

SNMF-Net: Learning a Deep Alternating Neural Network for Hyperspectral Unmixing

Fengchao Xiong, *Member, IEEE*, Jun Zhou, *Senior Member, IEEE*, Shuyin Tao, Jianfeng Lu, *Member, IEEE* and Yuntao Qian, *Senior Member, IEEE*

Abstract—Hyperspectral unmixing is recognized as an important tool to learn the constituent materials and corresponding distribution in a scene. The physical spectral mixture model is always important to tackle this problem because of its highly ill-posed nature. In this paper, we introduce a linear spectral mixture model (LMM) based end-to-end deep neural network named as SNMF-Net for hyperspectral unmixing. SNMF-Net shares an alternating architecture and benefits from both model-based methods and learning-based methods. On one hand, SNMF-Net is of high physical interpretability as it is built by unrolling L_p sparsity constrained nonnegative matrix factorization (L_p -NMF) model belonging to LMM families. On the other hand, all the parameters and submodules of SNMF-Net can be seamlessly linked with alternating optimization algorithm of L_p -NMF and unmixing problem. This enables to reasonably integrate the prior knowledge on unmixing, optimization algorithm, and sparse representation theory into the network for robust learning so as to improve unmixing. Experimental results on the synthetic and real-world data show the advantages of the proposed SNMF-Net over many state-of-the-art methods.

Index Terms—Hyperspectral unmixing, nonnegative matrix factorization (NMF), model-based neural network, sparse representation.

I. INTRODUCTION

Hyperspectral image (HSI) not only captures the spatial information but also the spectral information indexed by numerous contiguous narrow bands of the object. These wealthy spectral bands offer fine details on the material information of the scene and dramatically augment HSIs discriminative ability. Benefiting from this, HSIs have enabled many practical applications such as object recognition [1], object tracking [2] and medical analysis [3], [4], just to name a few. However, HSIs suffer from low spatial resolution caused by the limitation of hyperspectral sensors, leading to many "mixed pixels" containing more than one material. Hyperspectral unmixing aims to decompose every pixel into the combination of constituent materials, i.e., endmembers and their fractional abundances. It provides an attractive way to tackle this problem and greatly improves the utility of HSIs.

The unmixing problem can be generally considered as an ill-posed inverse problem recovering endmembers and abundances from acquired HSIs. Prior knowledge is very important to address this problem. Traditional model-based methods make elaborate hypotheses on the physical interaction between light and the target surface, i.e., the spectral mixture model, and therefore are highly interpretable. As a simplified model, the linear spectral mixture model (LMM) assumes the observed light interacts with only one material and linearly represents a pixel with endmembers [5]. Depended on LMM, the unmixing can be sequentially decomposed in two stages, i.e., endmember extraction such as vertex component analysis (VCA) [6] and abundance estimation using regression, e.g., fully constrained least squares (FCLS) [7]. Alternatively, this problem can be regarded as a blind source separation problem that simultaneously infers the endmembers and abundances. Formulating unmixing as a matrix factorization problem, non-negative matrix factorization (NMF) decomposes an HSI into the product of two factor matrices to perform unmixing [8]–[10]. Thanks to the inherent advantages of learning parts-based representation of the data, the two matrices are respectively reasonably interpreted as endmembers and abundances in LMM. The nonconvex objective function of NMF makes its solution space very huge. Many extended NMFs embed several kinds of regularization terms encoding certain prior knowledge such as sparsity, spatial information, and spatial-spectral information into the NMF model to limit the number of trivial solutions [11]–[17]. Instead of a two-dimensional matrix, nonnegative tensor factorization (NTF) directly treats an HSI as a third-order tensor for unmixing [18], [19]. Albeit interpretability, the model-based methods entail an accurate model depicting the unmixing problem which is actually difficult to obtain. Additionally, burdensome hyperparameter setting and computationally demanding iterations also restrict their effectiveness and efficiency.

Instead of model-based unmixing, a vast majority of learning-based particularly deep neural networks (DNNs) based methods directly learn the hidden concepts by feeding a large number of training samples [20], [21]. In this way, learning-based methods are very fast. Blind unmixing can be cast as an unsupervised learning problem that jointly learns endmembers and abundances from data. Autoencoders (AEs) are mostly studied among these methods. AE comprises an encoder component mapping the input data into hidden concepts, i.e., abundances, and a decoder component recon-

This work was supported in part by the National Key Research and Development Program of China under Grant 2018YFB0505000 and 2018AAA0100500, "111" Program (No. B13022), Jiangsu Provincial Natural Science Foundation of China under Grant BK20200466, and the National Natural Science Foundation of China under Grant 62002169, 61905114 and 62071421.

F. Xiong, S. Tao and J. Lu are with the School of Computer Science and Engineering, Nanjing University of Science and Technology, Nanjing 210094, P.R. China (Corresponding author: F. Xiong; fxiong@njust.edu.cn).

J. Zhou is with the School of Information and Communication Technology, Griffith University, Nathan, Australia.

Y. Qian is with the College of Computer Science, Zhejiang University, Hangzhou 310027, China.

structing the input data by the learnable base, i.e., endmembers. Specifically, Su *et al.* employed variational autoencoder (VAE) to perform unmixing and stacked denoising AEs to generate favorable initialization for subsequent VAE [22]. Instead of Euclidian distance, EndNet used spectral angle distances in the loss function to more consider the nonlinearity of endmembers [23]. Min *et. al* [24] further introduced the Wasserstein distance as a regularization term to better consider the distribution similarity between the observation and the reconstruction. Under the observation that the abundance maps of different endmembers are nearly orthogonal, Dou *et al.* embedded orthogonal sparse prior into the AE to better consider the relationship among abundance maps [25]. Qu *et al.* introduced an untied denoising AE (uDAS) which decouples the decoder from the encoder weights for more accurate endmember extraction [26]. Moreover, cube-based convolutional AEs exploit the spatial information along with information of HSIs for spectral-spatial unmixing in which the encoder component is characterized by several convolutional layers [27], [28]. In addition to AE based unmixing, [29] and [30] learned the deep priors of the abundance maps rather than hand-crafted priors and integrated them in the unmixing framework to boost the unmixing quality. Without spectral mixture model, there is, however, still no human-interpretable way to understand why data-driven DNNs can perform unmixing, i.e., they are “black box” models. Their overparameterized architecture always demands many training samples. In practice, the high cost of collecting abundant HSIs and difficulties of obtaining ground-truth endmembers and abundances can also make the learning infeasible, dramatically hindering the generalization capability [31].

Complementary to model-based and data-driven learning-based methods, model-based DNNs embed physical models into the design of deep architectures through the deep unrolling technique. By introducing some learnable parameters, deep unrolling maps each iteration of model-based methods into a typical layer of DNNs and stacks a predefined number of layers to form a hierarchical deep network architecture. The unrolled networks naturally inherit advantages from two kinds of approaches such as high interpretability, good generalization capability, strong learning ability, and computational efficiency, overcome the difficulties of modeling the underlying physical mechanism and need much fewer training samples. Thanks to the above merits, deep unrolled networks have been successfully applied in many practical applications, such as image superresolution [32], [33], rain removal [34], image deblurring [35], [36] and semantic segmentation [37], [38].

Recently, Qian *et. al* [39] have introduced a L_1 -NMF based NN for blind unmixing (MNN-BU). MNN-BU is obtained from LMM and corresponding iterative shrinkage-thresholding algorithm (ISTA). It shares an AE structure. The encoder component is obtained by unrolling ISTA and the decoder component is an additional learnable endmember matrix set behind the encoder. The experimental results demonstrate that MNN-BU outperforms several baselines in terms of both efficiency and accuracy. Despite this, we analyze MNN-BU and find its drawbacks as follows:

- 1) The endmembers are hard-coded into the network via a learnable endmember matrix during training. This means once the network is learned, the endmembers are fixed at the testing stage, possibly causing incompatibility between endmembers and abundances, especially in complex scenarios with perturbations.
- 2) MNN-BU only implements unrolling iterative optimization procedure of abundances and composes two sequential steps: abundance estimation and reconstructing the input with a learnable endmember matrix. This makes the network short in fully absorbing the prior knowledge embodied by the model. Therefore, the network needs more parameters and training samples to cover the practical complexity of unmixing.

In order to address the above problems, this paper proposes a novel interpretable deep alternating neural network for hyperspectral unmixing. It not only equips with a more powerful physical model but also conforms to the alternating minimization of L_p -NMF. All the parameters and submodules are interpretable and can be connected with the optimization of L_p -NMF and unmixing problem. Specifically, based on LMM, we first build a non-convex L_p ($0 < p < 1$) sparse NMF model and derive its alternating iterative update rules following the proximal gradient algorithm. Thanks to the non-convex L_p sparse inducer, the model is capable of producing more accurate abundances. Afterward, the update rules of both endmembers and abundances are unrolled to yield an alternating network (SNMF-Net) including End-Net and Abun-Net submodules shown in Fig. 1. SNMF-Net is in line with the alternating optimization of L_p -NMF and no longer hard-codes the endmembers as a learnable matrix but alternatively predicts them. In this way, the network is more capable of leveraging the knowledge embodied by the model and producing more accurate endmembers and abundances at testing. Finally, all the parameters including p are learned by end-to-end learning, alleviating the hyperparameters selection and providing more adaptation to data.

The high interpretability of SNMF-Net makes it easy to understand its unmixing mechanism and combine the prior knowledge on the endmembers and abundances into the network for enhanced learning. Driven by recent theoretical research on the relationship between the network parameters and dictionaries in the learned iterative shrinkage-thresholding algorithm (LISTA) [40]–[43], we further precompute and initialize the values of weight matrices using mutual coherence minimization in sparse representation theory. This simplifies the network learning and improves the robustness of the network to noises and the effectiveness of unmixing. Enhanced physical model, advanced network architecture, and improved learning strategies help SNMF-Net dominate several alternatives methods on both synthetic data and real-world data.

This paper is organized as follows: Section II introduces the L_p sparsity-based unmixing model and its alternating iterative optimization. In Section III, we derive the network architecture of SNMF-Net and present its learning and implementation details. Section IV reports the experimental results on both synthetic and real-world data. This paper concludes in Section V.

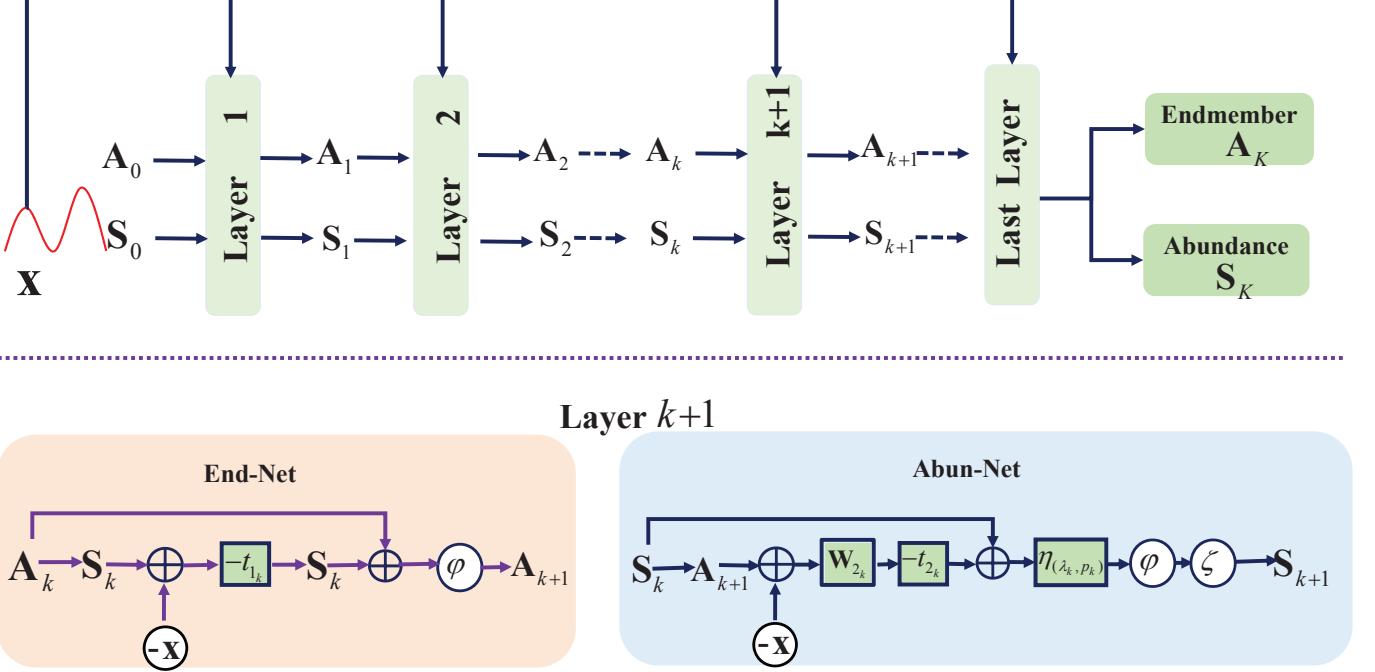


Fig. 1. The framework of proposed SNMF-Net. Based on L_p norm based sparse unmixing model, SNMF-Net takes the observed spectrum as the input and alternately predicts the endmembers and abundances using End-Net and Abun-Net submodules. All the parameters (labeled green) can be linked with the optimization of L_p -NMF and the unmixing problem and support data-driven training.

II. L_p -NMF UNMIXING MODEL

In this section, we introduce the L_p -NMF based unmixing model and derive its optimization procedure.

