ADMM-Based Hyperspectral Unmixing Networks for Abundance and Endmember Estimation

Chao Zhou, Miguel R.D. Rodrigues

***Abstract*—Hyperspectral image (HSI) unmixing is an increasingly studied problem in various areas, including remote sensing. It has been tackled using both physical model-based approaches and more recently machine learning-based ones. In this paper, we propose a new HSI unmixing algorithm combining both model- and learning-based techniques, based on algorithm unrolling approaches, delivering improved unmixing performance. Our approach unrolls the Alternating Direction Method of Multipliers (ADMM) solver of a constrained sparse regression problem underlying a linear mixture model. We then propose a neural network structure for abundance estimation that can be trained using supervised learning techniques based on a new composite loss function. We also propose another neural network structure for blind unmixing that can be trained using unsupervised learning techniques. Our proposed networks are also shown to possess a lighter and richer structure containing less learnable parameters and more skip connections compared with other competing architectures. Extensive experiments show that the proposed methods can achieve much faster convergence and better performance even with a very small training dataset size when compared with other unmixing methods such as MNN-AE&BU, UnDIP and EGU-Net.**

***Index Terms*—Hyperspectral unmixing, unfolding, learning, ADMM, ISTA.**

I. Introduction

HYPERSPECTRAL Imaging (HSI) has been largely adopted in remote sensing in order to acquire information about an object or scene with no physical contact. It relies on the fact that different materials within a scene reflect electromagnetic radiation differently, so that, when the radiation captured by sensors is measured at each wavelength over a sufficiently broad spectral band, the resulting spectral signature can be used to uniquely identify and characterize the constituent materials within the scene [1].

The common linear mixture model (LMM) [2] assumes the spectral signature associated with each scene pixel corresponds to a linear mixture of spectral signatures corresponding to each material present within such pixel. This then calls for methods capable of quantitatively decomposing, or unmixing, the captured spectral signature onto its spectral constituents – also known as "endmembers" – and their proportions within the mixture – also known as "abundances" [3].

In general, HSI unmixing involves three key operations [4]: (a) *dimensionality reduction* (b) *endmember extraction* and (c) *abundance estimation*. [1] *Endmember extraction algorithms* focus on identifying endmember spectral signatures present within the HSI image. There are various endmember extraction methods such as Vertex Component Analysis (VCA) [5] and simplex volume maximization (SiVM) [6], where VCA assumes that there are pure pixels for each endmember in the HSI data, and SiVM assumes that endmembers are located in the vertex of the maximum simplex encompassing the HSI data. Subsequently, *abundance estimation algorithms* identify the proportion of each endmember within each pixel in the HSI image. There are popular methods such as fully constrained least square (FCLS) [7] which assumes the endmembers are extracted by the aforementioned methods. Sparse unmixing by variable splitting and augmented Lagrangian (SunSAL) [8] is also a popular sparsity driven abundance estimation method by referring to a rich endmember spectral library.

On the other hand, *blind unmixing* methods perform endmember extraction and abundance estimation simultaneously. For example, nonnegative matrix factorization (NMF) [9] and nonnegative tensor factorization (NTF) [10] are very popular blind unmixing methods that map the HSI unmixing problem onto a matrix/tensor factorization problem by imposing various constraints, such as total-variation constraint, on endmember signatures or their abundances. However, these model-based algorithms tend to be computationally complex, because they rely on iterative solvers such as alternating direction method of multiplier (ADMM) [11], making them unsuitable to real-time unmixing scenarios. Thus, the unmixing problem is still a challenging problem in literature.

More recently, with the surge of interest in machine learning, especially neural networks, many learning-based approaches (in *lieu* of above model-based ones) have been proposed to tackle the HSI unmixing challenge [12], [13], [14], [15], [16], using both supervised or unsupervised learning algorithms. Supervised learning-based approaches assume access to a set of pairs of HSI reflectances and the corresponding abundances [17], [18], [19]. Whereas, in unsupervised learning-based approaches, the machine learning algorithm attempts to learn a function to extract both the endmembers and abundances based on HSI reflectance data alone. In general, these unmixing methods [12], [16], [20], [21], [22], [23], [24], [25], [26] are based on an auto-encoder network structure that is capable of delivering both the endmembers

This work is a result of UCL project ARTICT. This work was also supported by EPSRC grant EP/R032785/1. The work of Chao Zhou was supported in part by China Scholarship Council.

C. Zhou and M. R.D. Rodrigues are with the Department of Electronic and Electrical Engineering, University College London, London, WC1E 6BT, U.K. (e-mail: chao.zhou.18@ucl.ac.uk; m.rodrigues@ucl.ac.uk).

[1]Dimension reduction is a typical pre-processing operation in image processing problems to reduce the data into a low-dimensional space, so we mainly focus on developing the endmember extraction and abundance estimation algorithms. In this paper, we assume the number of endmembers is known.

(via the corresponding decoder) and their abundances (via the corresponding bottleneck). However, without proper guidance or initialization, the network could yield meaningless estimations. Recently, some unmixing works have been proposed to utilize the existing unmixing algorithms to provide guidance for the training of unmixing networks. For example, UnDIP [27] depends on a simplex volume maximization algorithm to extract the endmembers, which are then used to guide the training of an abundance estimation network using a deep image prior. EGU-Net [28] propose a two-stream deep network that learns an additional network from the (nearly) pure endmembers obtained from HSI data via certain endmember extraction methods. To further improve unmixing performance, adversarial autoencoder [13], [14] is introduced to train an unmixing autoencoder in an adversarial manner with an additional discriminator. However, despite the fruitful developments, the machine learning techniques still have several drawbacks. First of all, there are no principled approaches to guide the design of network architectures, especially the encoder network. Secondly, neural networks such as the encoder network often lack interpretability. Thus it is difficult to incorporate any prior information about the tasks into the design of network architecture. Thirdly, current machine learning approaches generally rely on a huge training dataset size , lots of learning parameters and a very long period of training time to achieve satisfactory performance.

Notably, algorithm unrolling or unfolding techniques [29], [30] have emerged as a potential solution to design interpretable network structures. With these techniques, the unmixing task is firstly mapped onto an optimization problem, which is usually solved by an iterative solver. Then each step in the iterative solver could be converted to a network operation. Finally, the network is constructed by concatenating several iterations (or layers in network language) of such operation. It has been shown in [31], [32], [33] that the network unfolded from an iterative algorithm can achieve better performance than the original iterative algorithm.

Recently, [19], [34] have proposed ISTA based unmixing networks, MNN-AE and MNN-BU, which use the unfolding technique for HSI unmixing purposes. In particular, building upon an ISTA solver of a constrained sparsity linear regression unmixing problem, they design a deep network architecture that can be further trained in a supervised or unsupervised manner to solve the unmixing challenge. Similarly, [15] has proposed an unmixing network by unrolling an alternating optimisation algorithm of a sparsity constrained nonnegative matrix factorisation model. These approaches have delivered state-of-the-art results surpassing existing methods such as pixel-based CNN [18], DAEN [25] and uDAS [22].

However, the ISTA based unmixing networks [19], [34] are not very efficient and there are many iterative solvers other than ISTA. It is therefore relevant to understand whether the unmixing performance can be further improved by leveraging other iterative algorithms in algorithm unrolling approaches to deliver state-of-the-art unmixing neural network architectures.

In this paper, in light of the fact that ADMM is a very powerful optimization problem solver that can lead to sufficient accuracy with fast convergence [11], [35], we propose two new networks for abundance estimation and blind unmixing, respectively, both of which are derived by unfolding ADMM. In particular, we make these contributions:

1) Building upon the sparsity-driven HSI linear unmixing model, we propose a novel neural network architecture derived from unrolling an ADMM algorithm, which can be trained using supervised principles, in order to estimate the abundances from HSI data. Through a detailed analysis of network structure and complexity, we show that the proposed network has a much richer (i.e., more skip-connections) and lighter (i.e., less learnable parameters) structure. The experiment comparisons with state-of-art algorithms such as MNN-AE [34] also show our proposed methods achieve 3X faster convergence speed, 4X better performance when the size of the training dataset is small, and better robustness (along with interpretability).
2) Likewise, we also propose an autoencoder-like neural network that can be trained using unsupervised principles in order to yield both the endmembers and their abundances from the HSI data directly. The experiments showcase that the proposed blind unmixing networks offer improved performance in comparison to state-of-art algorithms, such as MNN-BU [34], uDAS [22], UnDIP [27], and EGU-Net [28].
3) We also propose new ways to improve the network's parameters tuning process. In view of the fact that a typical MSE loss has implicit Gaussian assumption, we propose to train the supervised network with a novel composite loss function incorporating additional terms such as an abundance angle distance (AAD) and an abundance information divergence (AID) term, which are usually used as unmixing performance evaluation metrics instead of training loss terms. The experiments show that the proposed loss function can provide better performance.

The rest of this paper is organized as follows. Section II introduces related background. Section III elaborates about how to leverage the ADMM algorithm to unfold constrained sparse regression problem leading up to our first unfolding network for abundance estimation, whereas Sections IV concentrate on the development of our unfolding ADMM based blind unmixing networks. Section V analyses the network structure and parameters complexity. Section VI offers a number of results showcasing the performance of the proposed approaches. Finally, we draw a number of conclusions and future research directions in Section VII.

Throughout this paper, we use lower-case letters to denote scalars, bold lower-case letters to denote column vectors, and bold upper-case letters to denote matrices. We also denote the i^{th} element of a vector $\boldsymbol{x}$ by x_i and the i^{th} column of a matrix $\boldsymbol{X}$ by $\boldsymbol{x}_i$. We denote the vector $\boldsymbol{x}$ at k^{th} iteration/layers as $\boldsymbol{x}^k$. We let $\mathbf{1}$ denote a vector with all ones. The matrix inverse and matrix transpose operators are represented by $(\cdot)^{-1}$ and $(\cdot)^T$, respectively. The ℓ_1-norm and ℓ_2-norm of a vector are represented by $||\cdot||_1$ and $||\cdot||_2$, respectively.

II. Related Background

We concentrate on popular HSI LMM, which entails that the reflectance spectrum corresponding to a particular pixel is a linear weighted combination of the spectrum of the endmembers associated with such pixel [2], [3]. This model can be described as follows:

$$\boldsymbol{y} = \boldsymbol{A}\boldsymbol{x} + \boldsymbol{n} \tag{1}$$

where $\boldsymbol{y} = [y_1, \ldots, y_p]^T \in R^{p\times 1}$ is a vector containing the reflectance spectra across p bands associated with a given pixel in the scene, $\boldsymbol{x} = [x_1, \ldots, x_r]^T$ is a vector containing the abundances of r different endmembers present in the pixel of the scene, and $\boldsymbol{n} = [n_1, \ldots, n_p]^T$ is a vector modelling additive noise. The matrix $\boldsymbol{A} = [\boldsymbol{a}_1, \ldots, \boldsymbol{a}_r] \in R^{p\times r}$ contains the spectral signatures of the endmembers present in the HSI data, i.e. $\boldsymbol{a}_m \in R^{p\times 1}$ models the spectral signature of the m^{th} endmember ($m = 1, \ldots, r$).

There are also two physical constraints associated with the abundance vector: (1) First, the different elements within the abundance vector must be nonnegative, i.e. $x_m \geq 0, m = 1, \ldots, r$; (2) Second, the different elements within the abundance vector also sum to one, i.e. $\sum_{m=1}^{r} x_m = 1$. These physical constraints are usually known as ANC and ASC, respectively. Moreover, the endmember matrix is also constrained to have non-negative entries.

When the endmember matrix $\boldsymbol{A}$ is available in the form of a spectral library [36], one can adopt a fully constraint least square (FCLS) optimization problem to recover the abundance vector from the reflectance vector as follows:

$$\min_{\boldsymbol{x}} \frac{1}{2}||\boldsymbol{y} - \boldsymbol{A}\boldsymbol{x}||_2, \; s.t., \boldsymbol{x} \geq \boldsymbol{0}, \text{and } \boldsymbol{1^T x} = 1 \tag{2}$$

Alternatively, in view of the fact that the recovered vector tends to be sparse, one can also adopt a constrained sparse regression (CSR) based optimization problem in order to recover the abundances from the reflectance as follows: [2]

$$\min_{\boldsymbol{x}} \frac{1}{2}||\boldsymbol{y} - \boldsymbol{A}\boldsymbol{x}||_2 + \lambda||\boldsymbol{x}||_1, s.t., \boldsymbol{x} \geq \boldsymbol{0} \tag{3}$$

where $\lambda \geq 0$ is a regularization parameter that controls the sparsity of solutions.

The CSR optimization problem in (3) can be solved using a range of solvers. In particular, one of the most popular solvers to CSR problem is the ADMM algorithm, which leads to the well-known SunSAL algorithm [8]. In particular, by introducing an auxiliary variable $\boldsymbol{z}$ such that $\boldsymbol{x} = \boldsymbol{z}$ along with a dual variable $\boldsymbol{d}$, ADMM leads to an iterative scheme to compute the solution of the constrained sparse regression problem appearing in (3) as follows:

$$\boldsymbol{x}^{k+1} = (\boldsymbol{A}^T\boldsymbol{A} + \mu\boldsymbol{I})^{-1}(\boldsymbol{A}^T\boldsymbol{y} + \mu(\boldsymbol{z}^k + \boldsymbol{d}^k)) \tag{4}$$

$$\boldsymbol{z}^{k+1} = \max\left(soft\left(\boldsymbol{x}^{k+1} - \boldsymbol{d}^k, \frac{\lambda}{\mu}\right), 0\right) \tag{5}$$

$$\boldsymbol{d}^{k+1} = \boldsymbol{d}^k - \left(\boldsymbol{x}^{k+1} - \boldsymbol{z}^{k+1}\right) \tag{6}$$

[2]Note that one does not enforce the ASC constraint in (3), because otherwise, the CSR problem would reduce to the FCLS one.

where $\boldsymbol{x}^k$ is the value of variable $\boldsymbol{x}$ at k^{th} iteration (same for $\boldsymbol{z}^k, \boldsymbol{d}^k$) and $\mu \geq 0$ is a parameter that is usually chosen to be an upper bound to the largest eigenvalue of $\boldsymbol{A}^T\boldsymbol{A}$. The operator $Soft$ in (5) is the soft-threshold operator given by, $soft(x, \theta) = sign(x)(|x| - \theta)_+$.

