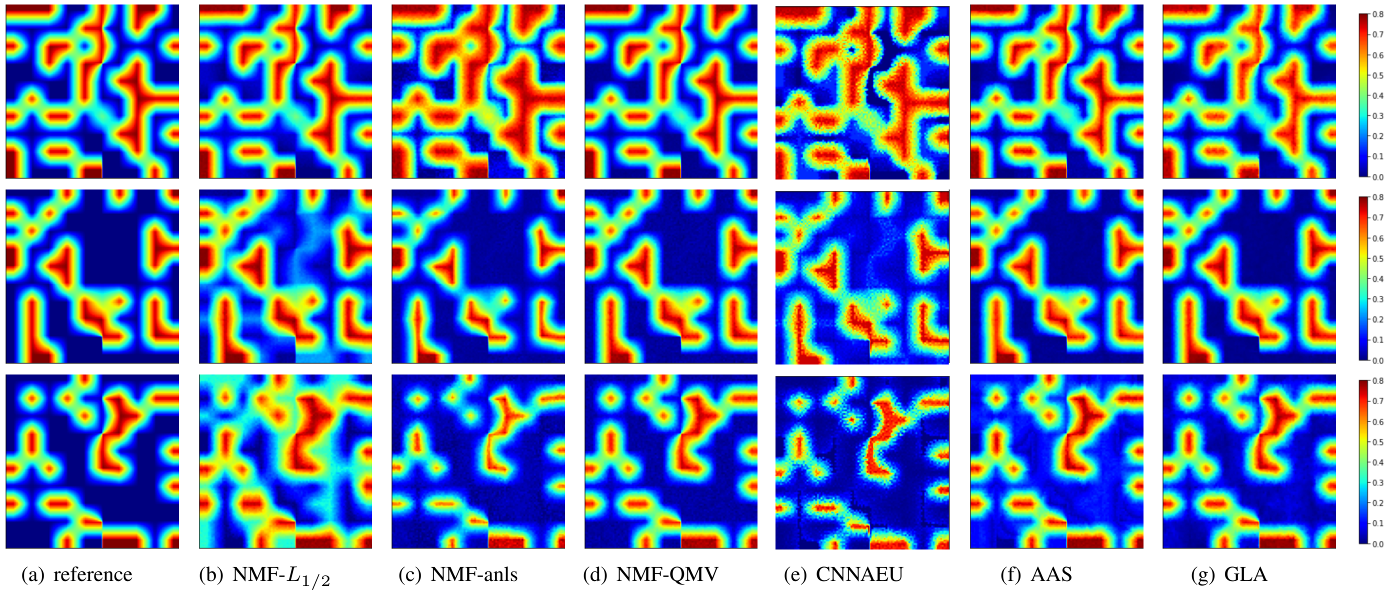
Table I and II show the MSE and standard deviation of Samson and Jasper Ridge datasets obtained by the NMF, geometric and deep learning methods. For Samson dataset, it can be observed that GLA obtains a competitive result with 0.0279 MSE and a low standard deviation of 0.0019. For Jasper Ridge dataset, GLA has a MSE of 0.0386 and a low standard deviation. Fig.5 and Fig.6 illustrate the abundance visualization of Samson dataset and Jasper Ridge dataset, respectively. The first column is the ground truth, and the other six columns correspond to the abundance maps estimated by different methods. It can be seen that the GLA has a competitive abundance estimation.

Table III shows MSE and standard deviation of abundance vectors in data1-edm3, data1-edm4, ..., data1-edm10 dataset. In more than half of the cases, GLA method obtains better abundance estimations and all standard deviations are less than



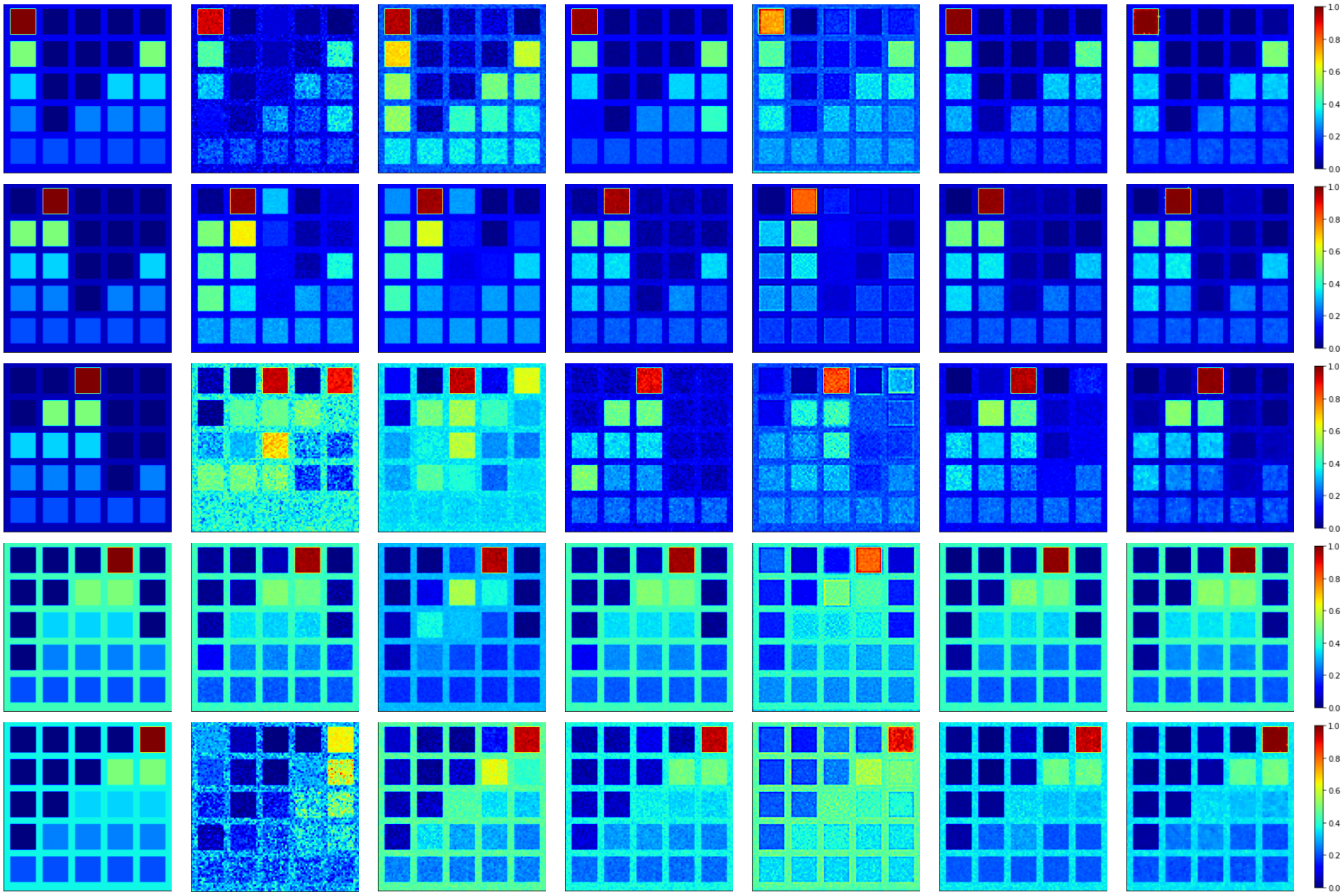
(a) reference (b) NMF-$L_{1/2}$ (c) NMF-anls (d) NMF-QMV (e) CNNAEU (f) AAS (g) GLA