A. Linear Spectral Mixture Model

The linear spectral mixture model (LMM) is a widely accepted physical model describing the interaction between endmembers. LMM assumes the observed spectrum can be linearly represented by a set of constituent endmembers, whose weights are the fractional abundances. Given an HSI with L bands, LMM can be formally formulated as

$$\mathbf{x} = \mathbf{As} + \mathbf{e} \quad (1)$$

where $\mathbf{x} \in \mathbb{R}^{L \times 1}$ is the spectrum vector, $\mathbf{A} \in \mathbb{R}^{L \times R}$ is the endmember matrix with R materials, $\mathbf{s} \in \mathbb{R}^{R \times 1}$ contains the fractional abundances of the endmembers for \mathbf{x} , and $\mathbf{e} \in \mathbb{R}^{L \times 1}$ accounts for measurement errors. Using matrix notation, Eq. (1) can be reexpressed as

$$\mathbf{X} = \mathbf{AS} + \mathbf{E} \quad (2)$$

Here \mathbf{X} is an HSI containing L bands and N pixels, $\mathbf{S} \in \mathbb{R}^{R \times N}$ is its corresponding fractional abundances, and $\mathbf{E} \in \mathbb{R}^{L \times N}$ is the additional matrix factoring in noises.

Two constraints are usually added to Eq. (2), i.e., abundance nonnegativity constraint (ANC) and abundance sum-to-one constraint (ASC) for physically meaningful unmixing. ANC means the contribution of each endmember should be never negative and ASC limits the entire contributions of endmembers should equal one. Mathematically, they are given by

$$\begin{aligned} \text{ANC : } & \mathbf{S} \succeq 0 \\ \text{ASC : } & \mathbf{1}_R^T \mathbf{S} = \mathbf{1}_N^T \end{aligned} \quad (3)$$

where $\mathbf{1}$ is a vector of all ones.

B. L_p -NMF for Blind Unmixing

Nonnegative matrix factorization (NMF) is a prevailing tool to solve Eq. (2) by decomposing the observed HSI \mathbf{X} into the product of two factor matrices, respectively corresponding to endmembers (\mathbf{A}) and abundances (\mathbf{S}). Specifically, NMF casts hyperspectral unmixing as an optimization problem that minimizes the cost function

$$\mathcal{L}(\mathbf{A}, \mathbf{S}) = \frac{1}{2} \|\mathbf{X} - \mathbf{AS}\|_F^2 \text{ s.t. } \mathbf{1}_R^T \mathbf{S} = \mathbf{1}_N^T, \mathbf{A}, \mathbf{S} \succeq 0 \quad (4)$$

Due to the non-convex objective function of NMF with respect to two variables simultaneously, various priors such as sparsity and spatial structural smoothness are embedded into the formulation of NMF to narrow and regularize the solution to a plausible one [44]–[46]. In particular, abundance sparsity prior assumes that only a subset of endmembers contribute to a pixel. Accordingly, the objective function of NMF can be reformulated as

$$\mathcal{L}(\mathbf{A}, \mathbf{S}) = \frac{1}{2} \|\mathbf{X} - \mathbf{AS}\|_F^2 + \lambda f(\mathbf{S}) \text{ s.t. } \mathbf{1}_R^T \mathbf{S} = \mathbf{1}_N^T, \mathbf{A}, \mathbf{S} \succeq 0 \quad (5)$$

where $f(\cdot)$ is a sparsity promoting function and λ balances the sparsity constraint term and data fidelity term.

Generally $f(\cdot)$ can be realized by L_0 norm of \mathbf{S} which is however intractable because of its NP-hard nature. An alternative approach is to use L_p norm ($0 < p \leq 1$), resulting the following L_p -NMF based unmixing model:

$$\mathcal{L}(\mathbf{A}, \mathbf{S}) = \frac{1}{2} \|\mathbf{X} - \mathbf{AS}\|_F^2 + \lambda \|\mathbf{S}\|_p \text{ s.t. } \mathbf{1}_R^T \mathbf{S} = \mathbf{1}_N^T, \mathbf{A}, \mathbf{S} \succeq 0 \quad (6)$$

where

$$\|\mathbf{S}\|_p = \sum_{r=1}^R \sum_{n=1}^N (s_{rn})^p \quad (7)$$

C. Model Optimization

Generally, many algorithm can be used to solve Eq. (6), for example, alternating direction method of multipliers (ADMM) [47] and alternating least square (ALS) algorithm [48]. In our paper, we use ALS algorithm considering its fast convergence speed and simple implementation. ALS alternately optimizes the endmembers and abundances with the other variable fixed. Specifically, fixing \mathbf{S} , the subproblem for optimizing \mathbf{A} corresponds to

$$\mathbf{A}_{k+1} \leftarrow \min_{\mathbf{A}} \frac{1}{2} \|\mathbf{X} - \mathbf{AS}_k\|_F^2, \text{ s.t., } \mathbf{A} \succeq 0 \quad (8)$$

where k refers to the iteration number. This is a quadratic programming problem with a nonnegative constraint and can be iteratively solved through the gradient descent step for the quadratic part followed by a projection step for the nonnegative constraint. Consequently, the updating rule of \mathbf{A} can be obtained, i.e.,

$$\mathbf{A}_{k+1} \leftarrow [\mathbf{A}_k - t_{1_k} (\mathbf{A}_k \mathbf{S}_k - \mathbf{X}) \mathbf{S}_k^T]_+ \quad (9)$$

where t_{1_k} is the step size and $[\cdot]_+ = \max(\cdot, 0)$ guarantees the nonnegativity of \mathbf{A} .

Similarly, when \mathbf{A} is fixed, we arrive at the optimization problem for \mathbf{S} :

$$\mathbf{S}_{k+1} \leftarrow \min_{\mathbf{S}} \frac{1}{2} \|\mathbf{X} - \mathbf{A}_{k+1} \mathbf{S}\|_F^2 + \lambda \|\mathbf{S}\|_p, \text{ s.t., } \mathbf{S} \succeq 0, \mathbf{1}_R^T \mathbf{S} = \mathbf{1}_N^T \quad (10)$$

Eq. (10) can be decomposed into two parts, respectively accounting for nonnegative sparsity constraint and sum-to-one constraint. The solution of \mathbf{S} concerning nonnegative sparsity constraint is

$$\mathbf{S}_{k+1} \leftarrow [\text{GST}_{(\lambda, p)} (\mathbf{S}_k - t_{2_k} \mathbf{A}_{k+1}^T (\mathbf{A}_{k+1} \mathbf{S}_k - \mathbf{X}))]_+ \quad (11)$$

where t_{2_k} is also the step size. Parameterized by λ and p , $\text{GST}_{(\lambda, p)}(\cdot)$ is the generalized shrinkage thresholding (GST) operator for solving the following L_p norm regularized sparse minimization problem:

$$\min_s \frac{1}{2} (y - s)^2 + \lambda |s|^p \quad (12)$$

The close-form definition of $\text{GST}_{(\lambda, p)}(\cdot)$ is

$$s = \begin{cases} \text{sign}(y) S_p^{\text{GST}}(|y|; \lambda), & \text{if } |y| > \Delta_p^{\text{GST}}(\lambda) \\ 0, & \text{otherwise} \end{cases} \quad (13)$$

where the threshold $\Delta_p^{\text{GST}}(\lambda)$ is given by

$$\Delta_p^{\text{GST}}(\lambda) = (2\lambda(1-p))^{\frac{1}{2-p}} + \lambda p (2\lambda(1-p))^{\frac{p-1}{2-p}} \quad (14)$$

Zuo *et al.* [49] have given the solution of $\text{GST}(|y|; \lambda)$ by performing the following iteration steps

$$x_{t+1} \leftarrow |y| - \lambda p x_t^{p-1} \quad (15)$$

where $x_0 = |y|$.

After every iteration, the columns of \mathbf{S}_{k+1} are normalized to satisfy the ASC constrain, i.e.,

$$S_{rn} \leftarrow \frac{S_{rn}}{\sum_{r=1}^R S_{rn}} \quad (16)$$

Eq. (16) can also serve as a normalization to prevent the trivial solution that \mathbf{S} is extremely small while \mathbf{A} is arbitrary large.

Till now, under the framework of LMM, we have built the L_p -NMF based unmixing model and induced its corresponding iterative updating rules. Thanks to LMM, this model is highly interpretable and two factor matrices can be respectively explained as endmembers and abundances. Generally, the endmembers and abundances can be obtained by alternately executing Eq. (9), Eq. (11) and Eq. (16) for numbers of iterations, which is always computationally demanding. Moreover, selecting proper parameters such as t_{1_k} , t_{2_k} , λ and p is also troublesome but critical for this algorithm. By contrast, the deep neural network has higher learning capacity and the feed-forward process is very computationally efficient because no iteration is needed. To this end, we construct a neural network by unrolling the iterative updating rules in Eq. (9), Eq. (11) and Eq. (16) while retaining the attractive interpretation of L_p -NMF based unmixing model.

III. SNMF-NET

In this section, we first build the connection between the proximal gradient method and deep neural network and then introduce the details of SNMF-Net and address its learning and implementation issues.

A. Connection between proximal gradient method and DNN

As shown in Fig. 1, the proposed SNMF-Net is a joint LMM-based network that alternately optimizes endmembers and abundances by End-Net and Abun-Net. Both End-Net and Abun-Net are obtained by unrolling the iterative optimization of \mathbf{A} and \mathbf{S} . The optimization problems of \mathbf{A} and \mathbf{S} both in essence share the following unconstrained problem:

$$\min_{\mathbf{u}} \frac{1}{2} \|\mathbf{Bu} - \mathbf{v}\|_2^2 + f(\mathbf{u}) \quad (17)$$

where the first part is convex and differentiable, representing the reconstruction term and the second part is neither necessarily convex nor differentiable but has a fast proximal operator, representing the regularization term.

Proximal gradient algorithm is a widely-used framework for such problem with the following update rule comprising of two fundamental steps, i.e.,

$$\mathbf{u}' \leftarrow \mathbf{u}_k - t \mathbf{B}^T (\mathbf{B} \mathbf{u}_k - \mathbf{v}) \quad (18)$$

$$\mathbf{u}_{k+1} \leftarrow \mathbf{prox}_{tf}(\mathbf{u}') \quad (19)$$

where \mathbf{prox} is the proximal operator for example soft-thresholding operator and $t > 0$ is the step size. As pointed out by [50], the upper bound of t is $\frac{1}{\|\mathbf{B}^T \mathbf{B}\|_2}$ to satisfy the strict convergence of the proximal gradient method. Running Eq. (18) and Eq. (19) a number of iterations till convergence yields the solution of \mathbf{u} .

From another point of view, the two steps of updating \mathbf{u} can be linked with a typical local block of a recurrent neural network (RNN). Concretely, Eq. (18) accounts for a linear operation to compute an update in the direction of the gradient and Eq. (19) mimics an element-wise nonlinearity with a proximal operator **prox**. Taking soft-thresholding operator as an example, it can be defined as $\mathcal{S}_\lambda(u) = \text{ReLU}(x - \lambda) - \text{ReLU}(-x - \lambda)$. When the number of iterations truncated to K and unrolling the RNN for K times, the network becomes a K -layer feed-forward neural network and supports discriminative learning. Introducing two differentiable variables respectively responsible for step size t and $\mathbf{W} = \mathbf{B}^T$ as network parameters, Eq. (18) and Eq. (19) can be unrolled into a network with consecutive layers being a function parameterized by $\{\mathbf{W}, t\}$ such that

$$\mathbf{u}_{k+1} = \varphi(\mathbf{u}_k - t\mathbf{W}(\mathbf{B}\mathbf{u}_k - \mathbf{v})) \quad (20)$$

Here $\varphi(\cdot)$ is a non-linear activation function and k indexes the layer corresponding to k -th iteration in the iterative update procedure.

B. L_p -NMF Inspired Deep Alternating Neural Network

Following the above perspective, we introduce the detailed network architecture for endmembers and abundances.

1) **End-Net for Endmembers:** Based on the iterative update in Eq. (9), the unrolled subnetwork for endmember can be defined by recursion

$$\mathbf{A}_{k+1} = \varphi(\mathbf{A}_k - t_{1_k}(\mathbf{A}_k\mathbf{S}_k - \mathbf{X})\mathbf{W}_{1_k})) \quad (21)$$

where $\varphi(\cdot) = \max(\cdot)$ is the ReLU activation function allowing for the nonnegativity of \mathbf{A} .

From Eq. (21), it can be found that \mathbf{W}_{1_k} shares the same size with the input samples, indicating that the network can only process a fixed number of pixels at a time in order to satisfy matrix multiplication. In practice, we always expect the network can accept any number of pixels for testing. Moreover, End-Net produces a set of endmembers given a set of pixels and the endmembers may change when provided with a different set of pixels. This is unreasonable for the unmixing task because all the pixels in a scene should share the same set of endmembers. To satisfy the above requirement, we instead replace \mathbf{W}_{1_k} with \mathbf{S}_k^T so that the network is able to accept any number of pixels and all the pixels share the same set of endmembers. Accordingly, the network defined in Eq. (21) follows the iteration

$$\mathbf{A}_{k+1} = \varphi(\mathbf{A}_k - t_{1_k}(\mathbf{A}_k\mathbf{S}_k - \mathbf{X})\mathbf{S}_k^T) \quad (22)$$

Unlike the original network defined in Eq. (21), only the step sizes $\{t_{1_k}\}$ are required to learn in the network defined by Eq. (22), significantly simplifying network training.