It is clear that this iterative scheme can be mapped onto different layer components of a neural network architecture using algorithm unrolling techniques [29]. In the next section, we will show our approach to supervised and unsupervised HSI unmixing building upon unfolding ADMM.

III. ADMM based Abundance Estimation Network

We first construct our ADMM based network structure for abundance estimation. We also introduce our new composite loss function underlying the optimization of our abundance estimation network.

A. Unfolding ADMM into a Neural Network Layer

The neural network layers consist of three distinct components deriving from unrolling the three different ADMM iterative operations appearing in (4), (5), and (6).

1) X-Update Component: The *X-update component* of the $(k+1)^{th}$ neural network layer is derived by unfolding (4). Concretely, we can write $(k+1)^{th}$ iterate $\boldsymbol{x}^{k+1}$ given the k^{th} estimates $\boldsymbol{z}^k$ and $\boldsymbol{d}^k$ along with $\boldsymbol{y}$ as follows:

$$\begin{aligned}\boldsymbol{x}^{k+1} &= f_X(\boldsymbol{z}^k, \boldsymbol{d}^k, \boldsymbol{y}; \boldsymbol{W}, \boldsymbol{B}) \\ &= \boldsymbol{W}^T\boldsymbol{y} + \boldsymbol{B}^T(\boldsymbol{z}^k + \boldsymbol{d}^k)\end{aligned} \tag{7}$$

where

$$\begin{aligned}\boldsymbol{W}^T &= (\boldsymbol{A}^T\boldsymbol{A} + \mu\boldsymbol{I})^{-1}\boldsymbol{A}^T \\ \boldsymbol{B}^T &= (\boldsymbol{A}^T\boldsymbol{A} + \mu\boldsymbol{I})^{-1}\mu\end{aligned} \tag{8}$$

However, in order to have greater flexibility, we will use learnable parameters $\boldsymbol{W}^{k+1} \in \mathbb{R}^{r\times p}$ and $\boldsymbol{B}^{k+1} \in \mathbb{R}^{r\times r}$ in this $(k+1)^{th}$ layer to replace the fixed parameter $\boldsymbol{W}$ and $\boldsymbol{B}$. The key idea is that these parameters $\boldsymbol{W}^{k+1}$ and $\boldsymbol{B}^{k+1}$ can differ from the original ones $\boldsymbol{W}$ and $\boldsymbol{B}$ associated with the model in (3), in order to better adapt to the characteristics of the data.

It is interesting to note that the operation performed by the *X-update component* can also be viewed as a typical linear layer [37] within a neural network. The *X-update component* of a neural network layer is shown in Fig. 1.

2) Z-Update Component: The *Z-update component* of the $(k+1)^{th}$ neural network layer is derived from unfolding (5). Concretely, the $(k+1)^{th}$ iterate $\boldsymbol{z}^{k+1}$ is obtained from the $(k+1)^{th}$ estimate $\boldsymbol{x}^{k+1}$ and the k^{th} estimate $\boldsymbol{d}^k$ by performing a soft-thresholding operation (with parameter λ/μ) followed by a max operation. We propose instead to re-express (5) as follows:

$$\begin{aligned}\boldsymbol{z}^{k+1} &= f_Z(\boldsymbol{x}^{k+1}, \boldsymbol{d}^k; \theta^{k+1}) \\ &= \max\left(soft\left(\boldsymbol{x}^{k+1} - \boldsymbol{d}^k, \theta^{k+1}\right), 0\right)\end{aligned} \tag{9}$$

where $\theta^{k+1} \in \mathbb{R}$ is a learnable parameter. The parameter θ^{k+1} – which can differ from layer to layer – plays the role of the parameter $\frac{\lambda}{\mu}$, with the advantage that it can also be

learnt in a data-driven manner in order to better adapt to the characteristics of the unmixing problem.

It is also interesting to note that the operation performed by the *Z-update component* can also be re-expressed as follows:

$$\boldsymbol{z}^{k+1} = ReLU(\boldsymbol{x}^{k+1} - \boldsymbol{d}^{k} - \theta^{k+1}\boldsymbol{I}) \tag{10}$$

where $ReLU(\cdot)$ is a component-wise rectified linear unit operation [37]. Therefore, the *Z-update component* – which also guarantees that one satisfies the ANC constraint – plays the role of a ReLU operation in a standard neural network. The *Z-update component* of a neural network layer is also shown in Fig. 1.

3) D-Update Component: Finally, the *D-Update Component* of the $(k+1)^{th}$ neural network layer is derived from unfolding (6). In particular, the $(k+1)^{th}$ iterate $\boldsymbol{d}^{k+1}$ is simply the difference between the k^{th} iterate $\boldsymbol{d}^{k}$ and the $(k+1)^{th}$ iterates $\boldsymbol{x}^{k+1}$ and $\boldsymbol{z}^{k+1}$. We propose however to re-express (6) as follows:

$$\begin{aligned}\boldsymbol{d}^{k+1} &= f_D(\boldsymbol{x}^{k+1}, \boldsymbol{z}^{k+1}, \boldsymbol{d}^{k}; \eta^{k+1}) \\ &= \boldsymbol{d}^{k} - \eta^{k+1}(\boldsymbol{x}^{k+1} - \boldsymbol{z}^{k+1})\end{aligned} \tag{11}$$

where η^{k+1} is also a learnable parameter offering additional flexibility. Such a learnable parameter plays the role of a step-size that can be further adjusted in order to adapt to the characteristics of the unmixing problem. The *D-update component* of a neural network layer is shown in Fig. 1.

4) Overall Neural Network Layer: By intertwining the *X-update*, *Z-update*, and *D-update* components, one immediately obtains the structure of a neural network layer shown in Fig. 1. Note that each network layer corresponds to an iteration of

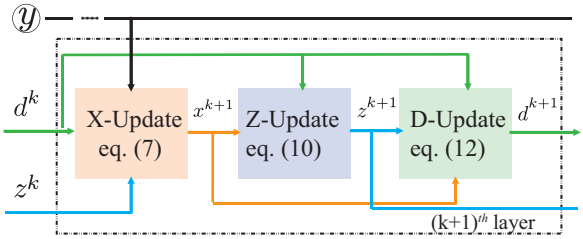


Fig. 1. The neural network layer consists of X-update, Z-update and D-update components, which performs eq.(7), eq.(9) and eq.(11) respectively. It mimics one iteration of the ADMM algorithm. There are four shortcuts: one from d^k to Z-update and D-update components, one from x^{k+1} to the D-update component, one from z^{k+1} to the X-update component in next layer, and one from the input $\boldsymbol{y}$.

the original ADMM algorithm. However, as discussed above, each layer is associated with learnable parameters $\boldsymbol{\Theta}^k = \{\boldsymbol{W}^k, \boldsymbol{B}^k, \theta^k, \eta^k\}$, whereas each iteration in the ADMM algorithm is associated instead with hand-coded parameters $\{\boldsymbol{A}, \mu, \lambda\}$. One of the advantages of using such a learnable parameterization is that the number of layers in the resulting network architecture can be much lower than the number of iterations associated with the original ADMM algorithm in order to achieve the desired performance level.

It is interesting to note that in Fig.1, there are four shortcuts: one from d^k to Z-update and D-update components, one from x^{k+1} to the D-update component, one from z^{k+1} to the X-update component in next layer, and one from the input $\boldsymbol{y}$. Whereas in a conventional neural network, the output of the previous component is only connected directly to the next component and the network input $\boldsymbol{y}$ is usually only connected to the first layer of the network.

B. Abundance Estimation Network Structure

We have previously suggested that each iteration block – corresponding to one iteration of the ADMM algorithm – can be seen as a neural network layer. We have also previously suggested that the model parameters $\{\boldsymbol{A}, \mu, \lambda\}$ associated with the ADMM algorithm can be replaced by learnable ones in each neural network layer. Therefore, we can construct two different feed-forward neural networks by concatenating K iteration blocks, in order to mimic K iterations of the ADMM algorithm.

Our first network (designated U-ADMM-AENet-I) results from concatenating K iteration blocks/layers – consisting of *X-Update*, *Z-Update*, and *D-Update* components – where the learnable parameters $\boldsymbol{\Theta}^k = \{\boldsymbol{W}', \boldsymbol{B}', \theta', \eta'\}, \forall k \in [1, K]$ are tied/shared across the network. A K-layer U-ADMM-AENet-I is shown in Fig.2a.

Our second network (designated by U-ADMM-AENet-II) also results from concatenating K iteration blocks/layers, but the learnable parameters are now not shared across layers but are rather layer-specific (i.e. $\boldsymbol{\Theta}^k = \{\boldsymbol{W}^k, \boldsymbol{B}^k, \theta^k, \eta^k\}, \forall k \in [1, K]$). Note that one potential advantage of U-ADMM-AENet-II in relation to U-ADMM-AENet-I relates to its increased capacity and flexibility, leading to improved unmixing results. A K-layer U-ADMM-AENet-II is shown in Fig.2b.

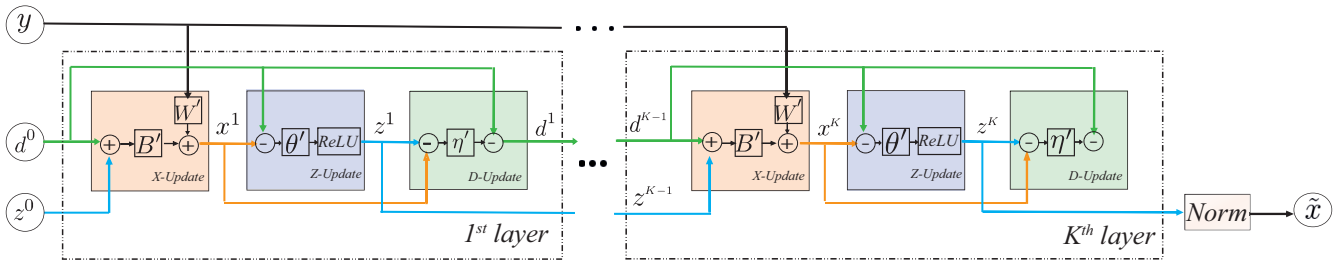
In both cases, the network takes as input the HSI spectrum $\boldsymbol{y}$, accompanied with the pseudo inputs $\boldsymbol{z}^0$ and $\boldsymbol{d}^0$, which we both set to 0 in line with the standard approach in ADMM algorithms. Then these inputs will go through K network layers with learnable parameters $\boldsymbol{\Theta} = \{\boldsymbol{\Theta}^k\}_{k=1}^K$. We also let the network output to be derived from the *Z-Update* component in the last iteration block because this component guarantees compliance with the ANC constraint. We also further normalize the network output using l_1 normalisation to meet the ASC constraint. Suppose the network output before normalisation is $\boldsymbol{z}$, then the normalised output $\tilde{\boldsymbol{x}}$ is computed as follows:

$$\tilde{\boldsymbol{x}}_i = \frac{\boldsymbol{z}_i}{\sum_j \boldsymbol{z}_j} \tag{12}$$

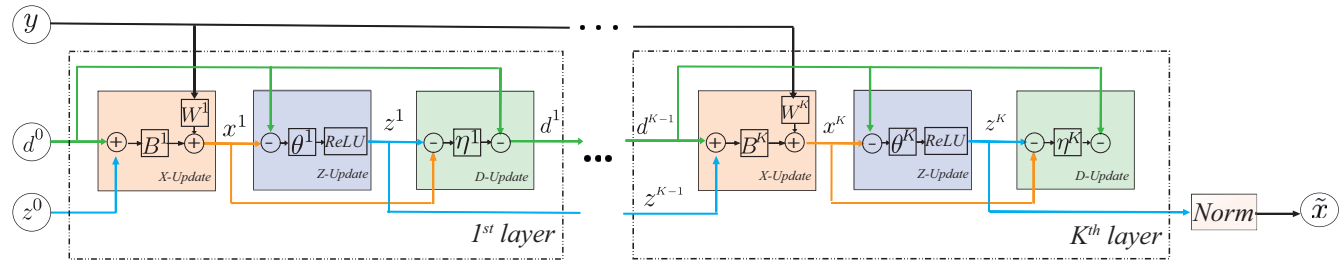
Note that $\boldsymbol{z}$ is guaranteed to be non-negative because it is the output of the ReLU operator. We add this normalisation because ASC is not directly imposed within the optimization formulation.

C. Abundance Estimation Network Initialization and Training strategies

1) Initialization Approach: The neural networks are trained by adopting warm instead of random initialization of the parameters in order to speed up the training procedure. In



(a) A K-layer U-ADMM-AENet-I.



(b) A K-layer U-ADMM-AENet-II.

Fig. 2. The structures of U-ADMM-AENet. (a) a K-layer U-ADMM-AENet-I, of which the learnable parameters in each layer are same/tied/shared, $\boldsymbol{\Theta}^k = \{\boldsymbol{W}', \boldsymbol{B}', \theta', \eta'\}, \forall k \in [1, K]$. (b) a K-layer U-ADMM-AENet-II, of which the learnable parameters in each layer are different/untied/unshared, $\boldsymbol{\Theta}^k = \{\boldsymbol{W}^k, \boldsymbol{B}^k, \theta^k, \eta^k\}, \forall k \in [1, K]$.

particular, we propose to initialize the different network parameters by leveraging the original parameters associated with the ADMM algorithms as follows:

- The parameters $\boldsymbol{W}^k$ and $\boldsymbol{B}^k$ corresponding to the *X-update component* of each layer are initialized by using (8). The endmember signature matrix $\boldsymbol{A}$ can be chosen to correspond to some adequate spectral library – where it is known – or else one can estimate it using an existing algorithm such as VCA [5], which is then concatenated with any candidate signatures that are related.
- The parameter θ^k corresponding to the *Z-update component* of each layer is initialised using λ/μ, where λ and μ relate to parameters associated with the ADMM algorithm.
- Finally, the parameter η^k associated with the *D-update component* of each layer does not have an immediate counterpart in the ADMM algorithm. We set, however, this parameter to be equal to one for initialisation purposes.

Note that with this initialisation strategy, an K-layer U-ADMM-AENet corresponds exactly to K iterations of the ADMM algorithm.

2) Training Approach: The neural networks are trained using a supervised learning approach, by leveraging access to a training set $D = \{\boldsymbol{y}_i, \boldsymbol{x}_i\}_{i=1}^N$ consisting of N reflectance spectra $\boldsymbol{y}_i$ and corresponding abundances $\boldsymbol{x}_i$.