Fig. 6. The abundance estimation of Jasper Ridge data set is compared with six methods. The first column is the reference abundance.



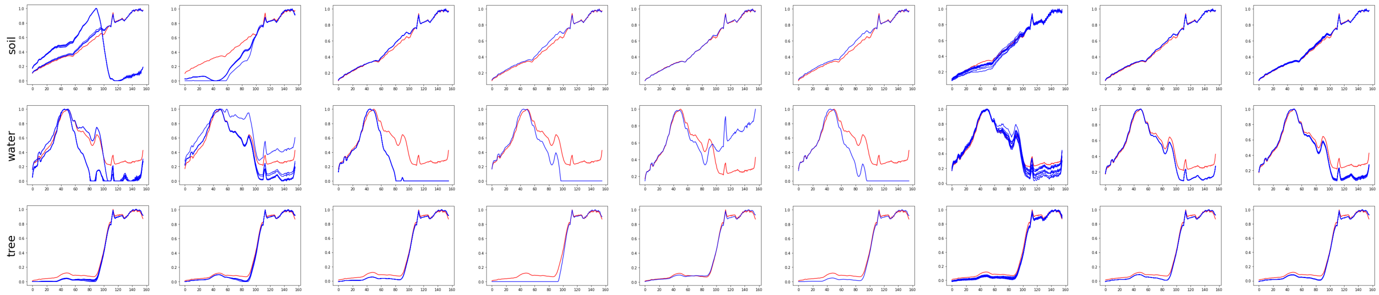
(a) reference (b) NMF-$L_{1/2}$ (c) NMF-anls (d) NMF-QMV (e) CNNAEU (f) AAS (g) GLA

Fig. 7. The abundance estimation of data1-edm3 data set is compared with six methods. The first column is the reference abundance.



(a) reference (b) NMF-$L_{1/2}$ (c) NMF-anls (d) NMF-QMV (e) CNNAEU (f) AAS (g) GLA

Fig. 8. The abundance estimation of data2-edm5 set is compared with six methods. The first column is the reference abundance.



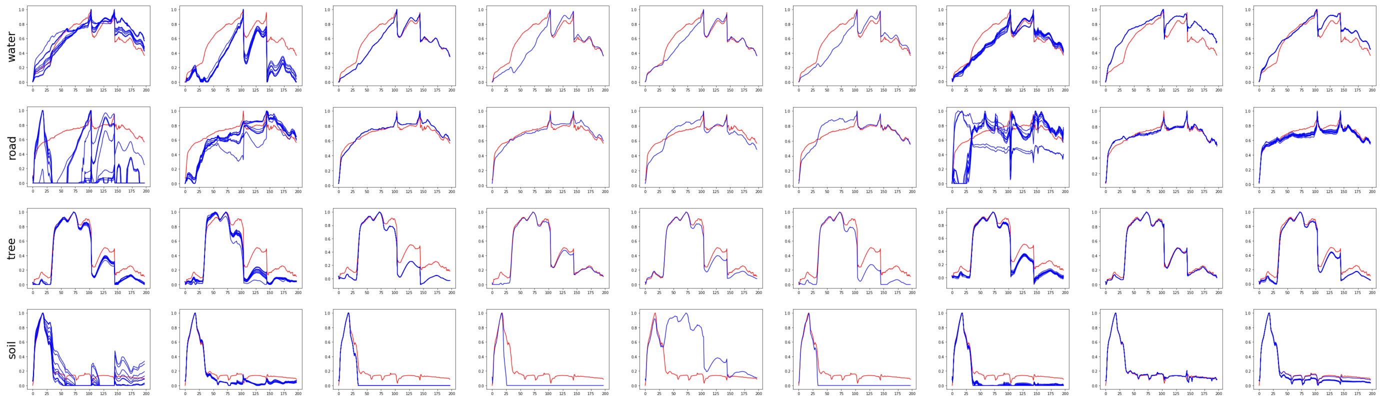
(a) NMF-$L_{1/2}$ (b) NMF-anls (c) NMF-QMV (d) MVES (e) HyperCSI (f) HISUN (g) CNNAEU (h) AAS (i) GLA

Fig. 9. Nine methods are used to extract and compare endmembers from Samson data set, and each method is run 10 times.

0.0028. And when the number of endmembers is 3, 4, 8, the NMF-QMV method performs better. Fig.7 shows the abundance map when the number of endmembers is 3 and most of the methods have good estimation results.