2) **Abun-Net for Abundances:** Different from \mathbf{A} , the update rule of \mathbf{S} contains an additional iterative optimization of S_p^{GST} , which makes it very difficult to analyze the relationship between \mathbf{S}_k and (λ, p) . For simplicity, we only iterate Eq. (15) for one time, yielding the following solution for GST

$$\eta_{(\lambda,p)}(y) = \begin{cases} \text{sign}(y)(|y| - \lambda p(|y|)^{p-1}), & \text{if } |y| > \Delta_p^{\text{GST}}(\lambda) \\ 0, & \text{otherwise} \end{cases} \quad (23)$$

Setting different parameters per iteration generally helps yield more favorable performance and accelerate the convergence. This strategy has been commonly employed, especially for non-convex problems [16], [51]–[53]. Moreover, decouple learning across layers also is beneficial to overcome gradient explosion and vanishing problems. To this end, we adopt this strategy and choose different parameters $\{\lambda_k, p_k\}$ across the iteration of \mathbf{S} . Consequently, the updating rule for \mathbf{S}_{k+1} in Eq. (11) becomes:

$$\mathbf{S}_{k+1} \leftarrow [\eta_{(\lambda_k, p_k)}(\mathbf{S}_k - t_{2_k}\mathbf{A}_{k+1}^T(\mathbf{A}_{k+1}\mathbf{S}_k - \mathbf{X}))]_+ \quad (24)$$

Accordingly, the succeeding layers in Abun-Net are given by

$$\mathbf{S}_{k+1} = \zeta(\varphi(\eta_{(\lambda_k, p_k)}(\mathbf{S}_k - t_{2_k}\mathbf{W}_{2_k}(\mathbf{A}_{k+1}\mathbf{S}_k - \mathbf{X})))) \quad (25)$$

where $\zeta(\cdot)$ is a normalization layer satisfying ASC in Eq. (16). The parameter set of Abun-Net is $\{t_{2_k}, \mathbf{W}_{2_k}, p_k, \lambda_k\}$. Instead of inefficiently tuning p to obtain the desired sparsity level, p is also layer-wise learned from data, offering better adaptation to the underlying properties of the data.

In summary, based on the LMM model, a L_p -NMF unmixing model is first built. Inspired by the connection between the PG optimization method and deep neural network, we unroll the optimization of L_p -NMF for \mathbf{A} and \mathbf{S} to yield a K -layer alternating network dubbed SNMF-Net for blind unmixing. Different from alternative deep learning-based unmixing methods, SNMF-Net is physically interpretable. On one hand, SNMF-Net is under the framework of LMM which considers the light interaction between materials and carefully designed optimization. On the other hand, all the key parameters can be seamlessly linked with the associate optimization and support end-to-end learning. Thanks to such interpretation, all the prior knowledge about the unmixing task, optimization algorithms, and sparse representation theory can be easily integrated into the network for robust learning.

C. Network Training

In blind unmixing, both \mathbf{A} and \mathbf{S} are unknown and thereby it's unrealistic to directly learn them by discriminative training. Instead, we take the Euclidean distance between the reconstructed spectrum and input spectrum as the loss function. Motivated by the phenomenon that adding auxiliary classifiers to intermediate layers helps to propagate gradients back through all the layers [54], we take the output of all the layers into the loss function. Given training dataset $\mathbf{X} = \{\mathbf{x}_1, \dots, \mathbf{x}_N\}$ including N spectrum, the loss function can be defined as

$$\mathcal{L}_\Theta = \frac{1}{2N} \sum_{k=1}^K \|\mathbf{X} - \mathbf{A}_k \mathbf{S}_k\|_F^2 \quad (26)$$

where $\Theta = \{t_{1_k}, t_{2_k}, \mathbf{W}_{2_k}, p_k, \lambda_k\}$ are the parameters to be learned and K is the number of layers. To learn the network, the partial derivative of \mathcal{L}_Θ concerning each variable should be computed first. For simplicity, we consider an individual

training sample \mathbf{x} and define the following intermediate variables

$$\begin{aligned}\tilde{\mathbf{A}}_k &= \mathbf{A}_{k-1} - t_{1_{k-1}} (\mathbf{A}_{k-1} \mathbf{s}_{k-1} - \mathbf{x}) \mathbf{s}_{k-1}^T \\ \tilde{\mathbf{s}}_k &= \mathbf{s}_{k-1} - t_{2_{k-1}} \mathbf{W}_{2_{k-1}} (\mathbf{A}_k \mathbf{s}_{k-1} - \mathbf{x}) \\ \hat{\mathbf{s}}_k &= \varphi(\eta_{(\lambda_{k-1}, p_{k-1})}(\tilde{\mathbf{s}}_k))\end{aligned}\quad (27)$$

The partial differentials of \mathcal{L}_{Θ} with respect to each variable are as follows:

$$\begin{aligned}\frac{\partial \mathcal{L}}{\partial t_{2_k}} &= \frac{\partial \tilde{\mathbf{s}}_{k+1}}{\partial t_{2_k}} \frac{\partial \mathcal{L}}{\partial \tilde{\mathbf{s}}_{k+1}} & \frac{\partial \mathcal{L}}{\partial p_k} &= \frac{\partial \tilde{\mathbf{s}}_{k+1}}{\partial p_k} \frac{\partial \mathcal{L}}{\partial \tilde{\mathbf{s}}_{k+1}} \\ \frac{\partial \mathcal{L}}{\partial \lambda_k} &= \frac{\partial \tilde{\mathbf{s}}_{k+1}}{\partial \lambda_k} \frac{\partial \mathcal{L}}{\partial \tilde{\mathbf{s}}_{k+1}} & \frac{\partial \mathcal{L}}{\partial \mathbf{W}_{2_k}} &= \frac{\partial \mathcal{L}}{\partial \tilde{\mathbf{s}}_{k+1}} \frac{\partial \mathbf{W}_{2_k}}{\partial \tilde{\mathbf{s}}_{k+1}} \\ \frac{\partial \mathcal{L}}{\partial \tilde{\mathbf{s}}_{k+1}} &= \frac{\partial \tilde{\mathbf{s}}_{k+1}}{\partial \tilde{\mathbf{s}}_{k+1}} \frac{\partial \mathcal{L}}{\partial \tilde{\mathbf{s}}_{k+1}} & \frac{\partial \mathcal{L}}{\partial \tilde{\mathbf{A}}_{k+1}} &= \frac{\partial \mathcal{L}}{\partial \mathbf{A}_{k+1}} \frac{\partial \mathbf{A}_{k+1}}{\partial \tilde{\mathbf{A}}_{k+1}} \\ \frac{\partial \mathcal{L}}{\partial t_{1_k}} &= \frac{\partial \tilde{\mathbf{A}}_{k+1}}{\partial t_{1_k}} \frac{\partial \mathcal{L}}{\partial \tilde{\mathbf{A}}_{k+1}}\end{aligned}\quad (28)$$

$\frac{\partial \mathbf{A}_{k+1}}{\partial \tilde{\mathbf{A}}_{k+1}}$ and $\frac{\partial \tilde{\mathbf{s}}_{k+1}}{\partial \tilde{\mathbf{s}}_{k+1}}$ are both related to corresponding activation functions and can be respectively obtained by

$$\frac{\partial \mathbf{A}_{k+1}}{\partial \tilde{\mathbf{A}}_{k+1}} = \mathcal{I}_{\tilde{\mathbf{A}}_{k+1} > 0} \quad \frac{\partial \tilde{\mathbf{s}}_{k+1}}{\partial \tilde{\mathbf{s}}_{k+1}} = \mathcal{I}_{\eta_{(\lambda_k, p_k)}(\tilde{\mathbf{s}}_{k+1}) > 0} \quad (29)$$

where \mathcal{I} is an indicator function. According to Eq. (27), we can get

$$\begin{aligned}\frac{\partial \tilde{\mathbf{s}}_{k+1}}{\partial t_{2_k}} &= \mathbf{W}_{2_k} (\mathbf{x} - \mathbf{A}_{k+1} \mathbf{s}_{k+1}) & \frac{\partial \tilde{\mathbf{A}}_{k+1}}{\partial t_{1_k}} &= (\mathbf{x} - \mathbf{A}_k \mathbf{s}_k) \mathbf{s}_k^T \\ \frac{\partial \tilde{\mathbf{s}}_{k+1}}{\partial \mathbf{W}_{2_k}} &= t_{2_k} (\mathbf{x} - \mathbf{A}_{k+1} \mathbf{s}_k)\end{aligned}\quad (30)$$

Based on Eq. (23), each element of $\frac{\partial \tilde{\mathbf{s}}_{k+1}}{\partial p_{k+1}}$ and $\frac{\partial \tilde{\mathbf{s}}_{k+1}}{\partial \lambda_{k+1}}$ can respectively be derived by

$$\frac{\partial \tilde{s}_{k+1}}{\partial p_k} = \begin{cases} 0, & \tilde{s}_{k+1} \leq \Delta_{p_k}^{\text{GST}}(\lambda_k) \\ -\lambda \text{sign}(\tilde{s}_{k+1}) |\tilde{s}_{k+1}|^{p-1} (p \ln(|\tilde{s}_{k+1}|) + 1), & \text{otherwise} \end{cases} \quad (31)$$

$$\frac{\partial \tilde{s}_{k+1}}{\partial \lambda_k} = \begin{cases} 0, & \tilde{s}_{k+1} \leq \Delta_{p_k}^{\text{GST}}(\lambda_k) \\ -\text{sign}(\tilde{s}_{k+1}) p (|\tilde{s}_{k+1}|)^{p-1} & \text{otherwise} \end{cases} \quad (32)$$

Based on Eq. (16), $\frac{\partial \mathbf{s}_{k+1}}{\partial \tilde{\mathbf{s}}_{k+1}}$ is

$$\frac{\partial \mathbf{s}_{k+1}}{\partial \tilde{\mathbf{s}}_{k+1}} = \frac{1}{T} \mathbf{I} - \frac{1}{T^2} \hat{\mathbf{s}}_{k+1} \mathbf{1}_R^T \quad (33)$$

where $T = \sum \hat{\mathbf{s}}_{k+1}$.

$\frac{\partial \mathcal{L}}{\partial \mathbf{A}_k}$ and $\frac{\partial \mathcal{L}}{\partial \mathbf{s}_k}$ can be respectively computed as

$$\begin{aligned}\frac{\partial \mathcal{L}}{\partial \mathbf{s}_K} &= \mathbf{A}_K^T (\mathbf{A}_K \mathbf{s}_K - \mathbf{x}) \\ \frac{\partial \mathcal{L}}{\partial \mathbf{A}_K} &= (\mathbf{A}_K \mathbf{s}_K - \mathbf{x}) \mathbf{s}_K^T - t_{2_{K-1}} \mathbf{W}_{2_{K-1}}^T \frac{\partial \mathcal{L}}{\partial \tilde{\mathbf{s}}_K} \tilde{\mathbf{s}}_K^T \\ \frac{\partial \mathcal{L}}{\partial \mathbf{s}_k} &= \mathbf{A}_k^T (\mathbf{A}_k \mathbf{s}_k - \mathbf{x}) + t_{1_k} (\mathbf{x} - 2 \mathbf{A}_k \mathbf{s}_k) \frac{\partial \mathcal{L}}{\partial \tilde{\mathbf{A}}_{k+1}} \\ &\quad + (\mathbf{I} - t_{2_k} \mathbf{W}_{2_k} \mathbf{A}_{k+1}) \frac{\partial \mathcal{L}}{\partial \tilde{\mathbf{s}}_{k+1}}, \quad k = 1 : K-1 \\ \frac{\partial \mathcal{L}}{\partial \mathbf{A}_k} &= (\mathbf{A}_k \mathbf{s}_k - \mathbf{x}) \mathbf{s}_k^T + \frac{\partial \mathcal{L}}{\partial \tilde{\mathbf{A}}_{k+1}} (\mathbf{I} - t_{1_k} \mathbf{s}_k \mathbf{s}_k^T) \\ &\quad - t_{2_{k-1}} \mathbf{W}_{2_{k-1}}^T \frac{\partial \mathcal{L}}{\partial \tilde{\mathbf{s}}_k} \tilde{\mathbf{s}}_k^T, \quad k = 1 : K-1\end{aligned}\quad (34)$$

Substituting Eq.s (29), (30), (31), (32), (34) into Eq. (28) we can get the derivations of all the variables. Once they are obtained, all the parameters in Θ can be learned by end-to-end training via any gradient descent method. The training process can be summarized as follows:

- 1) Estimate the endmember matrix \mathbf{A}_0 using any endmember extraction algorithm, e.g., VCA.
- 2) Randomly sample a number of pixels to construct the training set $\mathbf{X} = \{\mathbf{x}_1, \dots, \mathbf{x}_N\}$ and calculate their initial abundances \mathbf{S}_0 by any traditional unmixing methods, for example, FCLS.
- 3) Initialize the network with \mathbf{A}_0 and \mathbf{S}_0 . In contrast to alternative deep learning based methods, the embedded models make it easy to integrate the prior knowledge on unmixing into the network for enhanced learning.
- 4) Train SNMF-Net with stochastic gradient descent algorithm till convergence to learn the network parameters.

D. Network Implementation

It is always expected to incorporate the additional auxiliary knowledge at hand to guide the network learning especially when the training samples are small. Thanks to the high interpretability of SNMF-Net, prior knowledge can be reasonably integrated into the network to achieve satisfactory unmixing. NMF based unmixing methods usually initialize the factor matrices with the endmembers extracted by VCA [6] and abundance estimated by FCLS [7]. For this reason, we also adopt this strategy to initialize \mathbf{A}_0 and \mathbf{S}_0 . After that, t_{1_k} and t_{2_k} are accordingly set as $\frac{1}{\|\mathbf{S}_0 \mathbf{S}_0^T\|_2 + \epsilon}$ and $\frac{1}{\|\mathbf{A}_0^T \mathbf{A}_0\|_2 + \epsilon}$ where ϵ is a tiny number given by 0.01. p is initially assigned as 0.5 based on the observation in [12].