Instead of commonly used MSE loss function[34], we propose a new composite loss function to train the neural networks, which is given by:

$$L_{\boldsymbol{\Theta}} = \alpha_1 \cdot L_1 + \alpha_2 \cdot L_2 + \alpha_3 \cdot L_3 \tag{13}$$

where α_1, α_2, and α_3 are hyper-parameters controlling the contribution of each loss function components L_1, L_2, and L_3, respectively.

The first component of the loss function derives from the standard mean-squared error (MSE) given by:

$$\begin{aligned} L_1 &= MSE(\{\boldsymbol{x}_i, \tilde{\boldsymbol{x}}_i\}_{i=1}^N) \\ &= \frac{1}{N} \sum_{\{\boldsymbol{y}_i, \boldsymbol{x}_i\} \in D} \|\boldsymbol{x}_i - \tilde{\boldsymbol{x}}_i(\boldsymbol{y}_i)\|_2^2 \end{aligned} \tag{14}$$

where $\boldsymbol{y}_i$ is the i^{th} reflectance spectrum within the training set, $\boldsymbol{x}_i$ is the corresponding i^{th} abundance vector within the training set, and $\tilde{\boldsymbol{x}}_i(\boldsymbol{y}_i)$ corresponds to the network estimate of $\boldsymbol{x}_i$ given $\boldsymbol{y}_i$. This loss function is often adopted to train neural networks for unmixing purposes [19].

The second component of the loss function derives from the abundance angle distance (AAD) [18], [22], [38], [39], [40], as follows:

$$\begin{aligned} L_2 &= \frac{1}{N} \sum_{\{\boldsymbol{y}_i, \boldsymbol{x}_i\} \in D} AAD(\boldsymbol{x}_i, \tilde{\boldsymbol{x}}_i(\boldsymbol{y}_i)) \\ &= \frac{1}{N} \sum_{\{\boldsymbol{y}_i, \boldsymbol{x}_i\} \in D} \cos^{-1}\left(\frac{\boldsymbol{x}_i^T \tilde{\boldsymbol{x}}_i(\boldsymbol{y}_i)}{\|\boldsymbol{x}_i\|_2 \|\tilde{\boldsymbol{x}}_i(\boldsymbol{y}_i)\|_2}\right) \end{aligned} \tag{15}$$

The third component of the loss function derives instead from the abundance information divergence (AID) [22], [39] as follows:

$$\begin{aligned} L_3 &= \frac{1}{N} \sum_{\{\boldsymbol{y}_i, \boldsymbol{x}_i\} \in D} AID(\boldsymbol{x}_i, \tilde{\boldsymbol{x}}_i(\boldsymbol{y}_i)) \\ &= \frac{1}{N} \sum_{\{\boldsymbol{y}_i, \boldsymbol{x}_i\} \in D} KL(\boldsymbol{x}_i | \tilde{\boldsymbol{x}}_i(\boldsymbol{y}_i)) + KL(\tilde{\boldsymbol{x}}_i(\boldsymbol{y}_i) | \boldsymbol{x}_i) \end{aligned} \tag{16}$$

where

$$KL(\boldsymbol{x}_i | \tilde{\boldsymbol{x}}_i(\boldsymbol{y}_i)) = \sum_{m=1}^{r} \left(x_{i,m} \log\left(\frac{x_{i,m}}{\tilde{x}_{i,m}}\right)\right) \tag{17}$$

Note that the second and third components of the loss functions are additional dissimilarity measures, enabling to gauge how different the recovered abundance is from the true one.

We introduce these two additional components for two reasons: (1) the MSE loss implicitly assumes [41] that the abundance vectors estimated by neural networks are sampled from a Gaussian distribution, which is obviously not suitable as abundance vectors have ANC and ASC constraints. On the other hand, the AAD does not have such an assumption, as it corresponds to a high-dimensional generalization of an angle between vectors. (2) abundance vectors, given the ANC and ASC constraints, have a natural probabilistic interpretation. The Kullback-Leibler distance is often used to measure the distance between two probability distributions (such as two abundance vectors). AID is a symmetric version of the KL divergence that can also be used to capture the distance between two abundance vectors. Note that in [20], [24], the authors use the SID and SAD losses respectively within their overall loss function to measure HSI reflectance reconstruction performance. Instead, we use a richer combination of loss functions within our overall loss in order to capture abundance estimation performance. The experiments shown later indicate that this richer combination leads to improved performance.

3) Parameter Update Rule: We finally learn the parameters $\mathbf{\Theta}$ of the proposed neural networks *U-ADMM-AENet-I* and *U-ADMM-AENet-II* via minimizing the loss function $L_{\mathbf{\Theta}}$ using the stochastic gradient descent algorithm ADAM [42], while the hyper-parameters are optimised via cross-validation techniques, leading to $\alpha_1 = 1.0$, $\alpha_2 = 10^{-7}$, $\alpha_3 = 10^{-5}$. The stochastic gradient algorithm updates the learnable parameters $\mathbf{\Theta}$ as follows:

$$\mathbf{\Theta}_{new} = \mathbf{\Theta}_{old} - l\frac{\partial L_{\mathbf{\Theta}_{old}}}{\partial \mathbf{\Theta}_{old}} \tag{18}$$

where, l is the learning rate. The gradient for each learnable parameter in $\mathbf{\Theta}$ can be calculated via the back-propagation algorithm [37]. While the forward computation of a K-layer network, which starts from the input to the output, follows the path $\boldsymbol{y} \rightarrow \mathbf{\Theta}^1 \rightarrow \cdots \rightarrow \mathbf{\Theta}^k \rightarrow \cdots \rightarrow \mathbf{\Theta}^K \rightarrow \tilde{\boldsymbol{x}}$, the back-propagation algorithm calculates the gradient for the parameters in each layer in a reverse path as follows:

$$\frac{\partial L_{\mathbf{\Theta}}}{\partial \mathbf{\Theta}^k} = \frac{\partial L_{\mathbf{\Theta}}}{\partial \tilde{\boldsymbol{x}}}\frac{\partial \tilde{\boldsymbol{x}}}{\partial \mathbf{\Theta}^K}\cdots\frac{\partial \mathbf{\Theta}^{k+1}}{\partial \mathbf{\Theta}^k},\ k = 1, \cdots, K-1 \tag{19}$$

For example, the gradient for parameter θ^K in the last layer can be calculated as follows:

$$\frac{\partial L_{\mathbf{\Theta}}}{\partial \theta^K} = \frac{\partial L_{\mathbf{\Theta}}}{\partial \tilde{\boldsymbol{x}}}\frac{\partial \tilde{\boldsymbol{x}}}{\partial \theta^K} \tag{20}$$

where,

$$\frac{\partial L_{\mathbf{\Theta}}}{\partial \tilde{\boldsymbol{x}}} = \alpha_1\frac{\partial L_1}{\partial \tilde{\boldsymbol{x}}} + \alpha_2\frac{\partial L_2}{\partial \tilde{\boldsymbol{x}}} + \alpha_3\frac{\partial L_3}{\partial \tilde{\boldsymbol{x}}} \tag{21}$$

$$\frac{\partial \tilde{\boldsymbol{x}}}{\partial \theta^K} = \frac{\partial}{\partial \theta^K} f_Z(\boldsymbol{x}^K, \boldsymbol{d}^{K-1}; \theta^K) \tag{22}$$

according to eq.(13) and eq.(9). Note that on the right side of eq.(22) is f_Z instead of f_X because the network output $\tilde{\boldsymbol{x}}$ is derived from the *Z-Update* component in the last layer. The gradient for the remaining parameters can be calculated in a similar way.

This stochastic gradient algorithm then will run until predefined conditions are met. After training, the network can perform fast estimation of the abundance vector given a new HSI spectrum as input, because it only requires one forward network computation.

IV. ADMM BASED BLIND UNMIXING NETWORK

We can likewise also construct a neural network that can be trained (in an unsupervised manner) to estimate both endmembers and their abundances based on HSI reflectance data.

A. Blind Unmixing Network Structure

Our ADMM based blind unmixing network is inspired by the previously proposed abundance estimation network. In particular, the network architecture follows from the previous U-ADMM-AENet by appending an additional linear layer, yielding a reconstruction of the original HSI spectrum from the estimation of the abundances as follows:

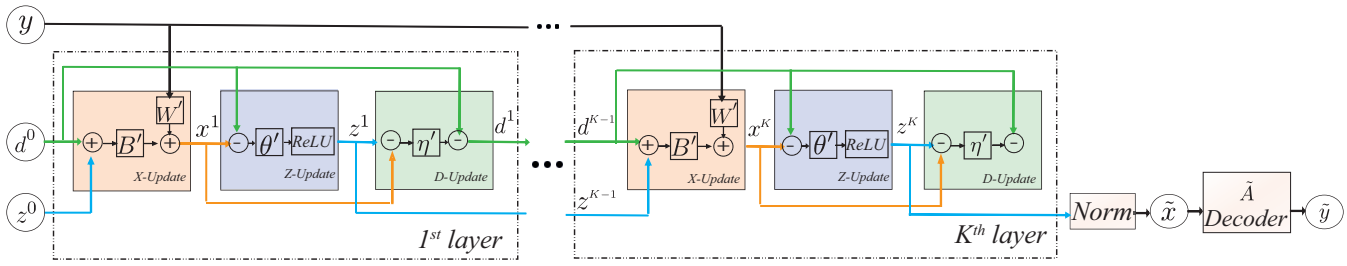
$$\tilde{\boldsymbol{y}} = \tilde{\boldsymbol{A}}\tilde{\boldsymbol{x}} \tag{23}$$

where $\tilde{\boldsymbol{x}}$ represents the estimated abundance by U-ADMM-AENet and $\tilde{\boldsymbol{y}}$ represents the reflectance spectrum reconstruction. The matrix $\tilde{\boldsymbol{A}}$ – corresponding to the additional linear layer – models the endmember signatures; this matrix is also non-negative because of physical constraints. Note that such an auto-encoder like structure has also been adopted in, e.g. uDAS [22] or MNN-BU [34]. However, uDAS is a conventional auto-encoder structure with a black-box encoder, and MNN-BU is an auto-encoder with the encoder derived from ISTA, while ours is a network structure derived from unfolding ADMM.

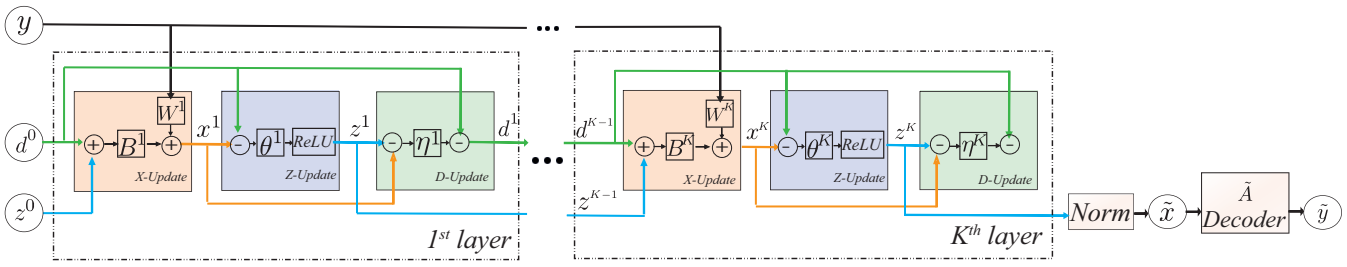
We can also construct two different blind unmixing networks corresponding to the two previous abundance estimation networks. When the network parameters are shared across the different network layers, we name the network as U-ADMM-BUNet-I and when the parameters are unshared, we name it as U-ADMM-BUNet-II (BU stands for blind unmixing). A K-layer U-ADMM-BUNet structure is shown in Fig. 3.

B. Blind Unmixing Network Initialization and Training strategies

1) Initialization Approach: The neural networks are also trained by adopting warm instead of random initialisation of the parameters. The parameters associated with the $X-update$ component, $Z-update$ component and $D-update$ component of each layer in the U-ADMM-BUNet are initialised by adopting the strategy used to initialise the same parameters in U-ADMM-AENet. In contrast, the extra decoding layer in U-ADMM-BUNet is initialised by taking it to correspond to some estimate of the endmember signature matrix, which can be obtained in the same way in III-C1.



(a) A K-layer U-ADMM-BUNet-I.



(b) A K-layer U-ADMM-BUNet-II.

Fig. 3. The structures of U-ADMM-BUNet. (a) a K-layer U-ADMM-BUNet-I, which is constructed by appending U-ADMM-AENet-I with a linear decoder layer. (b) a K-layer U-ADMM-BUNet-II, which is constructed by appending U-ADMM-AENet-II with a linear decoder layer.

2) Training Approach: The proposed networks, which are akin to well-known auto-encoders, can be trained in a completely unsupervised way. By assuming access to a training set $D = \{\boldsymbol{y}_i\}_{i=1}^N$ consisting of N reflectance spectra $\boldsymbol{y}_i$. We can then construct a loss function given by:

$$L_{\boldsymbol{\Theta}} = MSE(\{\boldsymbol{y}_i, \tilde{\boldsymbol{y}}_i\}_{i=1}^N) = \frac{1}{N} \sum_{\{\boldsymbol{y}_i\} \in D} \|\boldsymbol{y}_i - \tilde{\boldsymbol{y}}_i\|_2^2 \quad (24)$$

Note that this loss function is simpler than the previous one since we cannot use either AAD or AID due to the absence of abundance in the training set.

After learning, the U-ADMM-BUNet would output a reconstruction of reflectance spectra $\tilde{\boldsymbol{y}}_i$. Note that U-ADMM-BUNet is constructed by concatenating U-ADMM-AENet with a linear decoder layer with parameter $\tilde{\boldsymbol{A}}$, as a result, $\tilde{\boldsymbol{y}}_i$ can be expressed as follows:

$$\tilde{\boldsymbol{y}}_i = \tilde{\boldsymbol{A}} f_{AE}(\boldsymbol{y}_i) \quad (25)$$

where, $f_{AE}(\boldsymbol{y}_i)$ is the output of U-ADMM-AENet, which is the abundance estimation corresponding to the spectra $\boldsymbol{y}_i$, $\tilde{\boldsymbol{x}}_i = f_{AE}(\boldsymbol{y}_i)$. The parameter $\tilde{\boldsymbol{A}}$ then would correspond to the estimation of the endmember matrix.