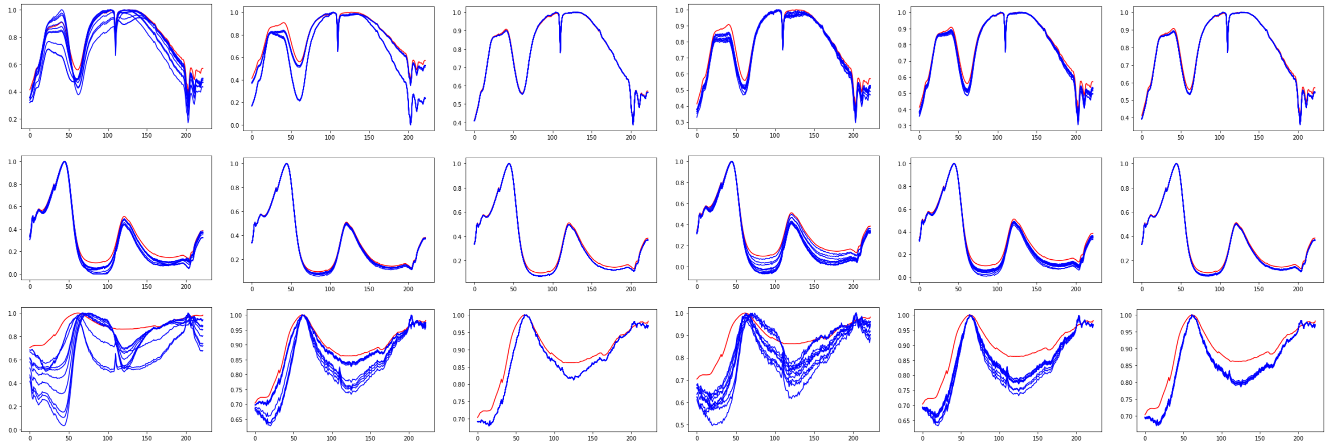
Table IV shows the MSE and standard deviation of abundance vectors in data2-edm3, data2-edm4, ..., data2-edm10 dataset. In most cases, NMF-QMV achieves better results. Only when the number of endmembers is 5 or 8, GLA performs better. Fig.8 shows the abundance map when the endmember number is 5. It can be seen that NMF-QMV, AAS and GLA all have good estimation results especially for the abundance estimation of the third endmember.

Geometry-based, NMF-based and deep-learning-based unmixing methods are compared on Data3 dataset. Table V shows the abundance estimation results of MVES, HyperCSI, HISUN, NMF-QMV, AAS, CNNAEU and GLA methods. The results indicate that HISUN have a good estimation when SNR is 20db and 30db, as compared to the other geometric methods. Most methods have good abundance estimations when SNR is 40db. GLA obtains the competitive MSE results when SNR is 20db



(a) NMF-$L_{1/2}$ (b) NMF-anls (c) NMF-QMV (d) MVES (e) HyperCSI (f) HISUN (g) CNNAEU (h) AAS (i) GLA

Fig. 10. Nine methods are used to extract and compare endmembers from Jasper Ridge data set, and each method is run 10 times.

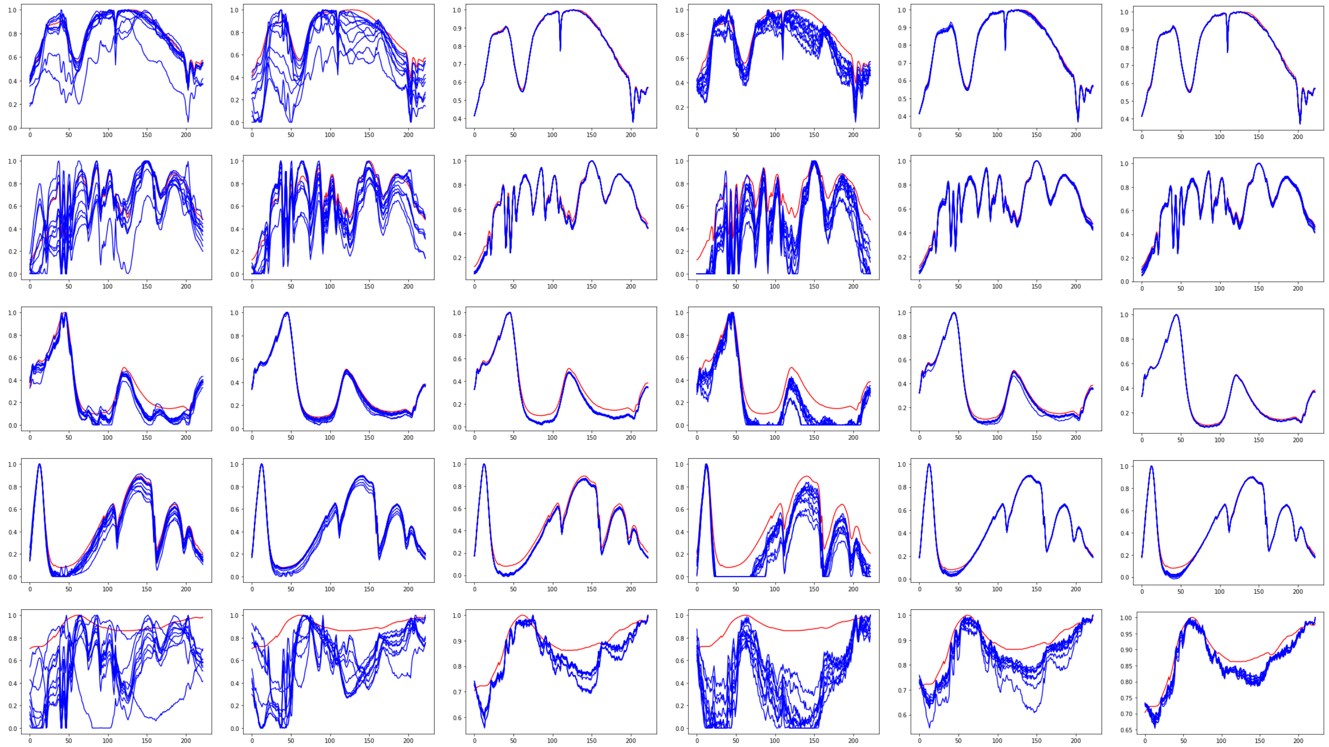


(a) NMF-$L_{1/2}$ (b) NMF-anls (c) NMF-QMV (d) CNNAEU (e) AAS (f) GLA

Fig. 11. Six methods are used to extract and compare endmembers from data1-edm3 data set, and each method is run 10 times.

TABLE III: Mean MSE and standard deviation of Data1 data set with 3-10 endmembers. The best result is bold.