We follow the mutual coherence minimization in sparse representation theory to initialize \mathbf{W}_{2_k} . In sparse representation, mutual coherence is an important indicator to depict the sparse recovery quality of a dictionary defined as

$$\mu(\mathbf{D}) = \max_{1 \leq i \neq j \leq L} \frac{\mathbf{D}_i^T \mathbf{D}_j}{\|\mathbf{D}_i\|_2 \|\mathbf{D}_j\|_2} \quad (35)$$

A dictionary with low coherence implies better sparse recovery ability. Based on above definition, we define a generalized mutual coherence as follows:

Definition 1. For a dictionary $\mathbf{D} \in \mathbb{R}^{L \times R}$, its generalized mutual coherence (GMC) is given by

$$\hat{\mu}(\mathbf{D}) = \inf_{\substack{\mathbf{W} \in \mathbb{R}^{L \times R} \\ \mathbf{W}^T \mathbf{D}_i = 1, 1 \leq i \leq R}} \left\{ \max_{1 \leq i \neq j \leq R} \mathbf{W}_i^T \mathbf{D}_j \right\} \quad (36)$$

As pointed out in [42], if $\mathbf{W} \in \mathcal{W}(\mathbf{D})$ where $\mathcal{W}(\mathbf{D}) = \{\mathbf{W} \in \mathbb{R}^{L \times R} : \text{attains the infimum in Eq. (36)}\}$, the sparsity recovery error is bounded and \mathbf{W} can be precomputed by solving the following problem:

$$\arg \min_{\mathbf{W} \in \mathbb{R}^{L \times R}} \|\mathbf{W}^T \mathbf{D}\|_F^2 \quad \text{s.t. } \mathbf{W}^T \mathbf{D} = \mathbf{I} \quad (37)$$

In this way, there is no need to learn \mathbf{W} in a data-drive manner, significantly reducing the training complexity and improving the robustness of a network.

When it comes to unmixing, the endmembers can be considered as dictionaries. Intuitively, we can compute \mathbf{W}_{2k} using Eq. (37) when the endmembers are ready. However, the endmembers are also unknown in the blind unmixing task, meaning that it is unrealistic to obtain an exact solution of \mathbf{W}_{2k} in advance. Fortunately, as mentioned above, VCA can provide a good initial estimation of endmembers. For this regard, we alternatively still keep \mathbf{W}_{2k} learnable and initialize them as the values computed by Eq. (37). By this means, the prior knowledge on sparse representation theory is integrated into the network which enhances the network learning as will be shown in the experiment section.

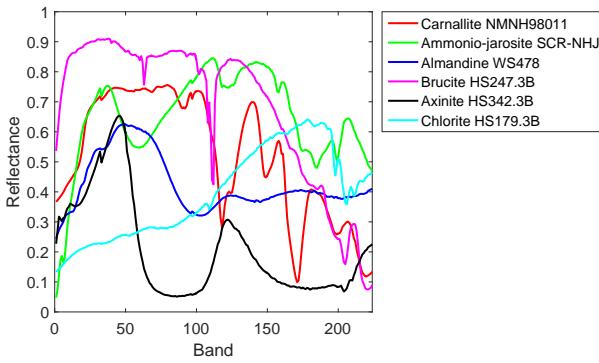


Fig. 2. The spectral signatures of six endmembers used in the synthetic data.

IV. EXPERIMENTS

In this section, we comprehensively evaluate the unmixing ability of the proposed SNMF-Net on both synthetic data and real-world data. Seven methods are selected as alternative methods to be compared with our method, including Arctan-NMF [17], TV-RSNMF [55], MV-NTF [18], uDAS [26], UnDIP [30], DAEN [22] and MNN-BU [39]. Among these compared methods, Arctan-NMF, TV-RSNMF and MV-NTF are model-based methods. Arctan-NMF and TV-RSNMF are based on NMF. MV-NTF is a tensor based method. uDAS and DAEN are learning-based methods based on autoencoder. MNN-BU is a model-based method whose network architecture is obtained by unrolling the optimization of abundances.

UnDIP takes advantage of deep prior knowledge embodied by deep neural networks to improve unmixing. All relevant parameters are set as suggested in the original implementation.

A. Performance Evaluation Criteria

The spectral angle distance (SAD) and root-mean-square error (RMSE) were used for quantitative performance evaluation. SAD measures the spectral angle between reference endmembers \mathbf{A}_r and estimated endmembers $\widehat{\mathbf{A}}_r$ defined as

$$\text{SAD}_r = \arccos \left(\frac{\mathbf{A}_r^T \widehat{\mathbf{A}}_r}{\|\mathbf{A}_r\| \|\widehat{\mathbf{A}}_r\|} \right) \quad (38)$$

RMSE computes the error between reference abundance map \mathbf{S}_r and estimated map $\widehat{\mathbf{S}}_r$, mathematically formulated as

$$\text{RMSE}_r = \left(\frac{1}{N} |\mathbf{S}_r - \widehat{\mathbf{S}}_r|^2 \right)^{\frac{1}{2}} \quad (39)$$

The small values of SAD and RMSE imply better unmixing.

B. Experiments on Synthetic Data

Six pure signatures (Carnallite, Ammonio-jarosite, Almandine, Brucite, Axinite, and Chlonte) covering 224 bands with wavelengths between $0.38\mu\text{m}$ and $2.5\mu\text{m}$ were first selected from the USGS library¹ to generate endmembers. Their spectral curves are displayed in Fig. 2. Following [19], the abundances were generated as follows:

- 1) A synthetic image containing $z^2 \times z^2$ pixels is segmented into z^2 nonoverlapped regions with the same size of $z \times z$.
- 2) Each pixel of a region is filled with two randomly selected endmembers whose ratios are respectively given as β and $1 - \beta$.
- 3) A spatial low-pass Gaussian filter whose variance is set to 2 is used to generate mixed pixels, resulting in a relatively higher degree of mixing.
- 4) Finally, the fractions of all endmembers in each pixel are rescaled to meet the ASC constraint.

In our experiment, we set $z = 8$ and $\beta = 0.8$ to generate the synthetic data. The generated clean HSI is further degraded by zero-mean additive Gaussian to produce noisy HSI with signal-to-noise ratios (SNRs) defined as

$$\text{SNR} = 10 \log_{10} \frac{E[\mathbf{y}^T \mathbf{y}]}{E[\mathbf{e}^T \mathbf{e}]} \quad (40)$$

where \mathbf{y} and \mathbf{e} are the clean signal and the noise at a pixel and $E[\cdot]$ denotes the expectation operator.

1) *Influence of the number of layers*: We first analyze the influence of layers on unmixing accuracy. The clean HSI was degraded by Gaussian noise with SNR=30dB and the training size was set to 500. By altering K from 3 to 30 with an interval of 3, Fig. 3 presents the changes of SAD and RMSE. It can be seen that both SAD and RMSE decrease rapidly when K increases from 3 to 9, implying the positive effect of deep models. After reaching the bottom at $K = 9$, SAD

¹<https://www.usgs.gov/labs/spec-lab/capabilities/spectral-library>

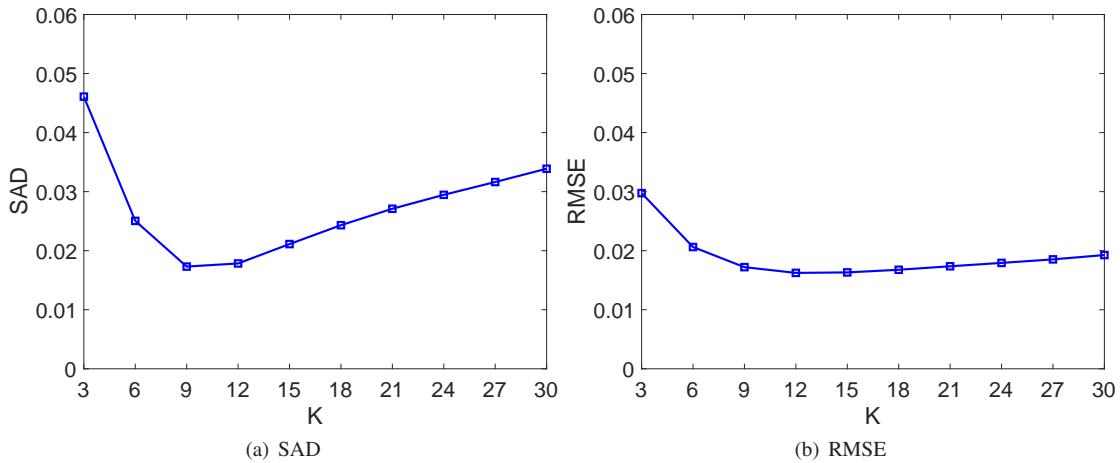


Fig. 3. SAD and RMSE with respect to the number of layers.

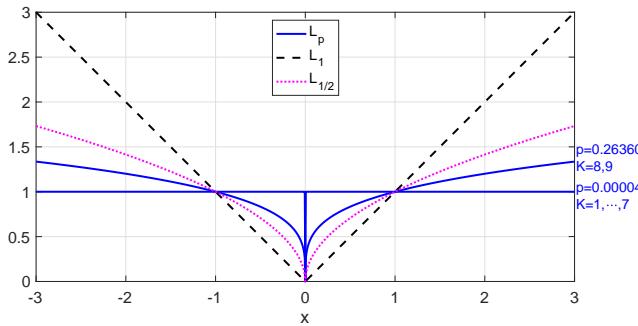


Fig. 4. Learned L_p sparse penalty with respect to K values.

gradually increases and RMSE almost stays steady. This may because the network parameters increase as the number of layers increases, which requires more data to learn. Since the best values of SAD and RMSE are obtained when $K = 9$, we set the number of layers to 9 in the following experiments. This experiment shows very few layers of SNMF-Net is able to produce promising unmixing, confirming its strong representing ability.

When $K = 9$, we demonstrate the learned p concerning the number of layers. As shown in Fig. 4, in the first few layers, L_p norm is close to L_0 norm, indicating that the network tends to obtain a very sparse estimation of abundance maps which helps to suppress the side-effect of perturbation. When K is over 7, the p value increases to 0.2636, meaning that the network is learning to produce less sparser solution. This is also in line with that as the number of iterations increases, the optimization algorithm gradually converges and smaller penalties should be added. Moreover, compared with the fixed L_1 norm, our network is more adaptive to the data as p is learned from the data. Unlike L_1 norm, the proposed is able to yield sparser and unbiased abundances because the gradient of learned L_p norm decreases as x increases, which means large magnitudes components are less possible to be shrunk to 0. In short, this experiment verifies the interpretability of the proposed SNMF-Net.

2) *Influence of the number of training samples:* In this experiment, we investigate the impact of the number of training samples on unmixing. We set SNR=30dB to gen-

erate the noisy HSI. Fig. 5 plots the SAD and RMSE changes with the number of training samples in the range of $\{64, 256, 500, 1000, 2000, 4096\}$. As can be seen from Fig. 5, without using all the pixels for training, i.e., only 500 pixels (12.21% of all the pixels), the network can achieve as satisfying performance as unmixing with all pixels. This phenomenon evidently shows the strong learning and generalization ability of SNMF-Net. The main reason is that SNMF-Net takes the spectral mixture model and the alternating optimization of L_p -NMF into consideration to build the network architecture. All the key components of the proposed SNMF-Net are explainable, facilitating the integration of prior knowledge of unmixing into the network for enhanced learning.

3) *Influence of Initialization:* As mentioned in the network implementation, we adopt Eq. (37) to initialize \mathbf{W}_{2_k} which is induced by GMC theory to enhance learning. Here we conducted a study on the effect of such initialization. Fixing the number of training samples as 500 and changing SNR from 15dB, 20dB, 25dB, 30dB to ∞ (noise-free), Fig. 6 reports the SAD and RMSE changes with or without GMC based initialization. Here “SNMF-Net without GMC” refers to setting $\mathbf{W}_{2_k} = \mathbf{A}^T$. It can be clearly observed that GMC-based initialization helps SNMF-Net achieve significantly better and robust unmixing performance. This experiment shows the effectiveness of GMC based initialization and the benefit of interpretable network architecture.

Attributing to the embedded physical model, the prior knowledge learned or based on domain knowledge concerning unmixing can be flexibly incorporated into the network to enhance network learning. We further initialized the network by NMF whose results is denoted as SNMF-Net-NMF. Fig. 7 compares SAD and RMSE with respect to different initializations. As can be seen, using NMF to initialize the network yields lower SAD and RMSE because NMF can provide more accurate endmembers and abundances, specially in lower SNR cases. This experiment evidently show the benefits of model-based neural network, i.e., easy incorporation of prior knowledge.

4) *Robustness to noise:* We further compared SNMF-Net with alternative methods to demonstrate its superiority in unmixing. The SNRs were selected as 15dB, 20dB, 25dB,

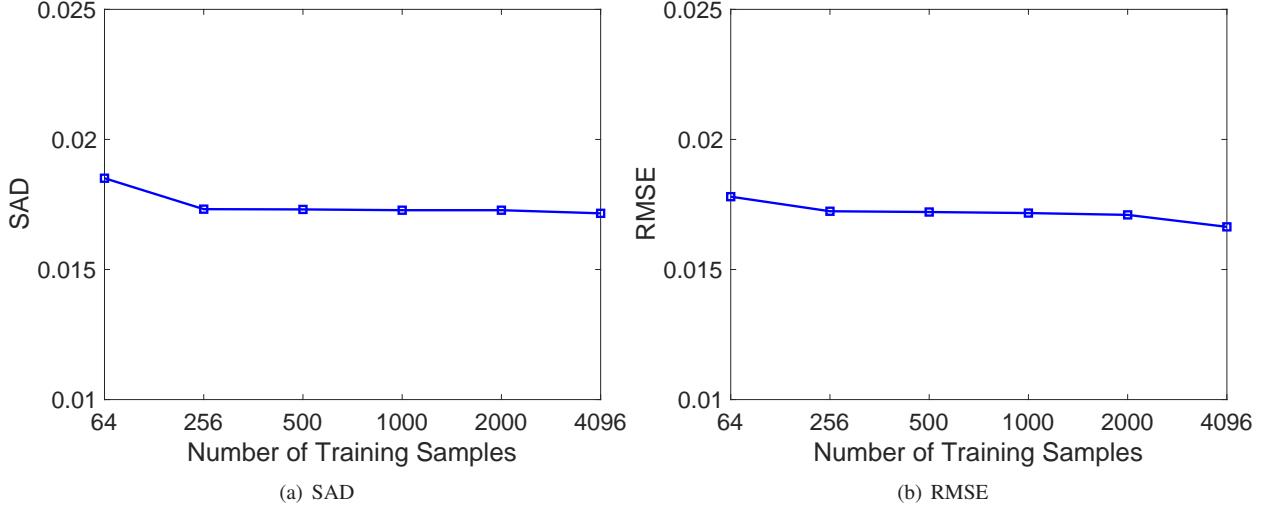


Fig. 5. SAD and RMSE with respect to the number of training samples.