3) Parameter Update Rule: Like the previous case, the network parameters are optimised using stochastic gradient descent algorithm ADAM [42], where the gradient for each parameter can be calculated via the back-propagation algorithm. In particular, the gradient of loss function with respect to the decoder parameters $\tilde{\boldsymbol{A}}$ and encoder output $\tilde{\boldsymbol{x}}_i$ are as follows,

$$\frac{\partial L_{\boldsymbol{\Theta}}}{\partial \tilde{\boldsymbol{A}}} = \frac{2}{N} \sum_{\{\boldsymbol{y}_i\} \in D} (\tilde{\boldsymbol{A}}\tilde{\boldsymbol{x}}_i - \boldsymbol{y}_i)\tilde{\boldsymbol{x}}_i^T \quad (26)$$

$$\frac{\partial L_{\boldsymbol{\Theta}}}{\partial \tilde{\boldsymbol{x}}_i} = \frac{2}{N} \tilde{\boldsymbol{A}}^T (\tilde{\boldsymbol{A}}\tilde{\boldsymbol{x}}_i - \boldsymbol{y}_i) \quad (27)$$

The remaining gradients are calculated in the same way as U-ADMM-AENet. Note that during each update, the decoder parameter $\tilde{\boldsymbol{A}}$ is enforced to be non-negative in line with the physical constraints.

V. Network Structure/Complexity Analysis

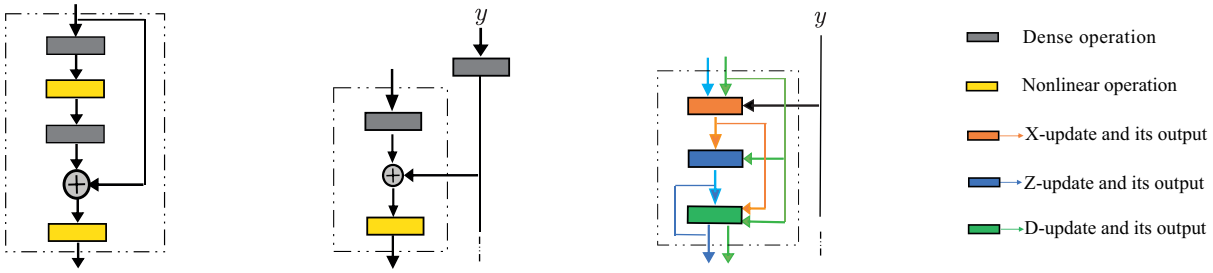
In this section, we will discuss the network complexity from both structure and parameter perspectives.

A. Network Structure

It is interesting to contrast the structure of the resulting ADMM based layer with the structure of layers in other unfolded networks used to solve the CSR problem. Fig.4 depicts the structure of various network layers including, (1) a typical ResNet [43] which is well-known for its performance improvement by using skip connections, (2) a network layer deriving from unfolding the CSR problem using the ISTA solver [19], and (3) the network layer deriving from unfolding the CSR problem using the proposed ADMM method. In particular, it is interesting to note that our proposed network layer can contain far many more shortcuts and skip connections in comparison with the competing layers. For example, ResNet only introduces shortcuts between adjacent operations, and the ISTA based layer includes skipping connections only from the input, whereas our ADMM based layer contains both types of connections. It will be shown that the proposed network can therefore lead to better unmixing results compared to competing ones, since such richer connections can improve neural network performance in various tasks [44], [45], [46], [47].

B. Network Complexity

It is also interesting to comment on the number of learnable parameters of the proposed architecture in relation to existing



(a) ResNet. (b) Unfolding ISTA. (c) Unfolding ADMM.

Fig. 4. Comparison of layer structures of various networks. (a) ResNet layer structure, which introduces shortcuts between adjacent operations. (b) ISTA based layer structure, which includes skipping connections only from the input. (c) Our ADMM based layer structure, which contains both types of connections

state-of-the-art ones. This acts as a proxy to gauge the complexity of the proposed networks. As it has been shown in [34] that unfolding networks generally have much less learnable parameters compared to conventional networks, we now will briefly compare our proposed U-ADMM-AENet-I &II derived from unfolding ADMM to other architectures derived from unfolding ISTA [34].

In U-ADMM-AENet-I, where the different parameters are shared across layers, the total number of parameters is $(r^2 + rp+2)$ where r^2 correspond to the matrix $\boldsymbol{B}'$, rp correspond to the matrix $\boldsymbol{W}'$, and the remaining ones relate to the two scalars θ', η'. In U-ADMM-AENet-II, where the different parameters are specific to each layer, there are $(r^2 + rp + 2)K$ learnable parameters, where K is the number of iteration blocks/layers in the network. In contrast, in similar networks derived from unfolding ISTA such as MNN-AE-1 (where parameters are shared across layers) and MNN-AE-2 (where the parameters are specific to each layer), the number of learnable parameters is $(r^2 + rp + 1)$ and $(r^2 + rp + 1)K$, respectively. We report the number of learnable parameters of various abundance estimation networks and blind unmixing networks in Table I and Table II respectively.

In summary, the number of learnable parameters in the proposed network is much less than most state-of-art unmixing networks such as UnDIP and EGU-Net. One advantages of the lighter (i.e., less learnable parameters) machine learning model is that it is less likely to suffer from overfitting [41]. It will be seen that the proposed network can lead to better performance than competing ones.

TABLE I
NUMBER OF LEARNABLE PARAMETERS: ABUNDANCE ESTIMATION

method	MNN-AE-1	MNN-AE-2	U-ADMM-AENet-I	U-ADMM-AENet-II
♯	1.3×10^3	2.7×10^3	1.3×10^3	2.7×10^3

VI. EXPERIMENTS

We now conduct extensive experiments on both synthetic data and real hyperspectral data to demonstrate the effectiveness of the proposed methods. Table III lists our various methods, their abbreviations, and their succinct description.

TABLE II
NUMBER OF LEARNABLE PARAMETERS: BLIND UNMIXING

method	MNN-BU-1	MNN-BU-2	U-ADMM-BUNet-I
♯	2.7×10^3	5.4×10^3	2.7×10^3
method	U-ADMM-BUNet-II	UnDIP	EGU-Net-pw
♯	5.4×10^3	1.3×10^6	1.9×10^5

TABLE III
PROPOSED METHODS, INCLUDING ABBREVIATION AND SUCCINCT DESCRIPTION

U-ADMM -AENet-I(II)	ADMM based abundance estimation network, I(II) represents (un)shared parameters
U-ADMM -BUNet-I(II)	ADMM based blind unmixing network, I(II) represents (un)shared parameters

A. Performance Metrics

We employ various metrics to gauge the performance of our unmixing algorithms. In particular, we use the well-known root mean square error (RMSE) and mean absolute error (MAE) [27] between the i^{th} true abundance vector $\boldsymbol{x}_i$ and the estimate of the i^{th} abundance vector $\tilde{\boldsymbol{x}}_i$ to quantify the abundance estimation quality. This is given by:

$$RMSE_i = \sqrt{\frac{1}{r}\sum_{m=1}^{r}(x_{i,m} - \tilde{x}_{i,m})^2} \tag{28}$$

$$MAE_i = \frac{1}{r}\sum_{m=1}^{r}|x_{i,m} - \tilde{x}_{i,m}| \times 100 \tag{29}$$

The RMSE and MAE are then further averaged over different instances within a test set.

We also use AAD (see (15)) and AID (see (16)) between the i^{th} true abundance vector $\boldsymbol{x}_i$ and the estimate of the i^{th} abundance vector $\tilde{\boldsymbol{x}}_i$ to quantify the abundance estimation quality. These quantities are also further averaged over different instances within a test set.

We employ the spectrum angle distance (SAD) [22] to measure the dissimilarity between a true endmember signature $\boldsymbol{a}_m$ and its estimate $\tilde{\boldsymbol{a}}_m$, where SAD has the same mathematical formulation as AAD. This metric is averaged instead over the different endmembers.

B. Data

We also employ various datasets, including synthetically generated and real ones, to gauge the performance of the various unmixing algorithms.

1) Synthetic Data: We adopt the well-known procedure in [9] to generate a dataset containing synthetic HSI spectral data as follows:

1) *Endmember generation* We generate the endmembers underlying the HSI spectral data by selecting mineral signatures from the well-known USGS spectral library denoted as splib06 [36]. This library contains spectral reflectance values associated with different minerals spanning the range 0.4 μm to 2.5 μm over 224 channels. We randomly select six spectral signatures. This process results in a 224×6 endmember matrix. The signature of these 6 endmembers is shown in Fig. 5.
2) *Abundance generation*. We then generate the different abundances underlying the HSI spectral data as follows. First, we extract a^2 disjoint patches of size $a \times a$ pixels from a synthetic image of size $a^2 \times a^2$ pixels. Second, for all pixels of a given patch, we assign a spectral signature corresponding to a mixture of two randomly selected endmembers with fractions γ and $1-\gamma$. Finally, in order to generate highly mixed synthetic HSI data, the abundance map is convolved with a Gaussian filter of size $(a+1) \times (a+1)$, whose variance is equal to 2, followed by further re-scaling to meet the ASC constraint per pixel. We set $a = 10$ and $\gamma = 0.8$.
3) *Noise contamination*. Finally, to understand the robustness of the unmixing algorithms against noise, we also contaminated the generated HSI data with additive white Gaussian noise (AWGN). We define the signal-to-noise ratio (SNR) of the corrupted signal as follows:

$$SNR = 10 \log_{10} \frac{E[\boldsymbol{x}^T \boldsymbol{A}^T \boldsymbol{A} \boldsymbol{x}]}{E[\boldsymbol{n}^T \boldsymbol{n}]} \tag{30}$$

where $E[\cdot]$ is the statistical expectation operator.

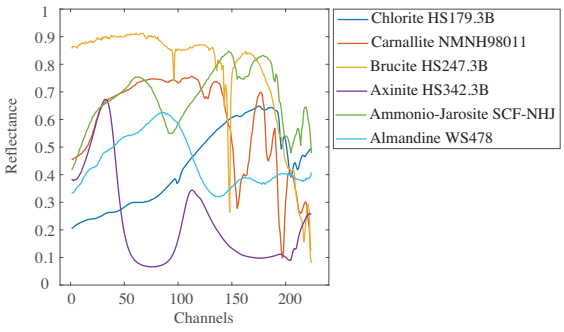
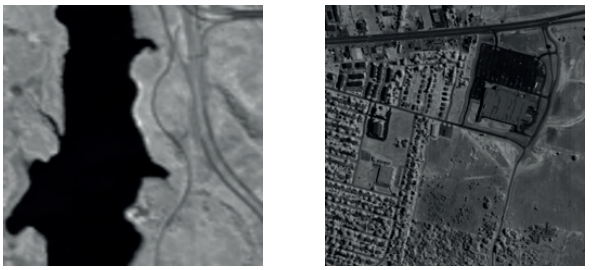


Fig. 5. Endmember signatures for synthetic data.

Then, the training pixels will be randomly picked from the synthetic HSI data, while the remaining data will be used for evaluation.

2) Real Data: We also adopt two commonly used real HSI datasets.

1) *Jasper Ridge*. We adopt the widely used Jasper Ridge HSI dataset [48]. This dataset contains 512×614 pixels images recorded using 224 channels ranging from 380 nm up to 2500 nm with a spectral resolution of 9.46 nm. There are also four different endmembers in the scene: Road, Soil, Water, and Tree. We consider 100×100 pixels sub-images of the original images in order to reduce computational burden and deploy faster experimental studies. We also only consider 198 out of the 224 channels for unmixing purposes: channels 1-3, 108-112, 154-166 and 220-224 are removed due to dense water vapour and atmospheric effects. A representative image – associated with the 80th channel image – is shown in Fig. 6a.
2) *Urban*. We also adopt the widely used Urban HSI dataset [49], [50]. It contains 307×307-pixel images with a spatial resolution of $2 \times 2m^2$ sensed using 210 channels ranging from 400 nm to 2500 nm with a spectral resolution of 10 nm. There are three versions of the ground truth coupled with this dataset, but we use the one containing four endmembers: Asphalt, Grass, Tree, and Roof. We also process the data further, whereby we only use 162 out of the original 210 channels by discarding channels 1-4, 76, 87, 101-111, 136-153 and 198-210 due to the dense water vapour and atmospheric effects. A representative image – associated with the 80th channel image – is also shown in Fig. 6b.



(a) Jasper Ridge (b) Urban

Fig. 6. HSI image at 80th channel. (a) Jasper Ridge. (b) Urban.

C. Experiments on Synthetic Data

We now evaluate the performance of our approaches on the synthetic data. For the abundance estimation case, we compare our proposed U-ADMM-AENet-I & II with a traditional sparse unmixing algorithm, SunSAL [8], as well as unfolding based learning algorithms MNN-AE-1 & 2 [34]. We do not compare our proposed approaches to state-of-the-art learning-based methods, such as pixel-based CNN [18], because it has been verified in [34] that the performance of MNN-AE is better than pixel-based CNN. Unless explicitly mentioned, the experiments are conducted under the default setting, where we use a training set consisting of 1000 randomly selected pixels (signatures) extracted from the synthetic HSI dataset. We also contaminate these spectral signatures with AWGN with $SNR = 15$ dB. We use U-ADMM-AENet and MNN-AE consisting of two layers (iteration blocks) only because

we have found empirically that additional layers do not result in significant performance improvements (see also [34]). We train the networks using Adam optimizer with a learning rate set to $1e-4$, a batch size set to 64, and the number of epochs set to 300.

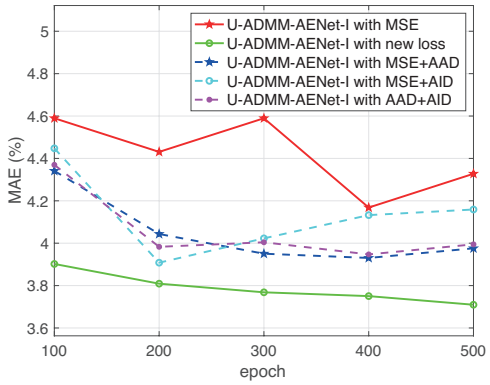
For the blind unmixing case, we compare our proposed unsupervised U-ADMM-BUNet-I & II with learning-based blind unmixing algorithms (uDAS [22], MNN-BU [34], UnDIP [27] and EGU-Net-pw [28]) and traditional nonnegative matrix factorization (NMF)-based algorithms such as matrix-vector nonnegative tensor factorization (MV-NTF) [10]. We also use the default experiment setting, where we use a training set consisting of 1000 randomly selected pixels (signatures) extracted from the synthetic HSI dataset that are further contaminated with AWGN with $SNR = 25$ dB. We use U-ADMM-BUNet and MNN-BU networks consisting of two layers (iteration blocks) followed by a linear (decoding) layer. We also train the networks using Adam optimizer with a learning rate set to $1e-4$, a batch size set to 64, and the number of epochs set to 300.