	NMF-$L_{1/2}$		NMF-anls		NMF-QMV		CNNAEU		AAS		GLA	
	mean	std	mean	std	mean	std	mean	std	mean	std	mean	std
edm3	0.0423	0.0076	0.0836	0.0466	**0.0022**	0	0.0451	0.0109	0.0077	0.0010	0.0066	0.0017
edm4	0.0453	0.0093	0.0868	0.0343	**0.0048**	0	0.0919	0.0030	0.0083	0.0012	0.0053	0.0003
edm5	0.1247	0.0416	0.0843	0.0305	0.0158	0	0.1639	0.0567	0.0146	0.0015	**0.0080**	0.0004
edm6	0.1310	0.0459	0.0814	0.0560	0.0127	0	0.1623	0.038	0.0218	0.0013	**0.0078**	0.0003
edm7	0.2195	0.0338	0.1637	0.0562	0.0232	0	0.3126	0.0137	0.0331	0.0021	**0.0213**	0.0005
edm8	0.1801	0.0225	0.1903	0.0268	**0.0387**	0	0.2985	0.0334	0.0890	0.0040	0.0511	0.0028
edm9	0.2549	0.0137	0.1722	0.0206	0.0645	0	0.3097	0.0778	0.0930	0.0003	**0.0626**	0.0027
edm10	0.2622	0.0241	0.1495	0.0386	0.0627	0	0.3136	0.0401	0.0775	0.0016	**0.0621**	0.0028



(a) NMF-$L_{1/2}$ (b) NMF-anls (c) NMF-QMV (d) CNNAEU (e) AAS (f) GLA

Fig. 12. Six methods are used to extract and compare endmembers from data2-edm5 data set, and each method is run 10 times.

TABLE IV: Mean MSE and standard deviation of Data2 data set with 3-10 endmembers. The best result is bold.

	NMF-$L_{1/2}$		NMF-anls		NMF-QMV		CNNAEU		AAS		GLA	
	mean	std	mean	std	mean	std	mean	std	mean	std	mean	std
edm3	0.1073	0.0503	0.0538	0.0326	**0.0011**	0	0.1543	0.0191	0.0026	0.0001	0.0045	0.0004
edm4	0.0078	0.0069	0.0240	0.0106	**0.0022**	0	0.0569	0.0123	0.0029	0.0002	0.0032	0.0002
edm5	0.1041	0.0248	0.0636	0.0110	0.0119	0	0.0811	0.0082	0.0067	0.0003	**0.0051**	0.0010
edm6	0.0369	0.0083	0.0438	0.0233	**0.0031**	0	0.0582	0.0044	0.0066	0.0008	0.0039	0.0001
edm7	0.0793	0.0082	0.0588	0.0159	**0.0131**	0	0.0753	0.0023	0.0271	0.0009	0.0224	0.0029
edm8	0.0591	0.0115	0.0687	0.0033	0.0184	0	0.0827	0.0269	0.0450	0.0017	**0.0170**	0.0030
edm9	0.0705	0.0049	0.0751	0.0096	**0.0254**	0	0.0626	0.0054	0.0387	0.0013	0.0261	0.0006
edm10	0.0821	0.0042	0.0856	0.0043	**0.0479**	0	0.0880	0.0013	0.0543	0.0003	0.0540	0.0034

and 30db, and the MSE is 0.0129 and 0.1125, respectively. When SNR is 40db, HyperCSI achieves a good estimation.

C. *Endmember experimental results*

Fig.9 shows the endmembers extracted by all methods and their reference endmembers for Samson data. The blue curves represent the endmember extraction results of 10 runs, and the red curve is the reference endmember. As can be seen from Fig.9, CNNAEU, AAS, and GLA are closed to the ground truth. Table VI shows the SAD of Samson dataset. It can be observed that CNNAEU, AAS and GLA have similar estimation results and obtain good results on water endmember, tree endmember and soil endmember, respectively. Table VIII shows the endmember results on Cuprite data set. It can be seen that HISUN has good estimation on Chalcedony, Buddingtonite and Pyrope endmembers, NMF-QMV has good performance on Montmorillonite, Hematite and Andradite endmembers, and the proposed GLA obtains competitive SAD results in the estimation of the Muscovite, Kaolinite and Alunite endmembers. Besides, HyperCSI has good estimations for Chalcedony and Hematite endmembers of Cuprite data set.

Fig.10 shows the endmembers extraction for Jasper Ridge dataset. It can be seen that most methods have good estimations. Table VII shows the SAD and standard deviation of endmember vectors. It can be seen that GLA obtains a competitive SAD result of 0.1056 on water endmember.

TABLE V: Endmember and abundance results of Data3. The best result is bold.

	SNR=20db		SNR=30db		SNR=40db	
	edm(mean±std)	abd(mean±std)	edm(mean±std)	abd(mean±std)	edm(mean±std)	abd(mean±std)
MVES	0.3124±0	0.1293±0	0.3899±0	0.1328±0	0.0760±0	0.1425±0
HyperCSI	0.3887±0	0.0999±0	0.2796±0	0.0430±0	**0.0202**±0	**0.0297**±0
HISUN	0.2004±0	0.0459±0	0.2408±0	0.0438±0	0.0391±0	0.0407±0
NMF-QMV	0.2050±0	0.0805±0	0.2736±0	0.0479±0	0.0454±0	0.0463±0
AAS	**0.0639**±0.0002	0.0140±0.0002	0.1262±0.0049	0.0511±0.0022	0.0608±0.0024	0.0476±0.0052
CNNAEU	0.2642±0.0702	0.1677±0.0367	0.2315±0.0349	0.2859±0.0453	0.0908±0.0232	0.1983±0.0613
GLA	0.0667±0.0011	**0.0129**±0.0008	**0.1125**±0.0105	**0.0423**±0.0104	0.0357±0.0064	0.0448±0.0030

TABLE VI: Mean SAD and standard deviation of Samson data set. The best result is bold.