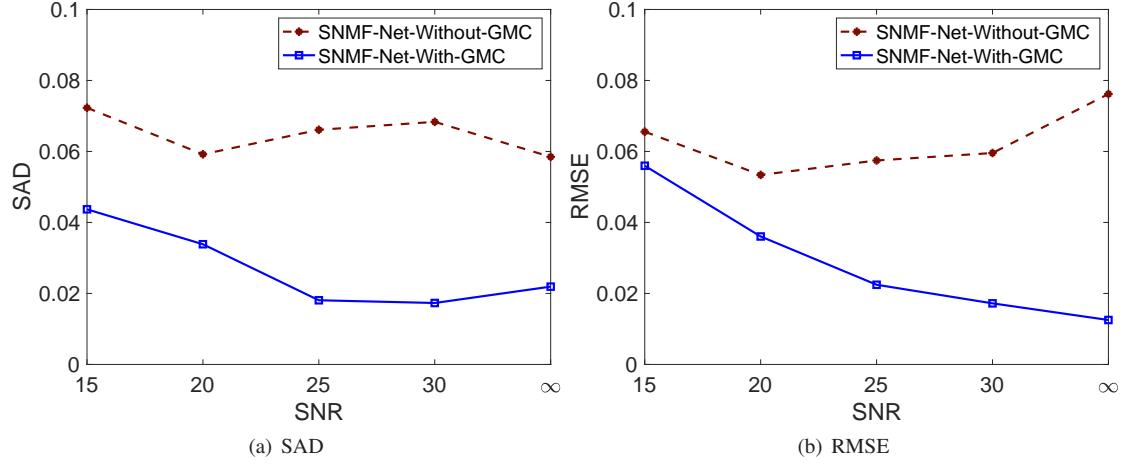


Fig. 6. The impact of GMC based initializations.

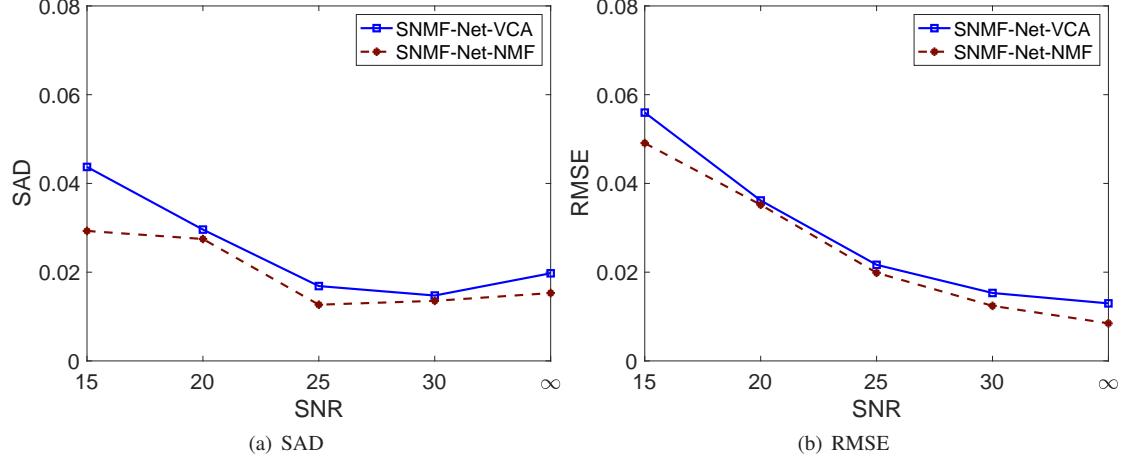


Fig. 7. The impact of different initializations.

30dB and ∞ (noise-free). Based on the above experiment observations, we used 500 pixels to train SNMF-Net. The other compared methods utilized all the pixels to conduct unmixing. Fig. 8 presents the unmixing performance of all the methods regarding SNR settings. In general, most of the unmixing methods show descending trend as the SNR increases. UnDIP

provides inferior SAD and RMSE because it estimates end-members and abundances in sequence rather than simultaneously. The failure of endmember extraction unavoidably causes poor abundance estimation. Moreover, it can be seen from the figure that MV-NTF outperforms TV-RSNMF, especially in high SNR scenarios thanks to its inherent advantages of

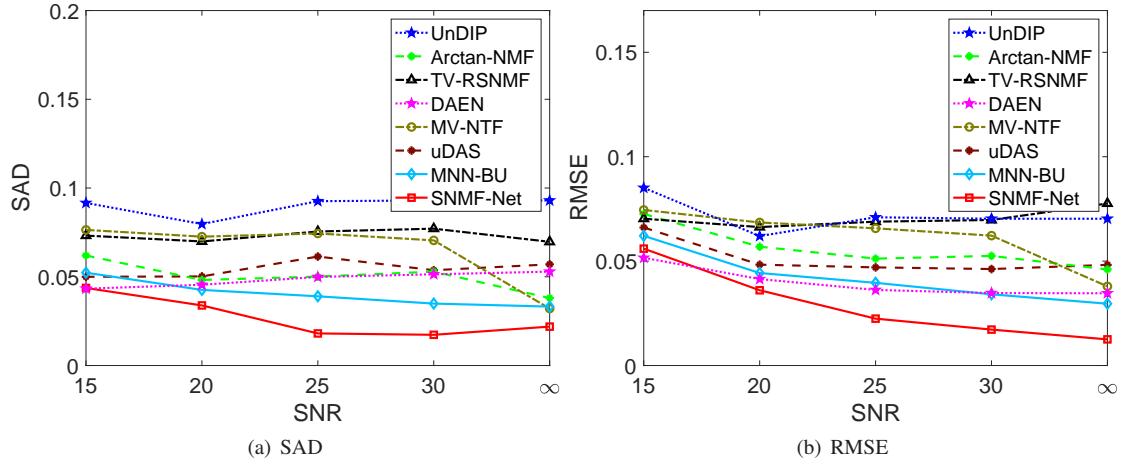


Fig. 8. Performance of all eight unmixing methods with respect to different noise levels in terms of (a) SAD and (b) RMSE.



Fig. 9. Three real-world data sets shown in gray. (a) 3rd band image of Jasper Ridge. (b) 80th band image of Urban. (c) 3rd band image of Washington DC Mall.

TABLE I
MEANS AND STANDARD DEVIATIONS OF THE SAD ON JASPER RIDGE DATASET.

| Algorithm | TV-RSNMF [55] | MV-NTF [18] | DAEN [22] | uDAS [26] | MNN-BU [39] | UnDIP [30] | SNMF-Net |
|-----------|----------------------|---------------------|---------------------------------------|----------------------|---------------------------------------|------------------|---------------------------------------|
| Tree | $0.0776 \pm 4.91\%$ | $0.2209 \pm 2.17\%$ | $0.1774 \pm 0.82\%$ | $0.1511 \pm 1.37\%$ | $0.0468 \pm 0.32\%$ | $0.1493 \pm 0\%$ | $0.0478 \pm 1.62\%$ |
| Water | $0.1337 \pm 1.31\%$ | $0.2352 \pm 1.65\%$ | $0.3237 \pm 1.42\%$ | $0.1742 \pm 11.39\%$ | $0.1179 \pm 0.23\%$ | $0.2529 \pm 0\%$ | $0.0855 \pm 1.77\%$ |
| Soil | $0.1047 \pm 10.11\%$ | $0.1752 \pm 4.67\%$ | $0.1123 \pm 2.03\%$ | $0.1639 \pm 6.09\%$ | $0.0357 \pm 0.43\%$ | $0.1144 \pm 0\%$ | $0.0443 \pm 3.09\%$ |
| Road | $0.6515 \pm 27.46\%$ | $0.1741 \pm 4.35\%$ | $0.0588 \pm 1.59\%$ | $0.0597 \pm 0.27\%$ | $0.0901 \pm 0.37\%$ | $0.2757 \pm 0\%$ | $0.0875 \pm 0.05\%$ |
| Mean | $0.2419 \pm 3.06\%$ | $0.2015 \pm 1.77\%$ | $0.1680 \pm 0.43\%$ | $0.1373 \pm 3.27\%$ | $0.0726 \pm 0.11\%$ | $0.1981 \pm 0\%$ | $0.0663 \pm 1.04\%$ |

preserving the spatial-spectral structure of HSIs. Compared with three model-based methods, DAEN, uDAS, and SNMF-Net provide more desirable unmixing. Among them, the most promising unmixing results belong to the proposed SNMF-Net even using only 12.21% of pixels. This attributes to that SNMF-Net is equipped with the hybrid advantages of model-based methods and learning-based methods.

C. Experiments on Real-world Data

We further carried out experiments on three widely used real-world HSIs, Jasper Ridge, Urban, and Washington DC Mall to more comprehensively show the advantages of SNMF-Net. The same as the synthetic experiment, 500 randomly sampled pixels were adopted for training SNMF-Net. Because of the unavailability of ground-truth endmembers and abundances, we instead selected several pixels that are most likely

to be pure pixels and set their spectral signatures as reference endmembers. After that, the reference abundances maps were obtained using CVX MATLAB software.

1) *Jasper Ridge Dataset*: Jasper Ridge dataset was collected by AVIRIS sensor and contains 512×613 pixels and 224 spectral channels, covering wavelength from 380 to 2500 nm. We considered a part of it with 100×100 pixels doing experiments. After removing bands corrupted by noise and low SNR and water-vapor absorption, 198 bands were retained for experiment. Fig. 9(a) presents the third band image of Jasper Ridge HSI. In the experiment, we set four kinds of materials as endmembers, including tree, water, soil, and road.

Table I provides a quantitative comparison of all the competing methods with respect to the SAD index. From Table I, it can be seen that MV-NTF performs better than TV-RSNMF thanks to the consideration of the spectral-spatial structure of hyperspectral images powered by tensor algebra. The standard

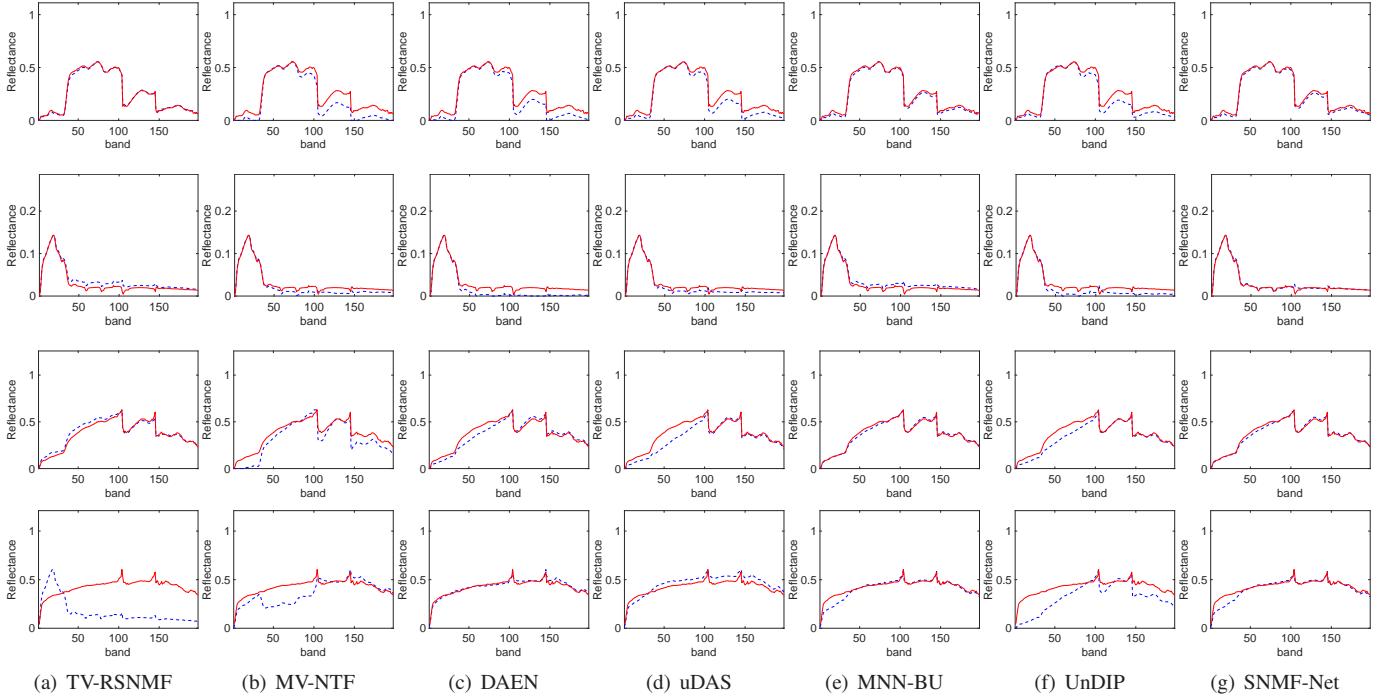


Fig. 10. Estimated endmembers by seven unmixing methods on the Jasper Ridge dataset. From top to bottom, the rows are the spectral signatures of tree, water, soil, and road, respectively. Solid lines denote the reference endmembers and dashed lines denote the estimated endmembers.