Our proposed methods use the training strategies, initialization strategies, and hyper-parameter settings reported in earlier sections. In turn, competing methods use the default hyper-parameter settings proposed in their original papers.

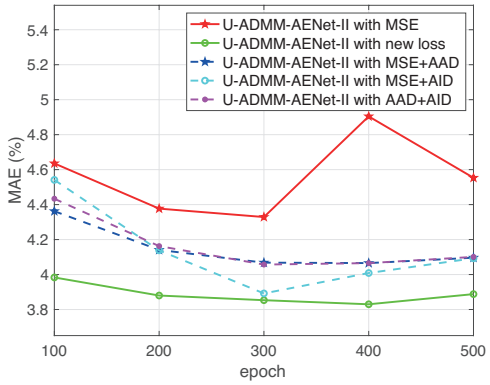
1) Effect of proposed composite loss: In this experiment, we evaluate the impact of the new proposed composite loss function in the abundance estimation case. We also compare it with other combinations of loss functions, such as MSE+AAD, MSE+AID and AAD+AID to better illustrate the improvement induced by the proposed composite loss function. We broadly use the default experimental settings, except that we set the number of training data to be 256 and the training epochs varying from 100 to 500.

We here report the MAE result in Fig. 7 to better show the superiority of the new loss. We do not report it using other metrics such as AAD because it would be an unfair comparison as those metrics are included in the proposed loss function. It can be seen that for both network structures, the new proposed composite loss function provides better and more stable performance. This is because that the MSE loss function has the implicit assumption [41] that the abundance estimated by the network follows a Gaussian distribution, which is incompatible with ASC and ANC constraints. The proposed new loss function, on the other hand, has reduced the impact of such implicit assumptions.

2) Performance vs. number of layers: We now evaluate the performance of the various approaches as a function of the number of layers. Note that both abundance estimation case and blind unmixing case show a similar trend thus we only report the result of the former to save space. We use the aforementioned default experiment setting except that we now vary the number of layers from 1 to 7. The results are shown in Fig. 8. It is clear that for an unfolding based unmixing network, the number of layers does not have a significant impact on the final performance, whereas a conventional neural network would typically benefit from a deeper structure [51]. We attribute this observation to the fact that unfolding networks architecture comes with a strong inductive bias, so that strong performance



(a) U-ADMM-AENet-I



(b) U-ADMM-AENet-II

Fig. 7. The impact of different composite loss on abundance estimation performance: MAE. (a) U-ADMM-AENet-I. (b) U-ADMM-AENet-II.

can be achieved with a small number of layers. For this reason, we choose the number of layers to be 2 for the remaining experiments.

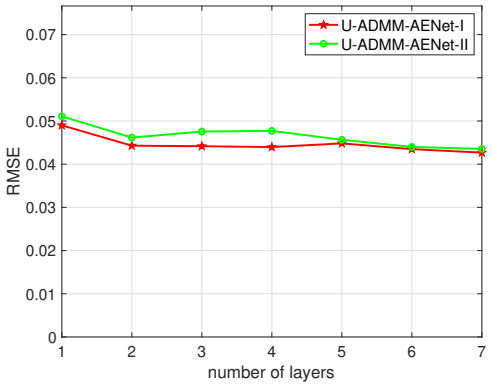
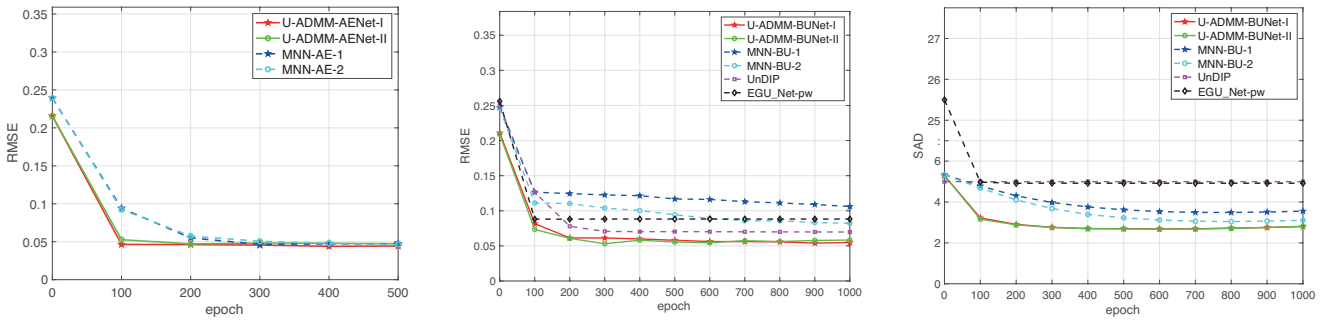


Fig. 8. The impact of number of layers on abundance estimation performance.

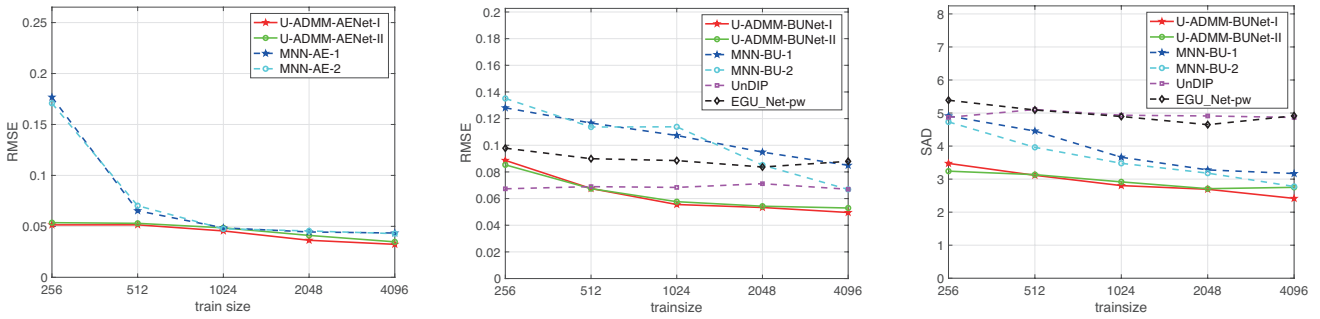
3) Performance vs. Training Epochs: We now assess the convergence performance of the various approaches as a function of the number of training epochs. For the abundance estimation case, we use the default experiment setting except for the number of epochs varying from 0 to 500. For the blind unmixing case, we set the number of epochs ranging from 0 to 1000.

We report abundance estimation performance vs. the number



(a) Abundance estimation: RMSE versus epochs. (b) Blind unmixing: abundance RMSE versus epochs. (c) Blind unmixing: endmember SAD versus epochs.

Fig. 9. The impact of training epochs on the performance. (a) Abundance estimation case: RMSE. (b) Blind unmixing case: RMSE. (c) Blind unmixing case: SAD in degree.



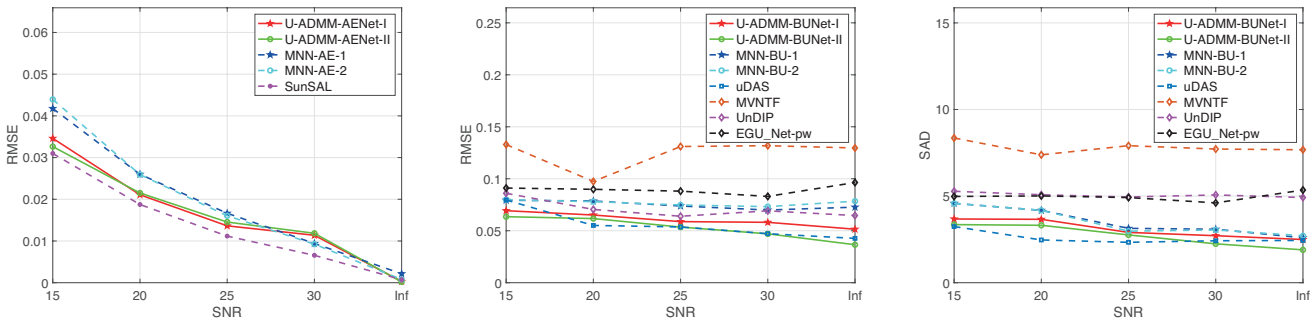
(a) Abundance estimation: RMSE versus training dataset size. (b) Blind unmixing: abundance RMSE versus training dataset size. (c) Blind unmixing: endmember SAD versus training dataset size.

Fig. 10. The impact of size of training dataset on the performance. (a) Abundance estimation case: RMSE. (b) Blind unmixing case: RMSE. (c) Blind unmixing case: SAD in degree.

of epochs in Fig. 9a for U-ADMM-AENet and MNN-AE, respectively. Note that we only show RMSE results because other metrics trends are akin to RMSE ones. We also report blind unmixing performance vs. number of epochs in Fig. 9b and Fig. 9c. It is evident that our proposed methods can achieve faster convergence than MNN-AE and MNN-BU. Specifically, in the abundance estimation case, we observe that U-ADMM-AENet converges around 100 training epochs, while MNN-AE networks only converge after 300 training epochs. In the blind unmixing case, we also observe a similar performance. Specifically, in terms of SAD, the proposed U-ADMM-BUNet achieves impressive performances at around 200 epochs while MNN-BU needs around 600 epochs. We attribute this to the fact that ADMM based solvers typically converge faster than ISTA based ones [35]. We also attribute this to the fact the weighted loss function adopted in our learning algorithms is more complex than that adopted in competing ones (see [34]), allowing to promote additional dissimilarity. On the other hand, although the state-of-art UnDIP and EGU-Net achieve a similar convergence speed, the proposed networks have better unmixing performance. In particular, in terms of SAD, the proposed network is around 2.5 while both UnDIP and EGU-Net is around 5.0. Thus we can achieve 2X better performance than the state-of-art UnDIP and EGU-Net because they rely on existing endmember extraction algorithms to provide a guidance.

4) Performance vs. Training Data: We now assess the performance of the various approaches as a function of the number of training data points. Both for abundance estimation and for blind unmixing scenarios, we use a training set consisting of a number of randomly selected pixels (signatures) ranging from 256 up to 4096 whereas the other experiment settings are in line with the default setting.

We report abundance estimation performance vs. training size along with endmember estimation performance vs. training size in Fig. 10. It is clearly apparent that our proposed methods can achieve better performance than competing ones in the presence of smaller training datasets. Specifically, for the abundance estimation case, when the size of the training dataset is very small (e.g. 256), both types of U-ADMM-AENet are superior to MNN-AE in terms of abundance estimation performance. This may be due to the fact that the network deriving from ADMM has much more residual connections in relation to networks deriving from ISTA. For the blind unmixing case with small training datasets, U-ADMM-BUNet is also superior to competing ones both in terms of abundance estimation (RMSE) and endmember estimation (SAD) performance. Naturally, the performance of most networks improves with the increase in the size of the training set.



(a) Abundance estimation: RMSE versus SNR. (b) Blind unmixing: abundance RMSE versus SNR. (c) Blind unmixing: endmember SAD versus SNR.

Fig. 11. The impact of SNR on the performance. (a) Abundance estimation case: RMSE. (b)Blind unmixing case: RMSE. (c) Blind unmixing case: SAD in degree.

Furthermore, the state-of-art algorithms such as UnDIP and EGU-Net, in general, have worse unmixing performance. For example, the SAD of both UnDIP and EGU-Net is around 5.0, while that of the proposed method is around 2.5. This is because both UnDIP and EGU-Net rely on the existing endmember extraction algorithms to provide an estimation of endmembers as a guidance.

5) Performance in Presence of Noise: It is also illustrative to gauge the robustness of the proposed approaches against noise contamination. We retain the default experimental settings except that we contaminate the spectral signatures with different noise levels leading up to SNR values ranging within the set $[15, 20, 25, 30, inf]$ dB.

Fig. 11 suggests that our proposed approaches seem to be much more immune to noise in relation to competing ones. In particular, for the abundance estimation case, with a relatively noisy setting corresponding to a $SNR = 15$ dB, the RMSE pertaining to U-ADMM-AENet is lower than that pertaining to MNN-AE. As the traditional sparse unmixing algorithm SunSAL is a direct application of ADMM, it acts as a baseline in this comparison.

For the blind unmixing case, with noisy data, the proposed U-ADMM-BUNet also achieves better RMSE (for abundance estimation) and SAD (for endmember estimation) performance in relation to MNN-BU, UnDIP and EGU-Net. The proposed methods also achieve much better RMSE and SAD performance in relation to MVNTF.

6) Run Time Comparison: In this experiment, we compare the run time of various algorithms performing unmixing over synthetic data. The experiments are conducted in a Linux server with Intel XEON GOLD 5120 CPU at 2.2GHz, 251GB RAM and TESLA V100 GPU. All algorithms are implemented using python and MATLAB. We again use the default experimental settings. The results are summarized in Table IV and V.

It is clear that the proposed methods achieve faster unmixing compared to traditional model-based unmixing methods such as SunSAL and MVNTF. This is due to the fact that machine learning-based methods only require one forward computation in the prediction/unmixing phase while traditional methods usually require many iterations. On the other hand, compared to conventional machine-learning based methods such as uDAS, UnDIP and EGU-Net, the unfolding-based networks perform faster because of their highly efficient architecture. Finally, in line with our discussion about network complexity, our proposed networks have a similar unmixing speed as the unfolding ISTA-based networks, i.e., MNN-AE & MNN-BU.

TABLE IV
RUN TIME COMPARISON: ABUNDANCE ESTIMATION

method	SunSAL	MNN-AE-1	MNN-AE-2	U-ADMM-AENet-I	U-ADMM-AENet-II
time (s)	1.44	0.14	0.14	0.21	0.25

TABLE V
RUN TIME COMPARISON: BLIND UNMIXING

method	uDAS	MV-NTF	MNN-BU-1
time (s)	11.67	66.49	0.19
method	MNN-BU-2	U-ADMM-BUNet-I	U-ADMM-BUNet-II
time (s)	0.27	0.22	0.21
method	UnDIP	EGU-Net-pw	
time (s)	26.76	0.44	

D. Experiments with Real Data

We now evaluate the performance of our approaches on real hyperspectral datasets, including Jasper Ridge and Urban.