	NMF-$L_{1/2}$		NMF-anls		NMF-QMV		CNNAEU		AAS		GLA	
	mean	std	mean	std	mean	std	mean	std	mean	std	mean	std
soil	0.6744	0.4102	0.3274	0.1329	0.0337	0	0.0445	0.0177	0.0220	0.0004	**0.0215**	0.0050
water	0.4055	0.0933	0.3388	0.2893	0.5338	0	**0.1203**	0.0510	0.1582	0.0016	0.1580	0.0103
tree	0.0828	0.0057	0.0563	0.0052	0.0730	0	0.0690	0.0112	**0.0578**	0.0002	0.0590	0.0016

TABLE VII: Mean SAD and standard deviation of Jasper Ridge data set. The best result is bold.

	NMF-$L_{1/2}$		NMF-anls		NMF-QMV		CNNAEU		AAS		GLA	
	mean	std	mean	std	mean	std	mean	std	mean	std	mean	std
water	0.2329	0.1714	0.5180	0.2684	0.1697	0.0039	0.3478	0.3676	0.1582	0.0001	**0.1056**	0.0011
road	0.7929	0.3960	0.2612	0.0293	**0.0493**	0.0006	0.3039	0.0398	0.0496	0.0017	0.0539	0.0119
tree	0.1706	0.0092	0.2978	0.0452	0.2709	0.0021	0.1956	0.0124	**0.0322**	0.0001	0.1005	0.0020
soil	0.3327	0.0393	0.2532	0.0057	0.4158	0.0006	0.3622	0.0217	**0.0620**	0.0003	0.1548	0.0389

TABLE VIII: Endmember results of Cuprite. The best result is bold.

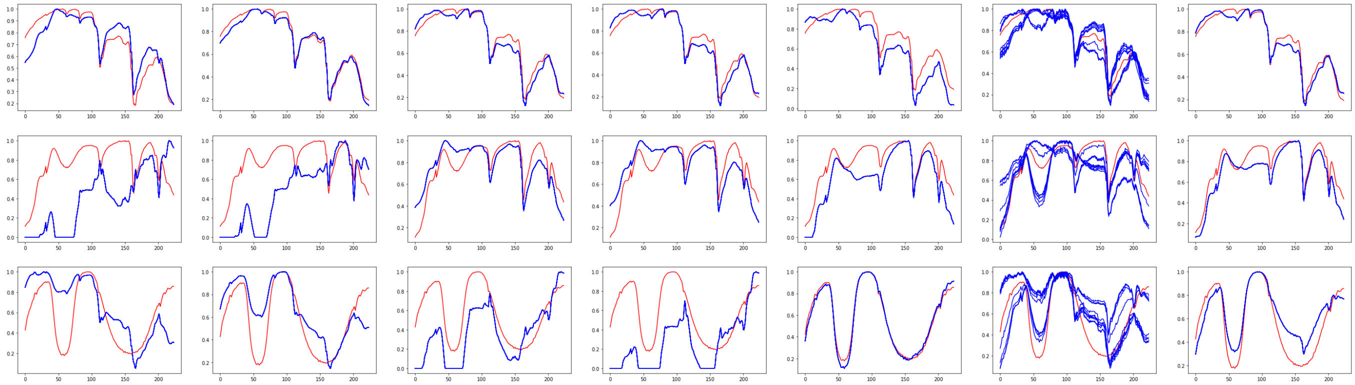
	Chalcedony	Montmorillonite	Buddingtonite	Muscovite	Hematite	Kaolinite	Alunite	Pyrope	Andradite
HISUN	**0.0866**	0.0708	**0.0560**	0.0741	0.1367	0.1596	0.1147	**0.0566**	0.1044
NMF-QMV	0.1661	**0.0518**	0.1082	0.0984	**0.1126**	0.1272	0.1456	0.1218	**0.0750**
GLA	0.1484	0.0694	0.0634	**0.0740**	0.1715	**0.0859**	**0.1034**	0.1021	0.1032
MVES	0.3441	0.1429	0.2518	0.0802	0.3832	0.4128	0.1751	0.8239	0.8345
HyperCSI	0.0923	0.0772	0.1044	0.1165	0.1550	0.2370	0.1276	0.3250	0.1546

TABLE IX: Mean SAD and standard deviation of Data1 with three endmembers. The best result is bold.

	NMF-$L_{1/2}$		NMF-anls		NMF-QMV		CNNAEU		AAS		GLA	
	mean	std	mean	std	mean	std	mean	std	mean	std	mean	std
1st endmember	0.0745	0.0304	0.1196	0.0814	**0.0058**	0	0.0390	0.0080	0.0273	0.0073	0.0151	0.0010
2nd endmember	0.1064	0.0310	**0.0272**	0.0095	0.0388	0	0.1736	0.0596	0.0851	0.0255	0.0345	0.0042
3rd endmember	0.2181	0.0948	0.0306	0.0193	**0.0237**	0	0.0851	0.0231	0.0395	0.0076	0.0273	0.0012

TABLE X: Mean SAD and standard deviation of Data2 with five endmembers. The best result is bold.

	NMF-$L_{1/2}$		NMF-anls		NMF-QMV		CNNAEU		AAS		GLA	
	mean	std	mean	std	mean	std	mean	std	mean	std	mean	std
1st endmember	0.2020	0.3214	0.2942	0.3450	**0.1192**	0.3401	0.2145	0.3082	0.1213	0.3392	0.1218	0.3390
2nd endmember	0.1945	0.0866	0.1896	0.0493	0.0391	0.0061	0.3563	0.1607	0.0273	0.0055	**0.0260**	0.0109
3rd endmember	0.1678	0.0289	0.0464	0.0252	0.1100	0.0098	0.3212	0.0601	0.0492	0.0157	**0.0199**	0.0024
4th endmember	0.1003	0.0371	0.0322	0.0336	0.0711	0.0046	0.3466	0.2847	**0.0304**	0.0061	0.0481	0.0078
5th endmember	0.4650	0.2927	0.3574	0.0772	0.1632	0.3420	0.5578	0.1432	0.0617	0.0625	**0.0476**	0.0664



(a) MVES (b) HyperCSI (c) HISUN (d) NMF-QMV (e) AAS (f) CNNAEU (g) GLA

Fig. 13. Seven methods are used to extract and compare endmembers from data3 when SNR is 30db, and each method is run 10 times.

