Hyperspectral Blind Unmixing using a Double Deep Image Prior

Chao Zhou, *Student Member, IEEE*, and Miguel R.D. Rodrigues, *Senior Member, IEEE*

***Abstract*—With the rise of machine learning, hyperspectral image (HSI) unmixing problems have been tackled using learning-based methods. However, physically meaningful unmixing results are not guaranteed without proper guidance. In this work, we propose an unsupervised framework inspired by Deep-Image-Prior (DIP) that can be used for both linear and nonlinear blind unmixing models. The framework consists of three modules: (1) an Endmember estimation module using DIP (EDIP), (2) an Abundance estimation module using DIP (ADIP), and (3) a Mixing module (MM). EDIP and ADIP modules generate endmembers and abundances, respectively, while MM produces a reconstruction of the HSI observations based on the postulated unmixing model. We introduce a composite loss function that applies to both linear and nonlinear unmixing models to generate meaningful unmixing results. Additionally, we propose an adaptive loss weight strategy for better unmixing results in nonlinear mixing scenarios. The proposed methods outperform state-of-art unmixing algorithms in extensive experiments conducted on both synthetic and real datasets.**

***Index Terms*—Hyperspectral image (HSI), linear unmixing, nonlinear unmixing, convolutional neural networks (CNN), Deep Image Prior (DIP)**

I. Introduction

In hyperspectral imaging (HSI) technology, sensors are able to capture the spectral reflectance of every pixel in a scene across hundreds or thousands of spectral channels [1]. This wealth of spectral information enables more accurate material identification compared to RGB imaging [2]. However, the observed reflectance is usually a mixture of the spectral signatures of the materials present in the scene due to the heterogeneity of the scene [3]. Consequently, there is a need for methods that can quantitatively decompose, or unmix, the captured spectral signature into its spectral components, also referred to as "endmembers," and their corresponding proportions within the mixture, referred to as "abundances" [4], [5].

The linear mixture model (LMM), shown in Fig. 1a, ideally models observed HSI signatures as a linear combination of endmembers' signatures weighted by their corresponding abundances [1]. However, when there are nonlinear effects such as multiple scattering, LMM is no longer applicable, because the signatures captured by sensors result from interactions with various materials at different levels/layers [3].

To address this nonlinearity, a nonlinear mixing model (NLMM) is proposed [3]. Two popular NLMMs are intimate mixture and multilayered mixture, illustrated in Fig. 1b and Fig. 1c, respectively. In intimate mixture, different materials are close to each other, while in multilayered mixture, interactions with different materials occur in different layers. Under both LMM and NLMM, the HSI blind unmixing problem involves two tasks [3]: (a) *endmember estimation* and (b) *abundance estimation*.

Generally, endmember estimation methods are based on geometrical approaches, assuming that the data is embedded in a simplex whose vertices correspond to the endmembers. Two widely used methods that fall under this category are vertex component analysis (VCA) [6] and simplex volume maximization (SiVM) [7]. Conversely, most abundance estimation methods in the literature are based on LMM [1]. For example, when the endmembers have been estimated by endmember estimation methods, the linear unmixing problem can be reduced to a least square problem, which can be solved using a fully constrained least squares (FCLS) [8] solver. If the endmembers are known in the form of a rich spectral library, the abundance estimation problem can be formulated as a linear sparse regression (SR) problem [9], which has been tackled by methods such as sparse unmixing by variable splitting and augmented Lagrangian (SunSAL) [9] and collaborative SUnSAL (CLSunSAL) [10]. There are also abundance estimation methods that take into account NLMM, such as the hierarchical Bayesian algorithm [11], where the unknown model parameters are estimated using the Metropolis-within-Gibbs sampling algorithm. Recently, a robust sparse unmixing (RSU) method with $\ell_{2,1}$-norm based loss function [12] has been proposed by considering the nonlinear terms as outliers. RSU is solved by the alternative direction method of multipliers (ADMM) [13].

Some methods perform endmember estimation and abundance estimation simultaneously, known as blind unmixing (BU). Nonnegative Matrix Factorization (NMF) [14]–[16] is an example of such a method, which decomposes a nonnegative observations matrix into two nonnegative matrices, where the endmember signature matrix and abundance matrix are interpreted as the endmembers and abundances, respectively, in the context of HSI. Another widely used BU method is Entropic Descent Archetypal Analysis (EDAA) [17], which is based on the concept that HSI data is generated by a linear combination of a small number of archetypes, representing the extreme points of the HSI data, and these archetypes are interpreted as endmembers. Some BU algorithms inspired by Nonnegative Blind Source Separation (nBSS) techniques [18]–[21] have been proposed by incorporating various constraints. For example, HiSun [22] introduces the John ellipsoid (JE) in nBSS to tackle ill-conditioned BU problems. However, these traditional approaches can be computationally complex.

This paper was produced by the IEEE Publication Technology Group. They are in Piscataway, NJ.