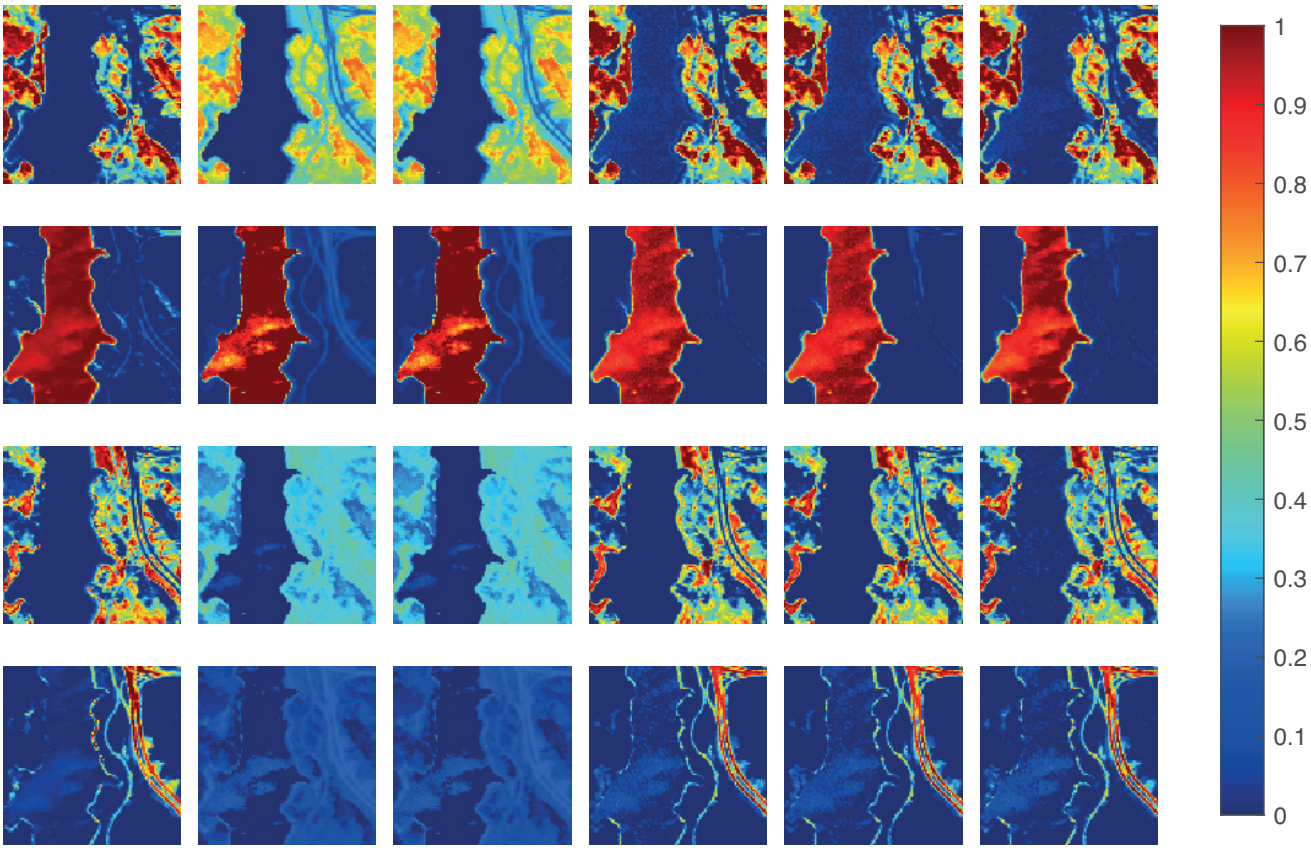
1) Abundance Estimation: We first concentrate on abundance estimation. We train the network with two iteration blocks using 256 pixels randomly chosen from the data while evaluating on the remaining pixels. The algorithms are run five times in order to report average metrics. In this case, we use the initialization strategy where the true $\boldsymbol{A}$ is known, which is provided by the real dataset. All learning-based algorithms that require such true $\boldsymbol{A}$ are also trained with this true value.

The performance of various algorithms using a number of metrics is reported in Table VI. It can be seen that overall our proposed U-ADMM-AENet achieve the best RMSE, AAD and AID performance.

The abundance maps of various algorithms on Jasper Ridge and Urban are illustrated in Fig. 12 and Fig. 13. It can also

TABLE VI
MEAN AND STANDARD DEVIATION OF ABUNDANCE RMSE, AAD (IN DEGREES), AID BY DIFFERENT ABUNDANCE ESTIMATION ALGORITHMS ON JASPER RIDGE AND URBAN. THE BEST RESULTS ARE IN BOLD.

Dataset	Metrics	SunSAL	MNN-AE-1	MNN-AE-2	U-ADMM-AENet-I	U-ADMM-AENet-II
Jasper Ridge	RMSE	0.0612±0.0001	0.1285±0.0021	0.1262±0.0008	**0.0212±0.0005**	0.0214±0.0004
	AAD	7.9068±0.0001	17.9929±0.3719	17.4182±0.1998	**2.6742±0.0660**	2.7447±0.0669
	AID	0.4564±0.0001	2.1823±0.0358	2.1215±0.0444	0.1765±0.0126	**0.1630±0.0197**
Urban	RMSE	0.1679±0.0001	0.2416±0.0007	0.2387±0.0003	0.0431±0.0019	**0.0417±0.0017**
	AAD	25.1087±0.0001	36.2268±0.1561	35.6648±0.0573	5.8625±0.2702	**5.6651±0.2415**
	AID	4.5491±0.0001	4.7990±0.0110	4.7601±0.0281	0.3625±0.0552	**0.3090±0.0491**



(a) SunSAL (b) MNN-AE-1 (c) MNN-AE-2 (d) U-ADMM-AENet-I (e) U-ADMM-AENet-II (f) Reference

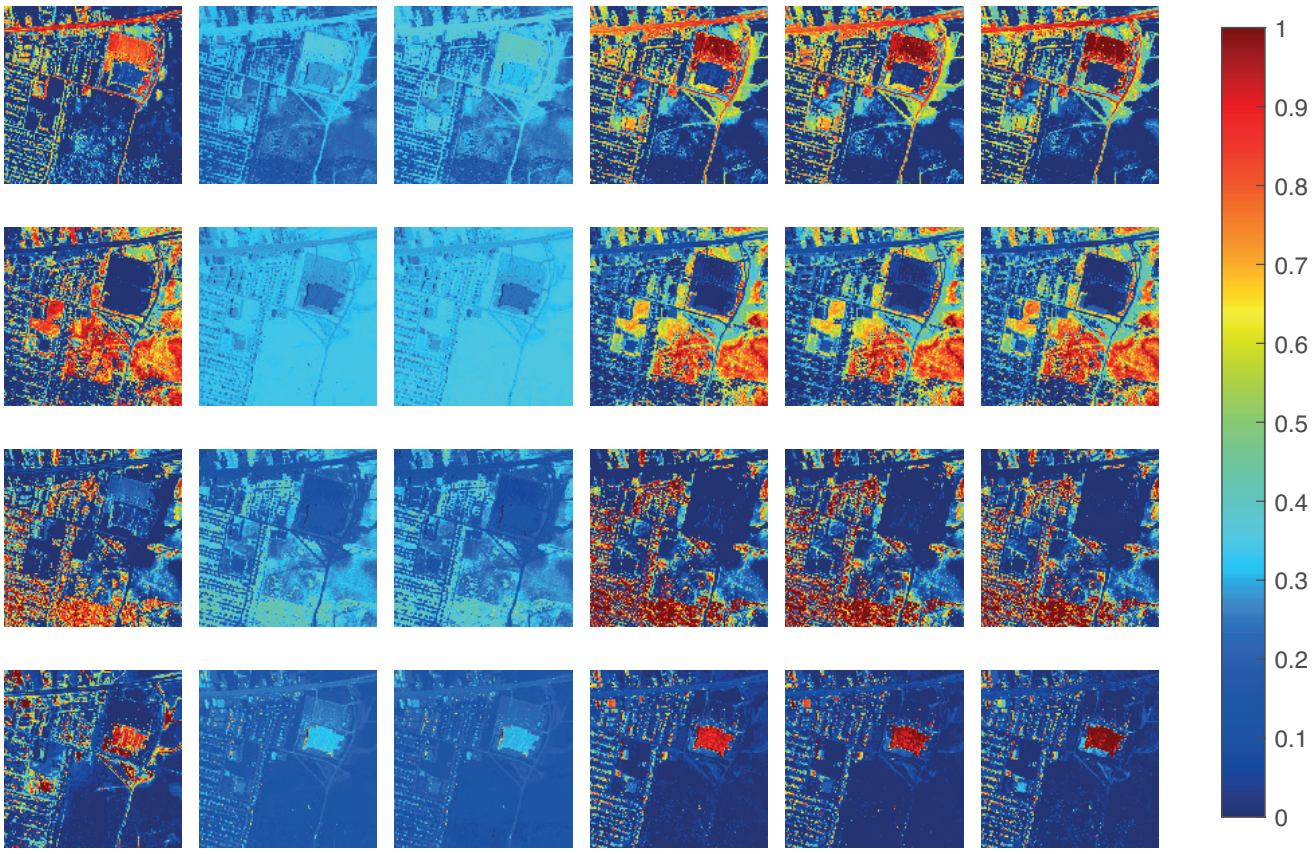
Fig. 12. Results of abundance estimation by different methods on Jasper Ridge dataset. From top to bottom: Tree, Water, Soil and Road. (a) SunSAL. (b) MNN-AE-1. (c) MNN-AE-2. (d) U-ADMM-AENet-I. (e) U-ADMM-AENet-II. (f) Reference.

be seen that overall our proposed U-ADMM-AENet lead to abundance maps closer to the ground truth in comparison to competing algorithms.

2) Blind Unmixing: In this case, we use the initialization strategy where the true signatures $\boldsymbol{A}$ are unknown.

We now concentrate on blind unmixing. We train and evaluate the network with two iteration blocks on the whole pixels. The algorithms are run five times in order to report the performance mean and standard deviation.

Table VII shows the SAD mean and standard deviation associated with various algorithms. For the Jasper Ridge dataset, it can be seen that the performance of different algorithms depends on the specific endmember. In particular, MNN-BU-2 offers the best spectral signature estimates for Road. However, it can also be seen that the performance mean and standard deviation of our proposed approaches (i.e. U-ADMM-BU-Net-II) tend to be better on average than that of competitors. For the Urban dataset, we can also observe that the performance of the different algorithms depends on the specific endmember; likewise, we can observe that the mean and standard deviation



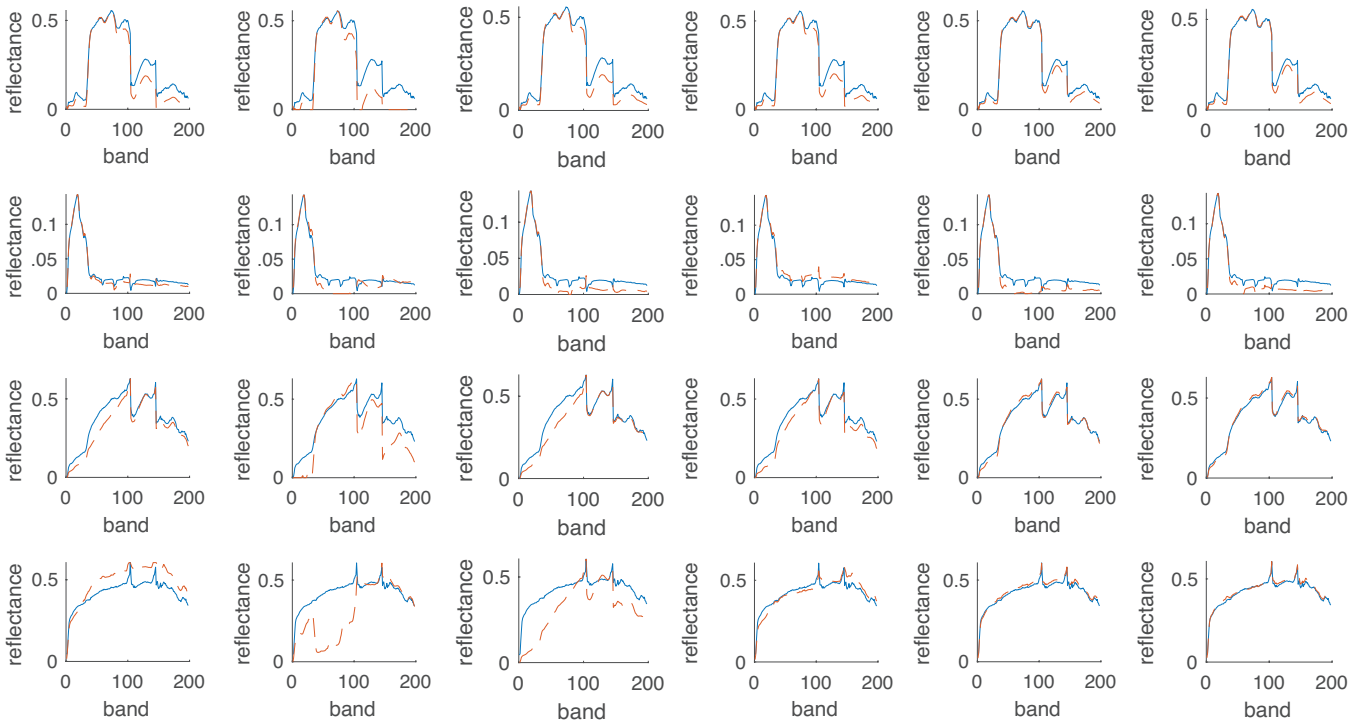
(a) SunSAL (b) MNN-AE-1 (c) MNN-AE-2 (d) U-ADMM-AENet-I (e) U-ADMM-AENet-II (f) Reference

Fig. 13. Results of abundance estimation by different methods on Urban dataset. From top to bottom: Asphalt, Grass, Tree, and roof. (a) SunSAL. (b) MNN-AE-1. (c) MNN-AE-2. (d) U-ADMM-AENet-I. (e) U-ADMM-AENet-II. (f) Reference.