Fig.11 shows the endmembers extraction for data1-edm3 data. All the methods have good estimations for first two endmembers. Table IX shows the SAD and standard deviation of Data1 with three endmembers. It can be observed that NMF-QMV and GLA have similar SAD results with low standard deviation on the third endmember. Fig.12 shows the endmembers extraction for data2-edm5 data. It can be seen that GLA has low fluctuations in the estimation of the fifth endmember. Table X shows the SAD and standard deviation of Data2 with five endmembers. It can be observed that GLA has competitive results in the estimation of the second, third and fifth endmember. Fig.13 shows the estimated endmember vectors on Data3 when SNR is 30db. It can be seen that all the methods have a good endmember estimation for the first endmember. And GLA obtains a competitive SAD result in the estimation of the first and second endmembers.

D. Ablation experiment

In order to test whether both LCS and GRS modules improve the results, ablation experiments on abundance estimation are shown in this section. To test the effectiveness of the LCS module, the GRS module is removed to the GLA as a comparison. Similarly, to test the effectiveness of the GRS module, only GRS module is used in the GLA method. Finally, in order to examine whether the combination of the two modules is better than the experiment of adding only one module, the GLA method is compared with the above two experimental results.

TABLE XI: Mean MSE and standard deviation of Samson and Jasper Ridge data set in Ablation Experiment. The best result is bold.

	AAS		LCS		GRS		GLA	
	mean	std	mean	std	mean	std	mean	std
Samson	0.0360	0.0011	0.0315	0.0009	0.0372	0.0036	**0.0279**	0.0019
Jasper Ridge	0.0678	0.0157	0.0519	0.0052	0.0636	0.0033	**0.0386**	0.0028

Table XI shows the MSE and standard deviation of ablation experiment in the Samson data set. The first column is the results of AAS. The second and third columns are the experimental results of removing only LCS module and GRS module, respectively. The last column is the result of GLA. It can be seen that, compared with AAS, the MSE of only having LCS module is reduced, but not as much as the proposed method.

TABLE XII: Mean MSE and standard deviation of Data1 data set with 3-10 endmembers in Ablation Experiment. The best result is bold.

	AAS		LCS		GRS		GLA	
	mean	std	mean	std	mean	std	mean	std
edm3	0.0077	0.0010	0.0078	0.0008	0.0140	0.0010	**0.0066**	0.0017
edm4	0.0083	0.0012	0.0089	0.0005	0.0147	0.0005	**0.0053**	0.0003
edm5	0.0146	0.0015	0.0131	0.0005	0.0231	0.0007	**0.0080**	0.0004
edm6	0.0218	0.0013	0.0125	0.0017	0.0241	0.0007	**0.0078**	0.0003
edm7	0.0331	0.0021	0.0387	0.0119	0.0538	0.0054	**0.0213**	0.0005
edm8	0.0890	0.0040	0.0660	0.0070	0.1030	0.0098	**0.0511**	0.0028
edm9	0.0931	0.0003	0.0822	0.0013	0.1182	0.0058	**0.0626**	0.0027
edm10	0.0775	0.0016	0.0657	0.0038	0.1012	0.0004	**0.0621**	0.0028

TABLE XIII: Mean MSE and standard deviation of Data2 data set with 3-10 endmembers in Ablation Experiment. The best result is bold.

	AAS		LCS		GRS		GLA	
	mean	std	mean	std	mean	std	mean	std
edm3	**0.0026**	0.0001	0.0046	0.0001	0.0047	0.0003	0.0045	0.0004
edm4	0.0029	0.0002	**0.0021**	0.0002	0.0025	0.0004	0.0032	0.0002
edm5	0.0067	0.0004	**0.0048**	0.0010	0.0184	0.0026	0.0051	0.0011
edm6	0.0066	0.0008	**0.0025**	0	0.0068	0.0009	0.0039	0.0001
edm7	0.0271	0.0009	**0.0159**	0.0015	0.0247	0.0010	0.0224	0.0029
edm8	0.0452	0.0017	**0.0144**	0.0006	0.0226	0.0029	0.0170	0.0030
edm9	0.0387	0.0013	**0.0227**	0.0003	0.0358	0.0025	0.0261	0.0006
edm10	0.0543	0.0003	0.0906	0.0138	0.0629	0.0039	**0.0540**	0.0034

Table XI shows the ablation experiment in Jasper Ridge data set. It can be seen that adding only LCS module or GRS module can improve the effect of abundance extraction, but the smallest MSE is still obtained by the proposed method.

Table XII shows the MSE and standard deviation of the ablation experiment on data1-edm3, data1-edm4, ..., data1-edm10 datasets. It can be seen that the MSE decreases after adding LCS module, except for data1-edm3, data1-edm4 and data1-edm7. The MSE becomes larger when any module of GLA is removed. Furthermore, the proposed method performs best on all data.

As for the ablation experiment on data2-edm3, data2-edm4, ..., data2-edm10 datasets in Table XIII, there is an unexpected discovery. Except for the data2-edm3 data set, the estimation result of GLA is better than that of the original method. However, only adding LCS module is not only better than AAS method, but also better than the proposed method.

This phenomenon can be explained from the way of the data generation. In the process of data2 generation, the abundance map is defined in advance, which divides the whole abundance map into several sub-blocks, and each sub-block is independent with each other. Thus the pixel is closely related to the adjacent pixels, but has little similarities with the distant pixels. This may lead to the accuracy of the abundance estimation using only LCS module combined with local information higher than GLA.

In the process of data1 generation, the whole hyperspectral image is divided into different sub-regions. Given the endmembers of each sub-region, the hyperspectral image is filtered to form mixed pixels. At this time, different sub-regions are related to each other. And because the endmembers of sub-regions are randomly determined, the spatial correlation of data1 is stronger than that of data2. It is better for the networks to extract local information and global information than to estimate abundance by any one of them.

IV. Conclusion

In this paper, a hyperspectral unmixing model with joint spatial-spectral information is proposed to explore the local homogeneity and global self-similarity of the hyperspectral imagery. Firstly, based on the conditional probability model, the LCS module is developed to smooth local homogeneity by generating adaptive weights. Secondly, considering the self-similarity of the hyperspectral imagery, GRS module is designed to extract the long-distance relationship between all pixels. In the experimental section, we compare the proposed method with five different methods on real and synthetic datasets, respectively. The experimental results show that the proposed method has competitive performance in abundance estimation.