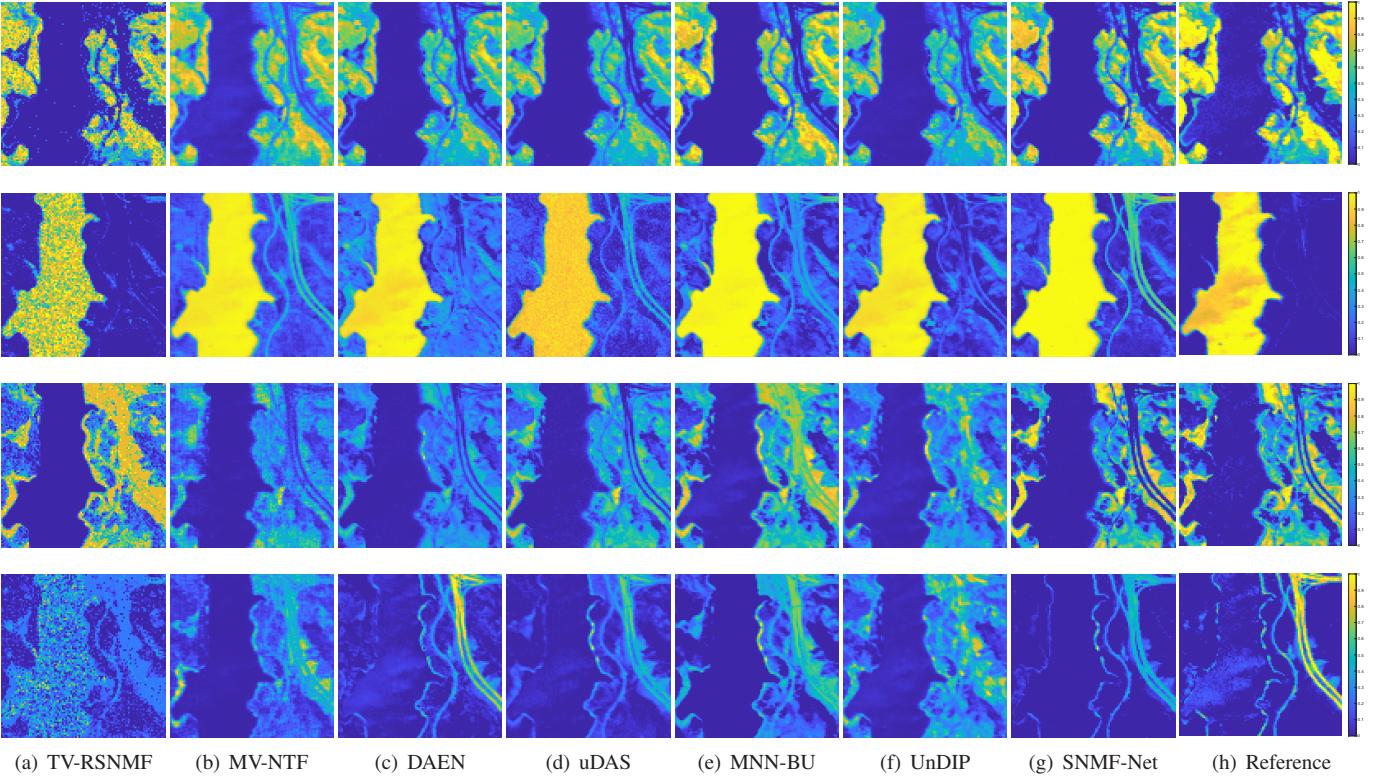


Fig. 11. Estimated abundance maps by seven unmixing methods on the Jasper Ridge dataset. From top to bottom, the rows are the abundance maps of tree, water, soil, and road, respectively.

deviations of UnDIP are zero because UnDIP adopts the simplex volume maximization (SiVM) [56] method to extract endmembers which is insensitive to initializations. Compared with DAEN, learning-based methods with sparsity constraints, i.e., uDAS, MNN-BU, and SNMF-Net perform better. MNN-

BU and SNMF-Net surpass uDAS because they simultaneously take the advantages of model-based and learning-based methods for unmixing. Though provided with better performance than two model-based methods, UnDIP provides unsatisfactory SAD than alternative learning-based methods.

TABLE II
MEANS AND STANDARD DEVIATIONS OF THE SAD ON URBAN DATA.

| Algorithm | TV-RSNMF [55] | MV-NTF [18] | DAEN [22] | uDAS [26] | MNN-BU [39] | SNMF-Net |
|-----------|---------------------|--------------|---------------------|---------------|---------------------|---------------------|
| Asphalt | 0.1028±1.17% | 0.1720±0.75% | 1.0224±0.27% | 0.2174±0.04% | 0.1227±1.24% | 0.2477±0.93% |
| Grass | 0.1463±1.40% | 0.1974±2.97% | 0.5122±0.29% | 0.6862±33.88% | 0.1234±0.88% | 0.2131±2.81% |
| Tree | 0.2628±1.72% | 0.1284±0.32% | 0.0801±0.01% | 0.0955±1.12% | 0.1113±0.20% | 0.1848±2.05% |
| Roof | 0.3332±8.00% | 0.3608±3.66% | 0.1069±0.03% | 0.2037±2.37% | 0.3558±3.80% | 0.0554±2.56% |
| Mean | 0.2113±1.04% | 0.2147±0.96% | 0.4304±0.02% | 0.3007±8.79% | 0.1783±1.22% | 0.1753±0.55% |

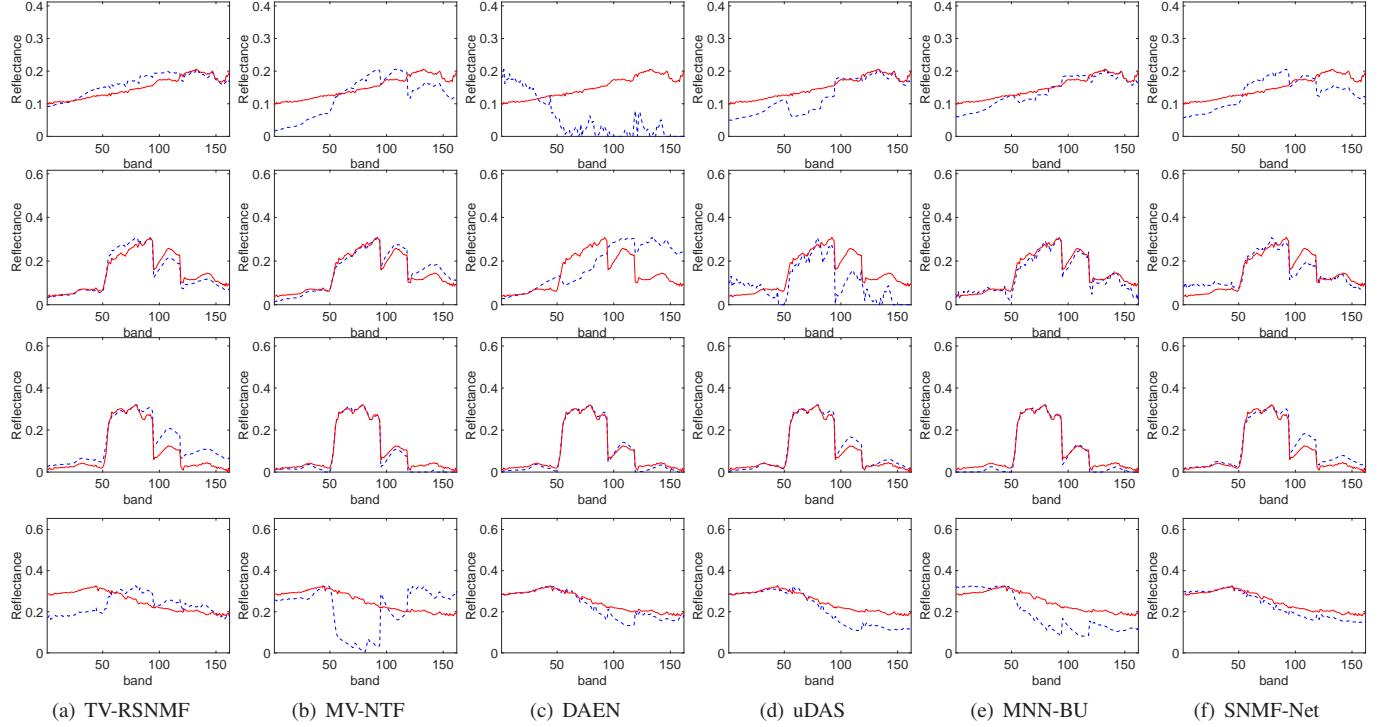


Fig. 12. Estimated endmembers by six unmixing algorithms on the HYDICE Urban dataset. From top to bottom, the rows are the spectral signatures of asphalt, grass, tree, roof, respectively.

The main reason is that UndIP precomputes the endmembers by SiVM and keeps endmembers fixed for abundance estimation. Proposed SNMF-Net stands out among all the methods. On one hand, SNMF-Net adopts alternating network architecture to estimate endmembers and abundances, which fully accords with the alternating optimization of L_p -NMF. On the other hand, SNMF-Net absorbs the spectral mixture model for network constructing and incorporates the prior knowledge on the unmixing problem and sparse representation theory for enhanced learning.

Fig. 10 and Fig. 11 respectively demonstrates the endmembers and abundance map generated by all the methods. The higher brightness (black to white) indicates larger abundance values (0 to 1). Learning-based approaches produce more attractive abundances because of higher learning capacity. SNMF-Net matches best with reference targets and ranks first among all methods, especially the soil and road target whereas other methods such as TV-RSNMF and MV-NTF have some difficulties in accurate estimation. The experiment further proves the superior unmixing ability of SNMF-Net.

2) *HYDICE Urban Dataset*: This HSI was acquired by the HYDICE sensor and covers 307×307 pixels and 210 bands from 0.4 to 2.5 μm . Urban HSI was corrupted by

various noises, which threaten challenges for unmixing. We removed some relative low-SNR bands and water-vapor bands, leaving 162 bands to conduct experiment whose 80-th band is illustrated in Fig. 9(b). We set four material targets, including roof, grass, asphalt road, and tree. The Urban dataset contains outliers and noises, which makes it difficult to extract endmembers accurately. For this reason, UndIP can only extract one endmember, leaving and we didn't include its unmixing result in this experiment.

Table II quantitatively compares the endmembers generated by all the methods in terms of the SAD index. It can be observed MNN-BU and SNMF-Net provides comparatively better unmixing performance because of the hybrid advantages of learning-based and model-based methods. SNMF-Net is slightly superior to MNN-BU using half as many training samples as MNN-BU, showing its better learning ability. Fig. 12 and Fig. 13 respectively gives a visual comparison between all the competing methods. In general, all the methods can not adequately estimate roof and asphalt because of the their relatively less number of training samples and the negative effect of outliers and noises. Proposed SNMF-Net dominates other compared methods by sharing more similarities with the reference ones.

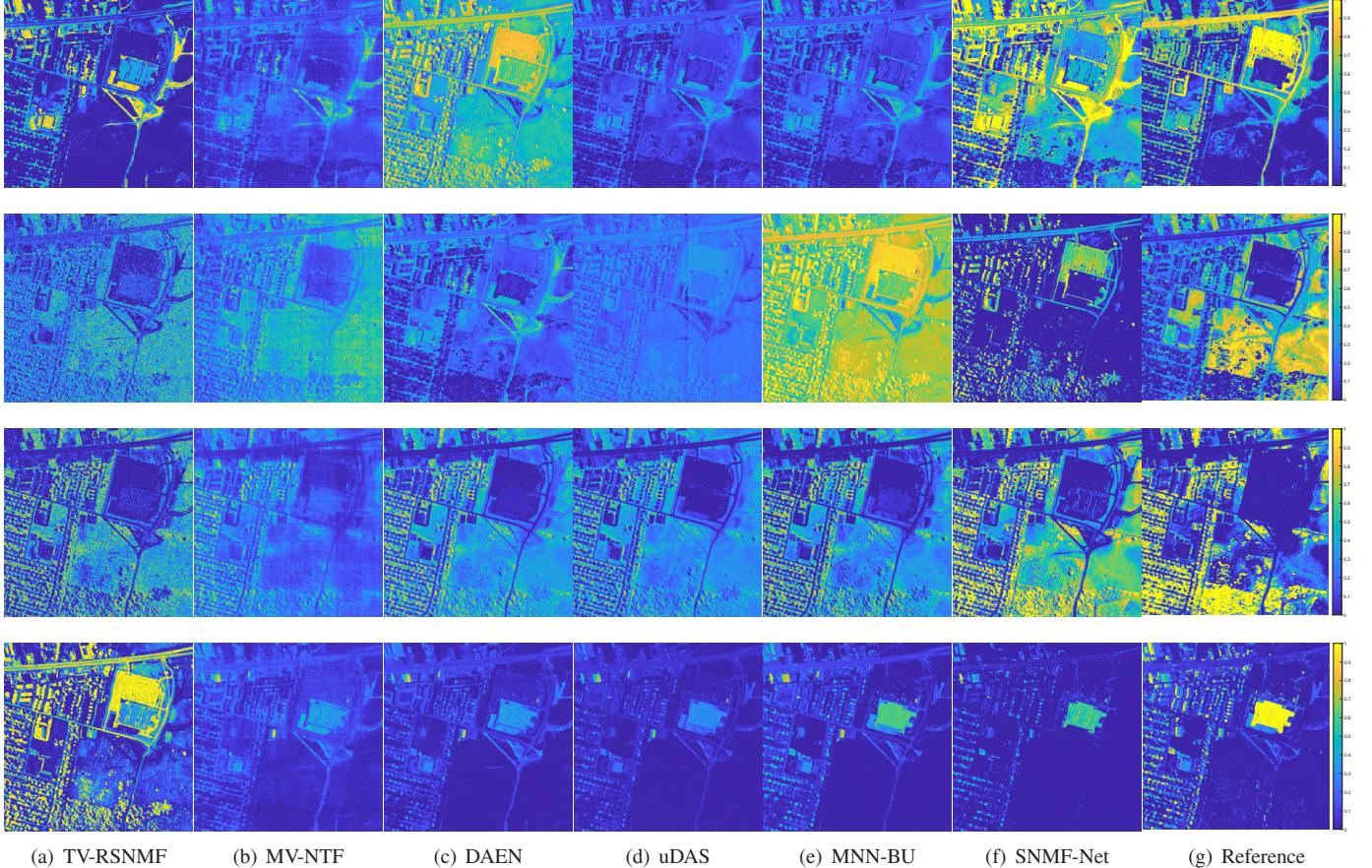


Fig. 13. Estimated abundance maps by six unmixing algorithms on the HYDICE Urban dataset. From top to bottom, the rows are the abundance maps of asphalt, grass, tree, roof, respectively.