TABLE VII
MEAN AND STANDARD DEVIATION OF ENDMEMBER SAD(IN DEGREES) BY DIFFERENT BLIND UNMIXING ALGORITHMS ON JASPER RIDGE AND URBAN. THE BEST RESULTS ARE IN BOLD.

Dataset	Endmember	uDAS	MV-NTF	UnDIP	EGU-Net-pw	MNN-BU-2	U-ADMM-BUNet-II
Jasper Ridge	Tree	10.0032±0.8436	15.3062±4.3640	8.5545±0.0000	8.1393±0.0006	5.0530±0.0017	**5.0308±0.0037**
	Water	7.6166±0.5672	15.1020±0.7648	14.4877±0.0000	**6.5646±0.0019**	16.4420±0.0276	13.4348±0.0931
	Soil	6.6608±0.1950	19.0967±13.5871	6.5558±0.0000	5.407±0.0020	2.2163±0.0037	**1.2418±0.0049**
	Road	3.6029±0.1044	27.3426±2.7321	15.7991±0.0000	4.3562±0.0013	**0.9679±0.0029**	1.0425±0.0058
	Mean	6.9709±0.2659	19.2119±1.9248	11.3493±0.0000	6.1168±0.0015	6.1698±0.0057	**5.1875±0.0240**
Urban	Asphalt	12.3003±0.1118	13.5129±1.0604	53.1045±0.0000	**5.9649±0.0027**	12.3451±0.2343	8.1085±0.3008
	Grass	63.6797±3.8035	19.3200±3.9887	61.1254±0.0000	17.599±0.0009	11.9362±0.0504	**9.0528±0.4471**
	Tree	**7.6825±1.2857**	7.8657± 0.3075	64.4102±0.0000	84.6801±0.0103	7.9079±0.0376	8.8417±0.2107
	Roof	14.4911±3.0348	27.7510±2.6030	32.0613±0.0000	20.6919±0.0022	**9.7467±0.0549**	11.6299±0.1128
	Mean	24.5384±2.0074	17.1124±1.4393	52.6754±0.0000	32.2340±0.0041	10.4840±0.0690	**9.4082±0.1626**

of the performance metric tends to be better on average for our proposed approaches in relation to competing ones. In particular, the endmember extraction algorithm used in UnDIP fails to properly extract the endmembers, as a result, it would also fail to estimate the abundance. Figs. 14 and 15, which illustrate the recovered endmember signatures along with reference signatures using various algorithms for the Jasper Ridge and Urban datasets, respectively, also corroborate



(a) uDAS (b) MV-NTF (c) UnDIP (d) EGU-Net-pw (e) MNN-BU-2 (f) U-ADMM-BUNet-II

Fig. 14. Estimated endmember signatures of blind unmixing by different methods on Jasper Ridge dataset. Solid line indicates the true value, while dot line indicates the scaled estimated value. From top to bottom: Tree, Water, Soil, and Road. (a) uDAS. (b) MV-NTF. (c) UnDIP. (d) EGU-Net-pw. (e) MNN-BU-2. (f) U-ADMM-BUNet-II.

TABLE VIII
MEAN AND STANDARD DEVIATION OF ABUNDANCE RMSE, AAD (IN DEGREES), AID BY DIFFERENT BLIND UNMIXING ALGORITHMS ON JASPER RIDGE AND URBAN. THE BEST RESULTS ARE IN BOLD.

Dataset	Metrics	uDAS	MV-NTF	UnDIP	EGU-Net-pw	MNN-BU-2	U-ADMM-BUNet-II
Jasper Ridge	RMSE	0.1228±0.0067	0.2114±0.0103	0.1748±0.0252	0.1169±0.0073	0.0773±0.0009	**0.0720±0.0012**
	AAD	15.4408±0.9854	29.5696±1.7837	25.3249±4.4570	15.3437±0.0019	9.8228±0.1362	**9.1563±0.1625**
	AID	0.9564±0.0932	2.0981±0.1082	3.6022±0.6558	0.9444±0.0040	0.4974±0.0088	**0.4604±0.0153**
Urban	RMSE	0.2919±0.0111	0.2617±0.0056	0.3112±0.0069	0.2571±0.0031	0.2203±0.0003	**0.2196±0.0013**
	AAD	44.9756±2.6502	38.8271±1.3656	51.2773±1.5093	37.1957±0.0009	33.4093±0.0276	**32.5843±0.2192**
	AID	2.4992±0.2185	**1.9328±0.0499**	6.8164±0.1329	2.2176±0.0017	2.2413±0.0018	2.2301±0.0168

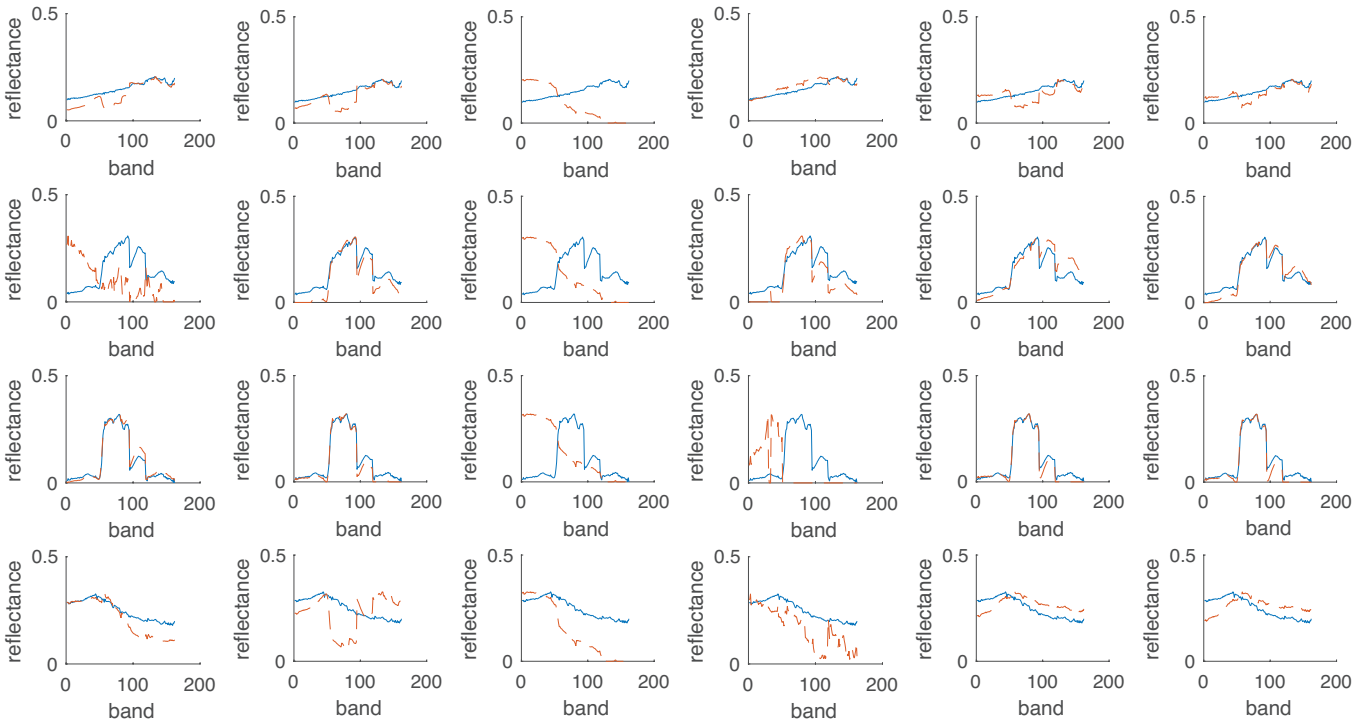
these observations.

Finally, the abundance maps associated with the various algorithms on the Jasper Ridge and Urban Datasets are shown in Figs. 16 and 17. We also report the various performance metrics in Table VIII. It is clear that our proposed approaches lead to abundance maps closer to the reference than competing ones. It should be noted that the RMSE of U-ADMM-AENet is 0.02 on Jasper Ridge and 0.04 on Urban, while the RMSE of U-ADMM-BUNet is 0.07 on Jasper Ridge and 0.22 on Urban. This is because U-ADMM-AENet assumes the access to both the spectral reflectances and the corresponding abundances in the training dataset, while U-ADMM-BUNet only assumes the access to the spectral reflectances.

VII. CONCLUSION

We have proposed new hyperspectral unmixing networks deriving from unfolding procedures. In particular, building upon a traditional constrained sparse regression approach to the linear unmixing challenge, we have shown how ADMM leads to a neural network architecture consisting of various interpretable learning modules with a counterpart in the machine learning literature.

Our proposed approach combines the advantages of model-based and learning-based unmixing methods. It leads to architectures that can be trained both in a supervised, or unsupervised manner using newly proposed weighted loss functions. It also leads to neural network architectures that possess very



(a) uDAS (b) MV-NTF (c) UnDIP (d) EGU-Net-pw (e) MNN-BU-2 (f) U-ADMM-BUNet-II

Fig. 15. Estimated endmember signatures of blind unmixing by different methods on Urban dataset. Solid line indicates the true value, while dot line indicates the scaled estimated value. From top to bottom: Asphalt, Grass, Tree, and roof. (a) uDAS. (b) MV-NTF. (c) UnDIP. (d) EGU-Net-pw. (e) MNN-BU-2. (f) U-ADMM-BUNet-II.

rich structures, including skipping connections and residual blocks, which offer superior performance in image analysis and processing tasks.

Our approach also offers state-of-the-art performance in comparison to existing approaches. Of particular relevance, extensive numerical results showcase that our approach outperforms state-of-the-art ones such as ISTA based network MNN-AE and MNN-BU, uDAS, UnDIP, EGU-Net and MV-NTF, in terms of unmixing quality, convergence speed, and training data needs both on real and synthetic HSI data. The improved performance potentially comes from four aspects: First of all, the ADMM solver can lead to sufficient accuracy with fast convergence compared to ISTA solver [11], [35]. As a result, the proposed network unfolded from ADMM could potentially inherit such advantages. Secondly, it has been shown in [31], [32], [33] that the network unfolded from an iterative algorithm can achieve better performance than the original iterative algorithm. Thirdly, as we discussed in Section V, the proposed network has less learnable parameters and richer skip connections, which can improve neural network performance in various tasks [41], [44], [45], [47], [46]. Finally, as we show in Section VI, the proposed loss function can also improve the network performance.

REFERENCES

[1] G. Shaw and D. Manolakis, "Signal processing for hyperspectral image exploitation," *IEEE Signal Process. Mag.*, vol. 19, no. 1, pp. 12–16, 2002.

[2] J. M. Bioucas-Dias, A. Plaza, G. Camps-Valls, P. Scheunders, N. Nasrabadi, and J. Chanussot, "Hyperspectral remote sensing data analysis and future challenges," *IEEE Geosci. Remote Sens. Mag.*, vol. 1, no. 2, pp. 6–36, 2013.

[3] N. Keshava and J. Mustard, "Spectral unmixing," *IEEE Signal Process. Mag.*, vol. 19, no. 1, pp. 44–57, 2002.

[4] G. Camps-Valls, D. Tuia, L. Gómez-Chova, S. Jiménez, and J. Malo, *Remote sensing image processing*. Morgan & Claypool Publishers, 2011.

[5] J. Nascimento and J. Dias, "Vertex component analysis: a fast algorithm to unmix hyperspectral data," *IEEE Trans. Geosci. Remote Sens.*, vol. 43, no. 4, pp. 898–910, 2005.

[6] 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.

[7] D. 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.

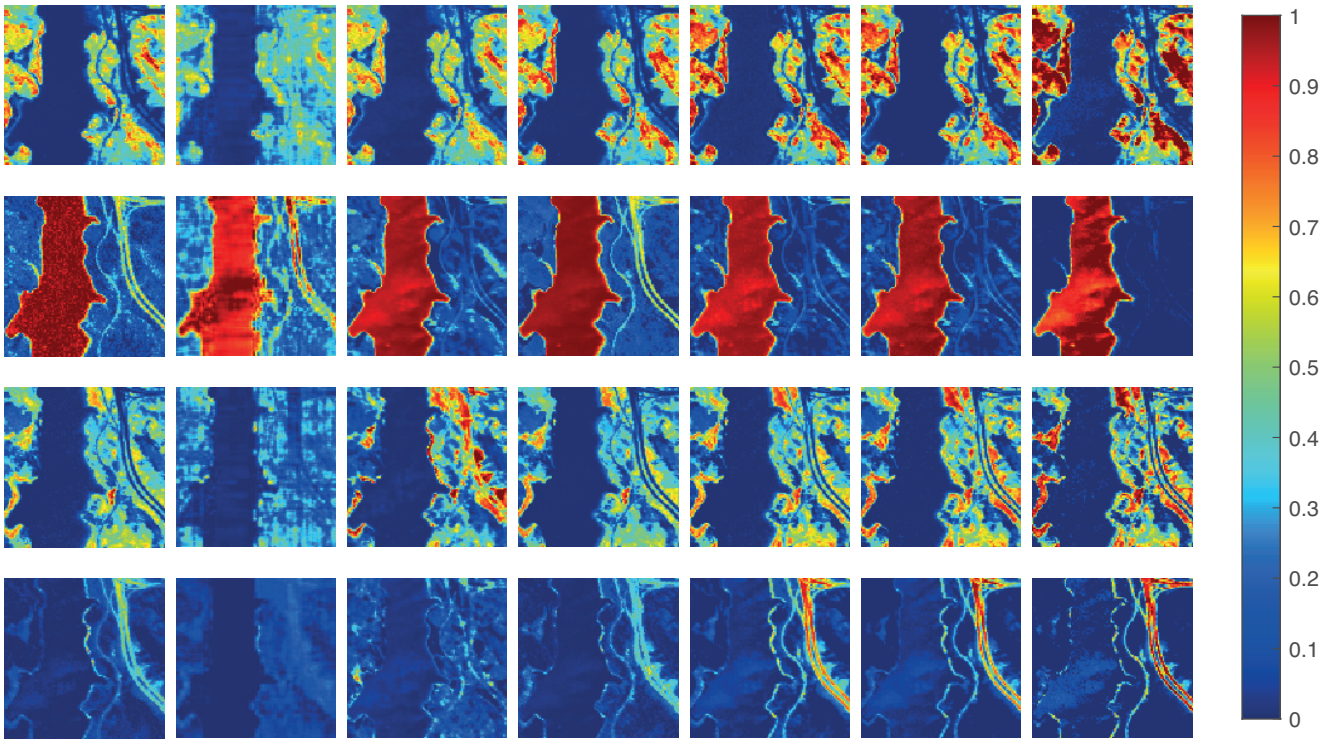
[8] J. M. Bioucas-Dias and M. A. Figueiredo, "Alternating direction algorithms for constrained sparse regression: Application to hyperspectral unmixing," in *Proc. 2nd IEEE GRSS Workshop on Hyperspectral Image and Signal Processing: Evol. Remote Sens. (WHISPERS)*, Reykjavik, Iceland, Jun. 2010, pp. 1–4.