In the future, we will try to consider the estimation of endmember vector fluctuation in the proposed method to deal with the endmember variability task.

References

[1] C. I. Chang and A. Plaza, "A fast iterative algorithm for implementation of pixel purity index," *IEEE Geoscience and Remote Sensing Letters*, vol. 3, no. 1, pp. 63–67, 2006.

[2] H.-C. Li, "An algorithm for fast spectral endmember determination in hyperspectral data," in *IGARSS 2018-2018 IEEE International Geoscience and Remote Sensing Symposium*. IEEE, 2018, pp. 2689–2692.

[3] C.-H. Lin and J. M. Bioucas-Dias, "Nonnegative blind source separation for ill-conditioned mixtures via john ellipsoid," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 32, no. 5, pp. 2209–2223, 2020.

[4] J. M. Nascimento and J. M. Dias, "Does independent component analysis play a role in unmixing hyperspectral data?" *IEEE Transactions on Geoscience and Remote Sensing*, vol. 43, no. 1, pp. 175–187, 2005.

[5] S. Jia and Y. Qian, "Constrained nonnegative matrix factorization for hyperspectral unmixing," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 47, no. 1, pp. 161–173, 2008.

[6] Y. Yuan, Z. Zhang, and Q. Wang, "Improved collaborative non-negative matrix factorization and total variation for hyperspectral unmixing," *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, vol. 13, pp. 998–1010, 2020.

[7] S. Ozkan, B. Kaya, and G. B. Akar, "Endnet: Sparse autoencoder network for endmember extraction and hyperspectral unmixing," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 57, no. 1, pp. 482–496, Jan 2019.

[8] R. Guo, W. Wang, and H. Qi, "Hyperspectral image unmixing using autoencoder cascade," in *2015 7th Workshop on Hyperspectral Image and Signal Processing: Evolution in Remote Sensing (WHISPERS)*, June 2015.

[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] ——, "Collaborative sparse regression for hyperspectral unmixing," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 52, no. 1, pp. 341–354, 2013.

[11] L. Zhuang, C. Lin, M. A. T. Figueiredo, and J. M. Bioucas-Dias, "Regularization parameter selection in minimum volume hyperspectral unmixing," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 57, no. 12, pp. 9858–9877, Dec 2019.

[12] Z. Li, Y. Altmann, J. Chen, S. Mclaughlin, and S. Rahardja, "Sparse spectral unmixing of hyperspectral images using expectation-propagation," in *2020 IEEE International Conference on Visual Communications and Image Processing (VCIP)*, 2020, pp. 197–200.

[13] C. Shi and L. Wang, "Incorporating spatial information in spectral unmixing: A review," *Remote Sensing of Environment*, vol. 149, pp. 70–87, 2014.

[14] X. Xu, J. Li, C. Wu, and A. Plaza, "Regional clustering-based spatial preprocessing for hyperspectral unmixing," *Remote Sensing of Environment*, vol. 204, pp. 333–346, 2018.

[15] X. Tao, T. Cui, Z. Yu, and P. Ren, "Locality preserving endmember extraction for estimating green algae area," in *2018 OCEANS - MTS/IEEE Kobe Techno-Oceans (OTO)*, 2018.

[16] X. Wang, Y. Zhong, L. Zhang, and Y. Xu, "Spatial group sparsity regularized nonnegative matrix factorization for hyperspectral unmixing," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 55, no. 11, pp. 6287–6304, 2017.

[17] M. 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, Nov 2012.

[18] S. Zhang, J. Li, H. Li, C. Deng, and A. Plaza, "Spectral–spatial weighted sparse regression for hyperspectral image unmixing," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 56, no. 6, pp. 3265–3276, June 2018.

[19] R. Wang, H. Li, W. Liao, and A. Pižurica, "Double reweighted sparse regression for hyperspectral unmixing," in *2016 IEEE International Geoscience and Remote Sensing Symposium (IGARSS)*, July 2016, pp. 6986–6989.

[20] A. Lagrange, M. Fauvel, S. May, and N. Dobigeon, "Matrix cofactorization for joint spatial–spectral unmixing of hyperspectral images," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 58, no. 7, pp. 4915–4927, July 2020.

[21] S. Zhang, J. Li, Z. Wu, and A. Plaza, "Spatial discontinuity-weighted sparse unmixing of hyperspectral images," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 56, no. 10, pp. 5767–5779, Oct 2018.

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

[23] F. Xiong, J. Chen, J. Zhou, and Y. Qian, "Superpixel-based nonnegative tensor factorization for hyperspectral unmixing," in *IGARSS 2018 - 2018 IEEE International Geoscience and Remote Sensing Symposium*, July 2018, pp. 6392–6395.

[24] L. Yang, J. Peng, H. Su, L. Xu, Y. Wang, and B. Yu, "Combined nonlocal spatial information and spatial group sparsity in nmf for hyperspectral unmixing," *IEEE Geoscience and Remote Sensing Letters*, vol. 17, no. 10, pp. 1767–1771, Oct 2020.

[25] M. Q. Alkhatib and M. Velez-Reyes, "Effects of region size on superpixel-based unmixing," in *2019 10th Workshop on Hyperspectral Imaging and Signal Processing: Evolution in Remote Sensing (WHISPERS)*, Sep. 2019.

[26] ——, "Superpixel-based hyperspectral unmixing with regional segmentation," in *IGARSS 2018 - 2018 IEEE International Geoscience and Remote Sensing Symposium*, July 2018, pp. 6384–6387.

[27] H. Li, R. Feng, L. Wang, Y. Zhong, and L. Zhang, "Superpixel-based reweighted low-rank and total variation sparse unmixing for hyperspectral remote sensing imagery," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 59, no. 1, pp. 629–647, Jan 2021.

[28] D. D. Lee and H. S. Seung, "Algorithms for non-negative matrix factorization," in *International Conference on Neural Information Processing Systems*, 2000.

[29] Y. Su, J. Li, A. Plaza, A. Marinoni, P. Gamba, and S. Chakravortty, "Daen: Deep autoencoder networks for hyperspectral unmixing," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 57, no. 7, pp. 4309–4321, July 2019.

[30] Y. Qu and H. Qi, "udas: An untied denoising autoencoder with sparsity for spectral unmixing," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 57, no. 3, pp. 1698–1712, March 2019.