TABLE III
MEANS AND STANDARD DEVIATIONS OF THE SAD ON WASHINGTON DC MALL DATA

| Algorithm | TV-RSNMF [55] | MV-NTF [18] | DAEN [22] | uDAS [26] | MNN-BU [39] | UnDIP [30] | SNMF-Net |
|-----------|---------------------|--------------|---------------|---------------|---------------------|------------------|---------------------|
| Tree | 0.20711±1% | 0.1211±0.53% | 0.1378±2.33% | 0.0939±2.15% | 0.1906±0.23% | 0.0795±0% | 0.1309±1.06% |
| Grass | 0.2961±4.34% | 0.2357±0.40% | 0.3773±17.84% | 0.2888±6.03% | 0.3513±6.84% | 0.2483±0% | 0.2180±2.09% |
| Street | 0.3338±1.57% | 0.4183±0.75% | 0.3870±17.49% | 0.3270±10.34% | 0.2376±6.46% | 0.4216±0% | 0.3836±0.79% |
| Roof | 0.2163±2.84% | 0.0818±3.46% | 0.1855±10.50% | 0.0801±3.06% | 0.0915±0.28% | 0.3539±0% | 0.0356±0.51% |
| Water | 0.0466±0.53% | 0.1317±0.70% | 0.1760±5.88% | 0.1264±9.28% | 0.1544±2.28% | 0.0784±0% | 0.0602±1.70% |
| Mean | 0.2200±0.51% | 0.1977±0.80% | 0.2527±3.42% | 0.1833±1.99% | 0.2051±0.53% | 0.2363±0% | 0.1657±0.63% |

3) *Washington DC Mall Dataset*: The original Washington DC Mall dataset includes 1208×307 pixels and 210 bands of spectral resolution 9.52nm. We used a subimage containing 150×150 pixels and 191 bands for experiments by removing the noisy spectral bands (103-106, 138-148, 207-210). Fig. 9(c) shows the third-band image of DC Mall dataset. It was assumed that the image provides tree, grass, street, roof, and water materials.

Table III presents a comparable SAD of all the methods. Using only 500 pixels, SNMF-Net outperforms all other competed methods, showing the advantages of absorbing the knowledge of model-based methods into learning-based methods for unmixing. Fig. 14 and Fig. 15 describe the endmembers and abundances generated by all the methods. The deep prior learned by the neural networks helps UnDIP achieve more accurate endmembers and abundance maps on the tree and roof target. As the phenomenon revealed in Table III,

SNMF-Net performs better in tree, water, roof target but is inferior to alternative methods in street and grass targets. This may because the spectral signature of the latter two targets are respectively very similar to water and tree, making SNMF-Net difficult to distinguish especially using only a subset of all the pixels to train the network parameters.

V. CONCLUSION

In this paper, we propose an interpretable network SNMF-Net for unmixing by unrolling the optimization of L_p sparsity constrained NMF unmixing method. SNMF-Net equips with the hybrid merits of model-based and learning-based methods. The network architecture of SNMF-Net is consistent with the linear spectral mixture model and the alternating optimization of L_p -NMF. All the parameters are explainable and support end-to-end training. Experimental results on both synthetic and real-world HSI show the advantages of SNMF-Net over many

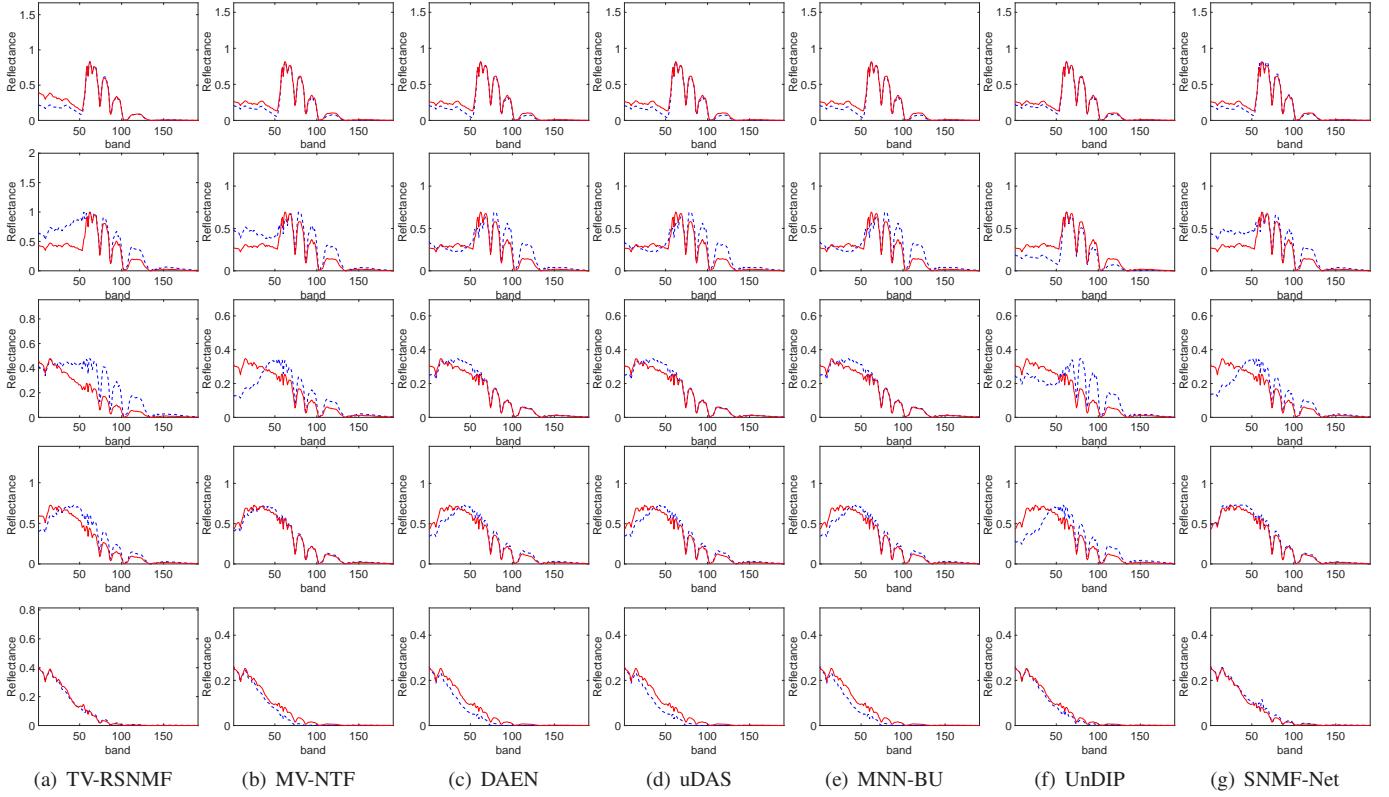


Fig. 14. Estimated endmembers by seven unmixing algorithms on the Washington DC Mall Data. From top to bottom, the rows are the spectral signatures of tree, grass, street, roof, water, respectively.

state-of-the-art methods, i.e., higher interpretability, stronger learning ability, less requirement of training samples, and easy integration of domain knowledge. We will incorporate the spatial information into the network for improved unmixing in our further research.

REFERENCES

- [1] S. Valero, P. Salembier, and J. Chanussot, “Object recognition in hyperspectral images using binary partition tree representation,” *Pattern Recognit. Lett.*, vol. 56, pp. 45 – 51, 2015.
- [2] F. Xiong, J. Zhou, and Y. Qian, “Material based object tracking in hyperspectral videos,” *IEEE Trans. Image Process.*, vol. 29, pp. 3719–3733, 2020.
- [3] G. Lu and B. Fei, “Medical hyperspectral imaging: a review,” *Journal of Biomedical Optics*, vol. 19, no. 1, pp. 1 – 24, 2014.
- [4] P. J. C. Weijtmans, C. Shan, T. Tan, S. G. Brouwer de Koning, and T. J. M. Ruers, “A dual stream network for tumor detection in hyperspectral images,” in *Proc. IEEE International Symposium on Biomedical Imaging (ISBI)*, 2019, pp. 1256–1259.
- [5] J. M. Bioucas-Dias, A. Plaza, N. Dobigeon, M. Parente, Q. Du, P. Gader, and J. Chanussot, “Hyperspectral unmixing overview: Geometrical, statistical, and sparse regression-based approaches,” *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 5, no. 2, pp. 354–379, 2012.
- [6] J. M. P. Nascimento and J. M. B. Dias, “Vertex component analysis: a fast algorithm to unmix hyperspectral data,” *IEEE Trans. Geosci. Remote Sens.*, vol. 43, no. 4, pp. 898–910, 2005.
- [7] D. C. Heinz and Chein-I-Chang, “Fully constrained least squares linear spectral mixture analysis method for material quantification in hyperspectral imagery,” *IEEE Trans. Geosci. Remote Sens.*, vol. 39, no. 3, pp. 529–545, 2001.
- [8] D. D. Lee and H. S. Seung, “Learning the parts of objects by non-negative matrix factorization,” *Nature*, vol. 401, no. 6755, pp. 788–791, 1999.
- [9] L. Miao and H. Qi, “Endmember extraction from highly mixed data using minimum volume constrained nonnegative matrix factorization,” *IEEE Trans. Geosci. Remote Sens.*, vol. 45, no. 3, pp. 765–777, 2007.
- [10] J. Peng, Y. Zhou, W. Sun, Q. Du, and L. Xia, “Self-paced nonnegative matrix factorization for hyperspectral unmixing,” *IEEE Trans. Geosci. Remote Sens.*, vol. 59, no. 2, pp. 1501–1515, 2021.
- [11] X. Wang, Y. Zhong, L. Zhang, and Y. Xu, “Spatial group sparsity regularized nonnegative matrix factorization for hyperspectral unmixing,” *IEEE Trans. Geosci. Remote Sens.*, vol. 55, no. 11, pp. 6287–6304, 2017.
- [12] 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.*, vol. 49, no. 11, pp. 4282–4297, 2011.
- [13] X. Lu, H. Wu, Y. Yuan, P. Yan, and X. Li, “Manifold regularized sparse NMF for hyperspectral unmixing,” *IEEE Trans. Geosci. Remote Sens.*, vol. 51, no. 5, pp. 2815–2826, 2013.
- [14] J. Yao, D. Meng, Q. Zhao, W. Cao, and Z. Xu, “Nonconvex-sparsity and nonlocal-smoothness-based blind hyperspectral unmixing,” *IEEE Trans. Image Process.*, vol. 28, no. 6, pp. 2991–3006, 2019.
- [15] L. Zhou, X. Zhang, J. Wang, X. Bai, L. Tong, L. Zhang, J. Zhou, and E. Hancock, “Subspace structure regularized nonnegative matrix factorization for hyperspectral unmixing,” *IEEE J. Sel. Topics Appl. Earth Observations Remote Sens.*, vol. 13, pp. 4257–4270, 2020.
- [16] W. Wang and Y. Qian, “Adaptive $L_{1/2}$ sparsity-constrained nmf with half-thresholding algorithm for hyperspectral unmixing,” *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 8, no. 6, pp. 2618–2631, 2015.
- [17] Y. E. Salehani and S. Gazor, “Smooth and sparse regularization for nmf hyperspectral unmixing,” *IEEE J. Sel. Topics Appl. Earth Observations Remote Sens.*, vol. 10, no. 8, pp. 3677–3692, 2017.
- [18] Y. Qian, F. Xiong, S. Zeng, J. Zhou, and Y. Y. Tang, “Matrix-vector nonnegative tensor factorization for blind unmixing of hyperspectral imagery,” *IEEE Trans. Geosci. Remote Sens.*, vol. 55, no. 3, pp. 1776–1792, 2017.
- [19] F. Xiong, Y. Qian, J. Zhou, and Y. Y. Tang, “Hyperspectral unmixing via total variation regularized nonnegative tensor factorization,” *IEEE Trans. Geosci. Remote Sens.*, vol. 57, no. 4, pp. 2341–2357, 2019.
- [20] X. Zhang, Y. Sun, J. Zhang, P. Wu, and L. Jiao, “Hyperspectral unmixing via deep convolutional neural networks,” *IEEE Geosci. Remote Sens. Lett.*, vol. 15, no. 11, pp. 1755–1759, 2018.
- [21] L. Zhou, B. Xiao, X. Liu, J. Zhou, E. R. Hancock *et al.*, “Latent distribution preserving deep subspace clustering,” in *Proc. International Joint Conference on Artificial Intelligence (IJCAI)*, 2019.