[9] S. Jia and Y. Qian, "Constrained nonnegative matrix factorization for hyperspectral unmixing," *IEEE Trans. Geosci. Remote Sens.*, vol. 47, no. 1, pp. 161–173, 2009.

[10] 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.

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

[12] M. Zhao, M. Wang, J. Chen, and S. Rahardja, "Hyperspectral unmixing



(a) uDAS (b) MV-NTF (c) UnDIP (d) EGU-Net-pw (e) MNN-BU-2 (f) U-ADMM-BUNet-II (g) Reference

Fig. 16. Estimated abundance map of blind unmixing by different methods on Jasper Ridge dataset. From top to bottom: Tree, Water, Soil, and Road. (a) uDAS. (b) MV-NTF. (c) UnDIP. (d) EGU-Net-pw. (e) MNN-BU-2. (f) U-ADMM-BUNet-II. (g) Reference. It is clear that the unfolding networks MNN-BU and U-ADMM-BUNet outperform the others. The numerical comparison in Table VIII shows the proposed network U-ADMM-BUNet performs better than MNN-BU.

for additive nonlinear models with a 3-d-cnn autoencoder network," *IEEE Trans. Geosci. Remote Sens.*, pp. 1–15, 2021.

[13] Q. Jin, Y. Ma, F. Fan, J. Huang, X. Mei, and J. Ma, "Adversarial autoencoder network for hyperspectral unmixing," *IEEE Trans. Neural Netw. Learn. Syst.*, pp. 1–15, 2021.

[14] 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.

[15] F. Xiong, J. Zhou, S. Tao, J. Lu, and Y. Qian, "Snmf-net: Learning a deep alternating neural network for hyperspectral unmixing," *IEEE Trans. Geosci. Remote Sens.*, pp. 1–16, 2021.

[16] K. T. Shahid and I. D. Schizas, "Unsupervised hyperspectral unmixing via nonlinear autoencoders," *IEEE Trans. Geosci. Remote Sens.*, pp. 1–13, 2021.

[17] G. A. Licciardi and F. Del Frate, "Pixel unmixing in hyperspectral data by means of neural networks," *IEEE Trans. Geosci. Remote Sens.*, vol. 49, no. 11, pp. 4163–4172, 2011.

[18] 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.

[19] Q. Qian, F. Xiong, and J. Zhou, "Deep unfolded iterative shrinkage-thresholding model for hyperspectral unmixing," in *Proc. IEEE Int. Geosci. Remote Sens. Symp. (IGARSS)*, Yokohama, Japan, Jul. 2019, pp. 2151–2154.

[20] F. Palsson, J. Sigurdsson, J. R. Sveinsson, and M. O. Ulfarsson, "Neural network hyperspectral unmixing with spectral information divergence objective," in *Proc. IEEE Int. Geosci. Remote Sens. Symp. (IGARSS)*, Fort Worth, TX, USA, Jul. 2017, pp. 755–758.

[21] 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.

[22] 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.

[23] S. Ozkan and G. B. Akar, "Improved deep spectral convolution network for hyperspectral unmixing with multinomial mixture kernel and endmember uncertainty," *arXiv preprint arXiv:1808.01104*, 2018.

[24] 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.

[25] 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.

[26] R. A. Borsoi, T. Imbiriba, and J. C. M. Bermudez, "Deep generative endmember modeling: An application to unsupervised spectral unmixing," *IEEE Trans. Comput. Imaging*, vol. 6, pp. 374–384, 2020.

[27] B. Rasti, B. Koirala, P. Scheunders, and P. Ghamisi, "Undip: Hyperspectral unmixing using deep image prior," *IEEE Trans. Geosci. Remote Sens.*, pp. 1–15, 2021.

[28] D. Hong, L. Gao, J. Yao, N. Yokoya, J. Chanussot, U. Heiden, and B. Zhang, "Endmember-guided unmixing network (egu-net): A general deep learning framework for self-supervised hyperspectral unmixing," *IEEE Trans. Neural Netw. Learn. Syst.*, pp. 1–14, 2021.

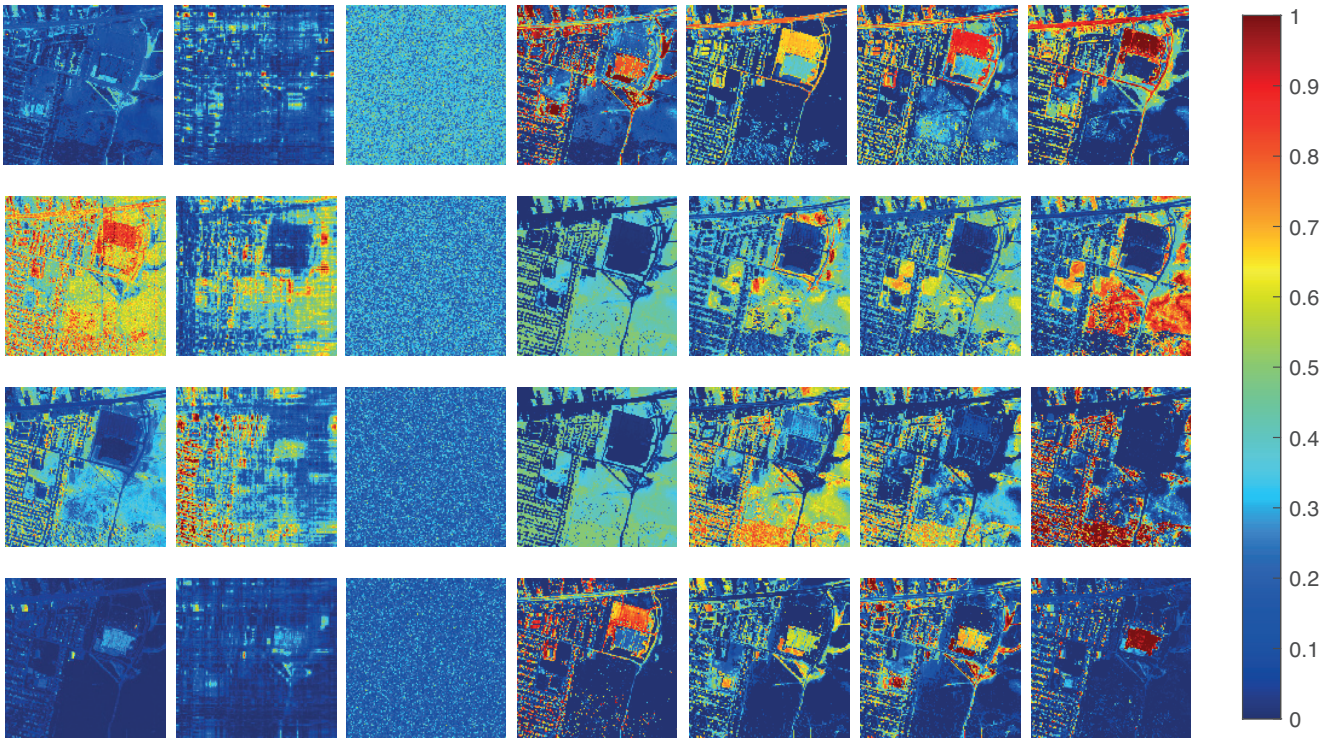
[29] V. Monga, Y. Li, and Y. C. Eldar, "Algorithm unrolling: Interpretable, efficient deep learning for signal and image processing," *IEEE Signal Process. Mag.*, vol. 38, no. 2, pp. 18–44, 2021.

[30] K. Gregor and Y. LeCun, "Learning fast approximations of sparse coding," in *Proc. 27th Int. Conf. Mach. Learn. (ICML)*, Haifa, Israel, Jun. 2010, pp. 399–406.

[31] X. Chen, J. Liu, Z. Wang, and W. Yin, "Theoretical linear convergence of unfolded ista and its practical weights and thresholds," *arXiv preprint arXiv:1808.10038*, 2018.

[32] R. Giryes, Y. C. Eldar, A. M. Bronstein, and G. Sapiro, "Tradeoffs between convergence speed and reconstruction accuracy in inverse problems," *IEEE Trans. Signal Process.*, vol. 66, no. 7, pp. 1676–1690, 2018.

[33] T. Moreau and J. Bruna, "Understanding trainable sparse coding via matrix factorization," *arXiv preprint arXiv:1609.00285*, 2016.



(a) uDAS (b) MV-NTF (c) UnDIP (d) EGU-Net-pw (e) MNN-BU-2 (f) U-ADMM-BUNet-II (g) Reference

Fig. 17. Estimated abundance map of blind unmixing by different methods on the Urban dataset. From top to bottom: Asphalt, Grass, Tree, and roof. (a) uDAS. (b) MV-NTF. (c) UnDIP. (d) EGU-Net-pw. (e) MNN-BU-2. (f) U-ADMM-BUNet-II. (g) Reference. It is clear that the unfolding network MNN-BU and U-ADMM-BUNet outperform the others. Particularly, the proposed U-ADMM-BUNet achieves best in terms of tree. The numerical comparison in Table VIII shows the superiority of the proposed network.

[34] 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.

[35] S. Tao, D. Boley, and S. Zhang, "Convergence of common proximal methods for l1-regularized least squares," in *24th Int. Joint Conf. Artif. Intell. (IJCAI)*, Buenos Aires, Argentina, Jul. 2015, p. 3849–3855.

[36] "Usgs library." [Online]. Available: https://www.usgs.gov/labs/spec-lab

[37] I. Goodfellow, Y. Bengio, and A. Courville, *Deep learning*. MIT press, 2016.

[38] R. Rajabi and H. Ghassemian, "Spectral unmixing of hyperspectral imagery using multilayer nmf," *IEEE Geosci. Remote Sens. Lett.*, vol. 12, no. 1, pp. 38–42, 2015.

[39] 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.

[40] X.-R. Feng, H.-C. 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.

[41] C. Bishop, *Pattern Recognition and Machine Learning*, ser. Information Science and Statistics. Springer New York, 2016.

[42] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

[43] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recog. (CVPR)*, Las Vegas, NV, USA, Jun. 2016, pp. 770–778.

[44] B. Lim, S. Son, H. Kim, S. Nah, and K. M. Lee, "Enhanced deep residual networks for single image super-resolution," in *Proc. IEEE Conf. Comput. Vis. Pattern Recog. Workshops (CVPRW)*, Honolulu, HI, USA, Jul. 2017, pp. 1132–1140.

[45] K. Zhang, W. Zuo, Y. Chen, D. Meng, and L. Zhang, "Beyond a gaussian denoiser: Residual learning of deep cnn for image denoising," *IEEE Trans. Image Process.*, vol. 26, no. 7, pp. 3142–3155, 2017.

[46] J. Kim, J. K. Lee, and K. M. Lee, "Accurate image super-resolution using very deep convolutional networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recog. (CVPR)*, Las Vegas, NV, USA, Jun. 2016, pp. 1646–1654.

[47] W.-S. Lai, J.-B. Huang, N. Ahuja, and M.-H. Yang, "Deep laplacian pyramid networks for fast and accurate super-resolution," in *Proc. IEEE Conf. Comput. Vis. Pattern Recog. (CVPR)*, Honolulu, HI, USA, Jul. 2017, pp. 5835–5843.

[48] F. Zhu, Y. Wang, S. Xiang, B. Fan, and C. Pan, "Structured sparse method for hyperspectral unmixing," *ISPRS J. Photogramm. Remote Sens.*, vol. 88, pp. 101–118, 2014.

[49] 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.

[50] F. Zhu, Y. Wang, B. Fan, S. Xiang, G. Meng, and C. Pan, "Spectral unmixing via data-guided sparsity," *IEEE Trans. Image Process.*, vol. 23, no. 12, pp. 5412–5427, 2014.

[51] Z. Lin, Y. Chen, X. Zhao, and G. Wang, "Spectral-spatial classification of hyperspectral image using autoencoders," in *Proc. IEEE 9th Int. Conf. Inf. Commun. Signal Process. (ICICS)*, Tainan, Taiwan, China, Dec. 2013, pp. 1–5.

Chao Zhou received the B.S. degree in Communication Engineering from Wuhan University of Technology (WHUT), China, in 2015, and the M.S. degree in Communication and Information Systems Engineering from University of Electronic Science and Technology of China (UESTC), China, in 2018. Chao is working toward the Ph.D. degree with the Department of Electronic and Electrical Engineering, University College London (UCL), London, UK. His research interest includes machine learning, image processing, hyperspectral image processing and communication systems.

Miguel R. D. Rodrigues (Senior Member, IEEE) received the Licenciatura degree in electrical and computer engineering from the University of Porto, Porto, Portugal, and the Ph.D. degree in electronic and electrical engineering from the University College London (UCL), London, U.K. He is currently a Professor of Information Theory and Processing, UCL, and a Turing Fellow with the Alan Turing Institute - the UK National Institute of Data Science and Artificial Intelligence. His research interests include the general areas of information theory, information processing, and machine learning. His work has led to more than 200 articles in leading journals and conferences in the field, a book on Information-Theoretic Methods in Data Science (Cambridge University Press), and the IEEE Communications and Information Theory Societies Joint Paper Award 2011. He is an Associate Editor for the IEEE Transactions on Information Theory, and the IEEE Open Journal of the Communications Society. He was an Associate Editor for the IEEE Communications Letters, and the Lead Guest Editor of the special issue on "Information-Theoretic Methods in Data Acquisition, Analysis, and Processing" of the IEEE Journal on Selected Topics in Signal Processing. He was the Co-Chair of the Technical Programme Committee of the IEEE Information Theory Workshop 2016, Cambridge, U.K. He is a member of the IEEE Signal Processing Society Technical Committee on "Signal Processing Theory and Methods", and the EURASIP SAT on Signal and Data Analytics for Machine Learning (SiG-DML).