[31] Y. Su, A. Marinoni, J. Li, J. Plaza, and P. Gamba, "Stacked nonnegative sparse autoencoders for robust hyperspectral unmixing," *IEEE Geoscience and Remote Sensing Letters*, vol. 15, no. 9, pp. 1427–1431, Sep. 2018.

[32] M. Wang, M. Zhao, J. Chen, and S. Rahardja, "Nonlinear unmixing of hyperspectral data via deep autoencoder networks," *IEEE Geoscience and Remote Sensing Letters*, vol. 16, no. 9, pp. 1467–1471, Sep. 2019.

[33] Y. Su, X. Xu, J. Li, H. Qi, P. Gamba, and A. Plaza, "Deep autoencoders with multitask learning for bilinear hyperspectral unmixing," *IEEE Transactions on Geoscience and Remote Sensing*, 2020.

[34] Z. Dou, K. Gao, X. Zhang, H. Wang, and J. Wang, "Blind hyperspectral unmixing using dual branch deep autoencoder with orthogonal sparse prior," in *ICASSP 2020 - 2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, May 2020, pp. 2428–2432.

[35] L. Miao and H. Qi, "A constrained non-negative matrix factorization approach to unmix highly mixed hyperspectral data," in *2007 IEEE International Conference on Image Processing*, vol. 2, Sep. 2007, pp. II – 185–II – 188.

[36] Z. Hua, X. Li, Q. Qiu, and L. Zhao, "Autoencoder network for hyperspectral unmixing with adaptive abundance smoothing," *IEEE Geoscience and Remote Sensing Letters*, 2020.

[37] J. Yao, D. Hong, L. Xu, D. Meng, J. Chanussot, and Z. Xu, "Sparsity-enhanced convolutional decomposition: A novel tensor-based paradigm for blind hyperspectral unmixing," *IEEE Transactions on Geoscience and Remote Sensing*, 2021.

[38] Y. Ranasinghe, S. Herath, K. Weerasooriya, M. Ekanayake, R. Godaliyadda, P. Ekanayake, and V. Herath, "Convolutional autoencoder for blind hyperspectral image unmixing," in *2020 IEEE 15th International Conference on Industrial and Information Systems (ICIIS)*, 2020, pp. 174–179.

[39] B. Palsson, M. O. Ulfarsson, and J. R. Sveinsson, “Convolutional autoencoder for spatial-spectral hyperspectral unmixing,” in *IGARSS 2019 - 2019 IEEE International Geoscience and Remote Sensing Symposium*, July 2019, pp. 357–360.

[40] F. Khajehrayeni and H. Ghassemian, “Hyperspectral unmixing using deep convolutional autoencoders in a supervised scenario,” *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, vol. 13, pp. 567–576, 2020.

[41] X. Zhang, Y. Sun, J. Zhang, P. Wu, and L. Jiao, “Hyperspectral unmixing via deep convolutional neural networks,” *IEEE Geoscience and Remote Sensing Letters*, vol. 15, no. 11, pp. 1755–1759, 2018.

[42] L. Dong, Y. Yuan, and X. Luxs, “Spectral–spatial joint sparse nmf for hyperspectral unmixing,” *IEEE Transactions on Geoscience and Remote Sensing*, vol. 59, no. 3, pp. 2391–2402, 2021.

[43] A. Danielyan, V. Katkovnik, and K. Egiazarian, “Bm3d frames and variational image deblurring,” *IEEE Transactions on Image Processing*, vol. 21, no. 4, pp. 1715–1728, 2012.

[44] K. Ristovski, V. Radosavljevic, S. Vucetic, and Z. Obradovic, “Continuous conditional random fields for efficient regression in large fully connected graphs,” in *AAAI*, 2013.

[45] D. Chen, D. Xu, H. Li, N. Sebe, and X. Wang, “Group consistent similarity learning via deep crf for person re-identification,” in *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, June 2018, pp. 8649–8658.

[46] S. Bell, C. L. Zitnick, K. Bala, and R. Girshick, “Inside-outside net: Detecting objects in context with skip pooling and recurrent neural networks,” in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016, pp. 2874–2883.

[47] L. Mou, P. Ghamisi, and X. X. Zhu, “Deep recurrent neural networks for hyperspectral image classification,” *IEEE Transactions on Geoscience and Remote Sensing*, vol. 55, no. 7, pp. 3639–3655, 2017.

[48] B. Palsson, J. Sigurdsson, J. R. Sveinsson, and M. O. Ulfarsson, “Hyperspectral unmixing using a neural network autoencoder,” *IEEE Access*, vol. 6, pp. 25 646–25 656, 2018.

[49] Y. Qian, S. Jia, J. Zhou, and A. Robles-Kelly, “Hyperspectral unmixing via $l_{1/2}$ sparsity-constrained nonnegative matrix factorization,” *IEEE Transactions on Geoscience and Remote Sensing*, vol. 49, no. 11, pp. 4282–4297, Nov 2011.

[50] J. Kim, Y. He, and H. Park, “Algorithms for nonnegative matrix and tensor factorizations: a unified view based on block coordinate descent framework,” *Journal of Global Optimization*, vol. 58, no. 2, pp. 285–319, 2014.

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

[52] C.-H. Lin, C.-Y. Chi, Y.-H. Wang, and T.-H. Chan, “A fast hyperplane-based mves algorithm for hyperspectral unmixing,” in *2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2015, pp. 1384–1388.

[53] Z. Han, D. Hong, L. Gao, B. Zhang, and J. Chanussot, “Deep half-siamese networks for hyperspectral unmixing,” *IEEE Geoscience and Remote Sensing Letters*, 2020.

[54] J. M. P. Nascimento and J. M. B. Dias, “Vertex component analysis: a fast algorithm to unmix hyperspectral data,” *IEEE Transactions on Geoscience and Remote Sensing*, vol. 43, no. 4, pp. 898–910, April 2005.

[55] D. C. Heinz and Chein-I-Chang, “Fully constrained least squares linear spectral mixture analysis method for material quantification in hyperspectral imagery,” *IEEE Transactions on Geoscience and Remote Sensing*, vol. 39, no. 3, pp. 529–545, March 2001.