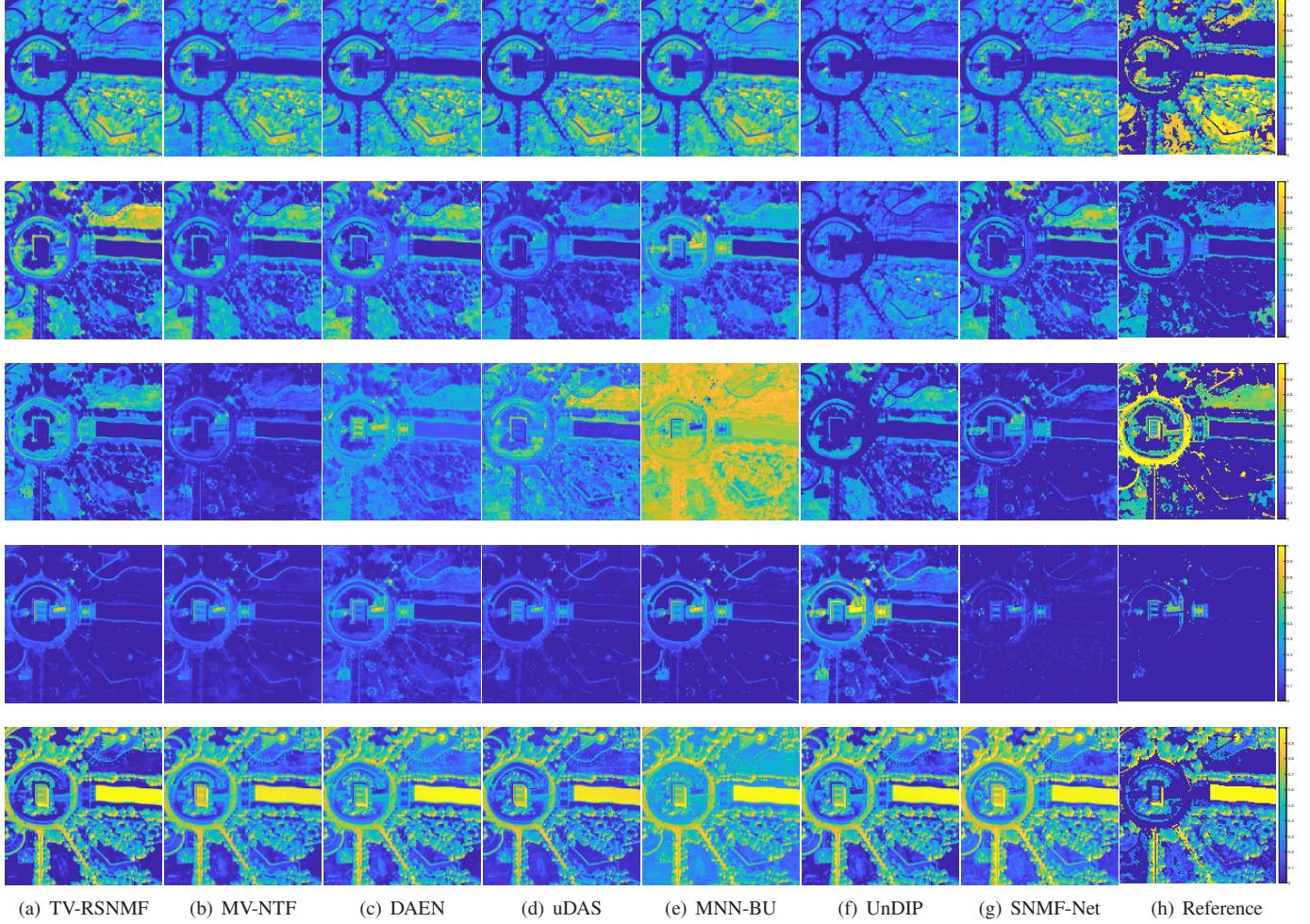


Fig. 15. Estimated abundance maps by seven unmixing algorithms on the HYDICE Urban dataset. From top to bottom, the rows are the abundance maps of tree, grass, street, roof, water respectively.

- [22] 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.*, vol. 57, no. 7, pp. 4309–4321, 2019.
- [23] S. Ozkan, B. Kaya, and G. B. Akar, “EndNet: Sparse autoencoder network for endmember extraction and hyperspectral unmixing,” *IEEE Trans. Geosci. Remote Sens.*, vol. 57, no. 1, pp. 482–496, 2019.
- [24] A. Min, Z. Guo, H. Li, and J. Peng, “JMnet: Joint metric neural network for hyperspectral unmixing,” *IEEE Trans. Geosci. Remote Sens.*, pp. 1–12, 2021.
- [25] Z. Dou, K. Gao, X. Zhang, H. Wang, and J. Wang, “Hyperspectral unmixing using orthogonal sparse prior-based autoencoder with hyper-laplacian loss and data-driven outlier detection,” *IEEE Trans. Geosci. Remote Sens.*, vol. 58, no. 9, pp. 6550–6564, 2020.
- [26] Y. Qu and H. Qi, “uDAS: An untied denoising autoencoder with sparsity for spectral unmixing,” *IEEE Trans. Geosci. Remote Sens.*, vol. 57, no. 3, pp. 1698–1712, 2019.
- [27] F. Khajehravani and H. Ghassemian, “Hyperspectral unmixing using deep convolutional autoencoders in a supervised scenario,” *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 13, pp. 567–576, 2020.
- [28] B. Palsson, M. O. Ulfarsson, and J. R. Sveinsson, “Convolutional autoencoder for spectral-spatial hyperspectral unmixing,” *IEEE Trans. Geosci. Remote Sens.*, pp. 1–15, 2020.
- [29] M. Zhao, X. Wang, J. Chen, and W. Chen, “A plug-and-play priors framework for hyperspectral unmixing,” *IEEE Trans. Geosci. Remote Sens.*, pp. 1–13, 2021.
- [30] B. Rasti, B. Koitala, P. Scheunders, and P. Ghamisi, “UnDIP: Hyperspectral unmixing using deep image prior,” *IEEE Trans. Geosci. Remote Sens.*, pp. 1–15, 2021.
- [31] Q. Xie, M. Zhou, Q. Zhao, Z. Xu, and D. Meng, “MHF-Net: An interpretable deep network for multispectral and hyperspectral image fusion,” *IEEE Trans. Pattern Anal. Mach. Intell.*, pp. 1–1, 2020.
- [32] X. Deng and P. L. Dragotti, “Deep coupled ISTA network for multi-modal image super-resolution,” *IEEE Trans. Image Process.*, vol. 29, pp. 1683–1698, 2020.
- [33] D. Liu, K. Sun, Z. Wang, R. Liu, and Z. J. Zha, “Frank-wolfe network: An interpretable deep structure for non-sparse coding,” *IEEE Trans. Circuits Syst. Video Technol.*, vol. 30, no. 9, pp. 3068–3080, 2020.
- [34] H. Wang, Q. Xie, Q. Zhao, and D. Meng, “A model-driven deep neural network for single image rain removal,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recog. (CVPR)*, June 2020.
- [35] C. J. Schuler, M. Hirsch, S. Harmeling, and B. Schlkopf, “Learning to deblur,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 38, no. 7, pp. 1439–1451, 2016.
- [36] Y. Li, M. Tofighi, J. Geng, V. Monga, and Y. C. Eldar, “Efficient and interpretable deep blind image deblurring via algorithm unrolling,” *IEEE Trans. Comput. Imaging*, vol. 6, pp. 666–681, 2020.
- [37] S. Zheng, S. Jayasumana, B. Romera-Paredes, V. Vineet, Z. Su, D. Du, C. Huang, and P. H. Torr, “Conditional random fields as recurrent neural networks,” in *Proc. Int. Conf. Computer Vision. (ICCV)*, 2015, pp. 1529–1537.
- [38] Z. Liu, X. Li, P. Luo, C. C. Loy, and X. Tang, “Deep learning markov random field for semantic segmentation,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 40, no. 8, pp. 1814–1828, 2018.
- [39] Y. Qian, F. Xiong, Q. Qian, and J. Zhou, “Spectral mixture model inspired network architectures for hyperspectral unmixing,” *IEEE Trans. Geosci. Remote Sens.*, vol. 58, no. 10, pp. 7418–7434, 2020.
- [40] T. Moreau and J. Bruna, “Understanding trainable sparse coding via matrix factorization,” in *Proc. Int. Conf. Learn. Representations (ICLR)*, 2016.
- [41] B. Xin, Y. Wang, W. Gao, D. Wipf, and B. Wang, “Maximal sparsity with deep networks?” in *Proc. Adv. Neural Inf. Process. (NeurIPS)*, 2016.
- [42] J. Liu and X. Chen, “ALISTA: Analytic weights are as good as learned

- weights in LISTA,” in *Proc. Int. Conf. Learn. Representations (ICLR)*, 2019.
- [43] C. Yang, Y. Gu, B. Chen, H. Ma, and H. C. So, “Learning proximal operator methods for nonconvex sparse recovery with theoretical guarantee,” *IEEE Trans. Signal Process.*, vol. 68, pp. 5244–5259, 2020.
- [44] L. Dong, Y. Yuan, and X. Luxi, “Spectral-spatial joint sparse nmf for hyperspectral unmixing,” *IEEE Trans. Geosci. Remote Sens.*, vol. 59, no. 3, pp. 2391–2402, 2021.
- [45] F. Xiong, J. Zhou, J. Lu, and Y. Qian, “Nonconvex nonseparable sparse nonnegative matrix factorization for hyperspectral unmixing,” *IEEE J. Sel. Topics Appl. Earth Observations Remote Sens.*, vol. 13, pp. 6088–6100, 2020.
- [46] X. Feng, H. Li, J. Li, Q. Du, A. Plaza, and W. J. Emery, “Hyperspectral unmixing using sparsity-constrained deep nonnegative matrix factorization with total variation,” *IEEE Trans. Geosci. Remote Sens.*, vol. 56, no. 10, pp. 6245–6257, 2018.
- [47] D. Hajinezhad, T.-H. Chang, X. Wang, Q. Shi, and M. Hong, “Nonnegative matrix factorization using ADMM: Algorithm and convergence analysis,” in *Proc. IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2016, pp. 4742–4746.
- [48] C.-J. Lin, “Projected gradient methods for nonnegative matrix factorization,” *Neural Comput.*, vol. 19, no. 10, pp. 2756–2779, 2007.
- [49] W. Zuo, D. Meng, L. Zhang, X. Feng, and D. Zhang, “A generalized iterated shrinkage algorithm for non-convex sparse coding,” in *Proc. Int. Conf. Computer Vision (ICCV)*, 2013, pp. 217–224.
- [50] X. Xie, J. Wu, G. Liu, Z. Zhong, and Z. Lin, “Differentiable linearized ADMM,” in *Proc. International Conference on Machine Learning (ICML)*, 2019.
- [51] J. Pan, Z. Hu, Z. Su, and M. Yang, “ L_0 -regularized intensity and gradient prior for deblurring text images and beyond,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 39, no. 2, pp. 342–355, 2017.
- [52] W. Zuo, D. Ren, D. Zhang, S. Gu, and L. Zhang, “Learning iteration-wise generalized shrinkage thresholding operators for blind deconvolution,” *IEEE Trans. Image Process.*, vol. 25, no. 4, pp. 1751–1764, 2016.
- [53] F. Xiong, J. Zhou, and Y. Qian, “Hyperspectral restoration via l_0 gradient regularized low-rank tensor factorization,” *IEEE Trans. Geosci. Remote Sens.*, vol. 57, no. 12, pp. 10410–10425, 2019.
- [54] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, “Going deeper with convolutions,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recog. (CVPR)*, 2015, pp. 1–9.
- [55] W. He, H. Zhang, and L. Zhang, “Total variation regularized reweighted sparse nonnegative matrix factorization for hyperspectral unmixing,” *IEEE Trans. Geosci. Remote Sens.*, vol. 55, no. 7, pp. 3909–3921, 2017.
- [56] R. Heylen, D. Burazerovic, and P. Scheunders, “Fully constrained least squares spectral unmixing by simplex projection,” *IEEE Trans. Geosci. Remote Sens.*, vol. 49, no. 11, pp. 4112–4122, 2011.



Jun Zhou (M’06-SM’12) received the B.S. degree in computer science and the B.E. degree in international business from Nanjing University of Science and Technology, Nanjing, China, in 1996 and 1998, respectively. He received the M.S. degree in computer science from Concordia University, Montreal, Canada, in 2002, and the Ph.D. degree from the University of Alberta, Edmonton, Canada, in 2006.

In June 2012, he joined the School of Information and Communication Technology, Griffith University, Nathan, Australia, where he is currently an Associate Professor. Prior to this appointment, he was a Research Fellow with the Research School of Computer Science, Australian National University, Canberra, Australia, and a Researcher with the Canberra Research Laboratory, NICTA, Canberra. His research interests include pattern recognitions, computer vision, and spectral imaging with their applications in remote sensing and environmental informatics.



Shuyin Tao received the B.E. degree in information engineering and the Ph.D. degree in optical engineering from Zhejiang University in 2009 and 2014, respectively. She is currently a Lecturer with the School of Computer Science and Engineering, Nanjing University of Science and Technology. Her research interests include image dehazing and image enhancement.



Jianfeng Lu (Member, IEEE) received his M.S. and Ph.D. degrees in Pattern Recognition and Intelligent System from the School of Computer Science and Engineering, Nanjing University of Science and Technology. He is currently a Professor with the School of Computer Science and Engineering, Nanjing University of Science and Technology. His research interests include image analysis, machine learning, data mining, and intelligent robot.



Fengchao Xiong (S’18-M’20) received the B.E. degree in software engineering from Shandong University, Jinan, China, in 2014 and the Ph.D. degree from the College of Computer Science, Zhejiang University, Hangzhou, China, in 2019.

He visited Wuhan University from 2011 to 2012 and Griffith University from 2017 to 2018. He is currently a Lecturer with the School of Computer Science and Engineering, Nanjing University of Science and Technology, Nanjing, China. His research interests include hyperspectral image processing, machine learning, and pattern recognition.



Yuntao Qian (M’04-SM’21) received the B.E. and M.E. degrees in automatic control from Xi’an Jiaotong University, Xi’an, China, in 1989 and 1992, respectively, and the Ph.D. degree in signal processing from Xidian University, Xi’an, China, in 1996.

During 1996–1998, he was a Postdoctoral Fellow with the Northwestern Polytechnical University, Xi’an, China. Since 1998, he has been with the College of Computer Science, Zhejiang University, Hangzhou, China, where he became a Professor in 2002. During 1999–2001, 2006, 2010, 2013, 2015–2016, and 2018 he was a Visiting Professor at Concordia University, Hong Kong Baptist University, Carnegie Mellon University, the Canberra Research Laboratory of NICTA, Macau University, and Griffith University. His current research interests include machine learning, signal and image processing, pattern recognition, and hyperspectral imaging. He is currently an Associate Editor of the IEEE JOURNAL OF SELECTED TOPICS IN APPLIED EARTH OBSERVATIONS AND REMOTE SENSING.