Manuscript received April 19, 2021; revised August 16, 2021

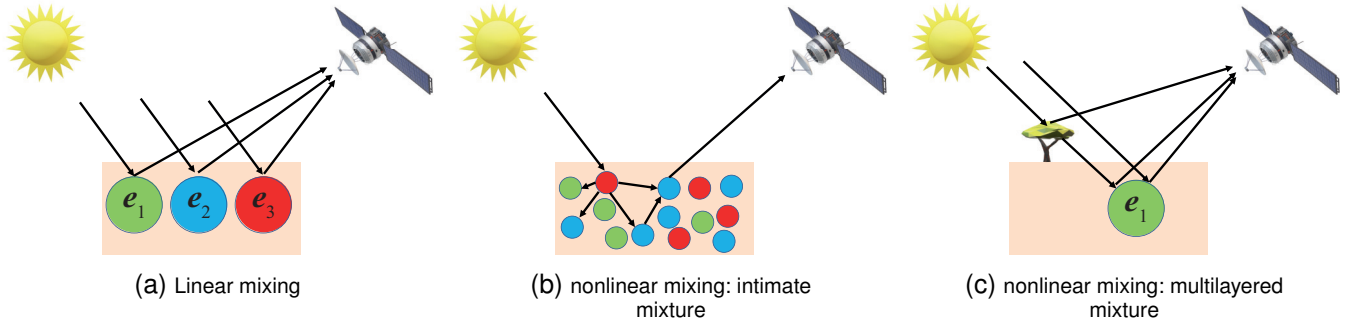


Fig. 1. Hyperspectral linear and nonlinear mixing scenarios. (a) Linear mixing. The image pixel is composed of 3 endmembers, $\mathbf{e}_1, \mathbf{e}_2, \mathbf{e}_3$. (b) Nonlinear mixing: intimate mixture. The image pixel is composed of a microscopic mixture of 3 endmembers, $\mathbf{e}_1, \mathbf{e}_2, \mathbf{e}_3$. (c) Nonlinear mixing: multilayered mixture. The image pixel is composed of 2 endmembers, tree and $\mathbf{e}_1$. Both individual reflection and interacted reflection reach the sensor.

HyperCSI [23] proposes a fast BU algorithm by exploiting the fact that the simplest simplex of N vertices can be defined by N associated hyperplanes.

More recently, machine learning techniques, particularly neural networks, have been utilized to tackle HSI unmixing tasks, resulting in numerous learning-based approaches [24]–[27]. These methods can be broadly classified into two categories: supervised and unsupervised. Supervised methods are trained using pairs of HSI reflectances and their corresponding abundances [27], [28], which enables the models to map HSI spectra to corresponding abundances. In contrast, unsupervised methods learn a function to estimate endmembers and abundances from only the HSI reflectances, without knowledge of true abundances. These blind unmixing methods [29]–[33] utilize an autoencoder network structure with a linear decoder to reconstruct HSI spectra. The bottleneck of the autoencoder provides the abundance estimation, while the weights of the linear decoder give the endmember estimation. However, these methods are only effective in solving the linear blind unmixing problem, and cannot be applied to nonlinear blind unmixing problems [24].

Recently, [24] proposed a deep autoencoder to address the nonlinear blind unmixing problem that involves an additive nonlinear mixture component. Similarly, [26] introduced a novel nonlinear autoencoder structure by incorporating a cross-product layer to account for nonlinear mixing mechanisms. In contrast, EGU-Net [2] models the nonlinear mixing of materials as spectral variability and solves the unmixing problem using a two-stream deep network that learns an additional network from the (nearly) pure endmembers extracted from HSI data via existing endmember extraction methods.

However, deep learning-based methods may produce unmixing results that lack physical meaning without appropriate guidance [2]. To address this issue, UnDIP [34] has proposed using the simplex volume maximization algorithm to extract endmembers, which are then utilized as guidance to train an abundance estimation network using a deep image prior (DIP).

Despite these developments, the quality of guidance remains a bottleneck for unmixing networks trained with guidance, as we demonstrate later. Furthermore, most learning algorithms are unable to generalize from linear to nonlinear unmixing problems, as they rely on the autoencoder structure to solve either linear or nonlinear problems. To address these issues, we propose a novel guidance-based framework that utilizes double deep image prior techniques, which can overcome the performance limitations of existing guidance-based methods. Furthermore, the proposed method does not rely on the popular autoencoder structure and can be applied to both linear and nonlinear blind unmixing problems. We refer to our method as BUDDIP, and it offers the following contributions:

1) A novel framework, BUDDIP, is proposed that builds on the deep image prior (DIP) technique [35]. Unlike the commonly used autoencoder structure, BUDDIP consists of three modules: an endmember estimation DIP (EDIP) module, an abundance estimation DIP (ADIP) module, and a mixing module (MM). BUDDIP has the flexibility to solve both linear and nonlinear blind unmixing problems.
2) Instead of using random noise as input like other DIP based methods [34], [35], the proposed framework takes meaningful input by exploiting existing unmixing methods such as SiVM. A novel DIP network structure that is more efficient than the common one used in DIP based methods [34] is designed based on this input strategy.
3) Inspired by training guidance, we propose a new composite loss function that can be applied to both linear and nonlinear blind unmixing cases. From the perspective of unmixing, the proposed loss function ensures that the network produces physically meaningful unmixing results and yields better endmember and abundance estimation than the guidance. From the perspective of network training, the proposed loss terms can be viewed as regularizations that alleviate overfitting problems in DIP techniques.
4) In the nonlinear blind unmixing case, due to the complexity of nonlinear unmixing, we also propose an adaptive loss weight strategy to yield better unmixing results.
5) Extensive numerical experiments on both synthetic and real datasets show that the proposed methods outperform state-of-the-art methods such as EGU-Net [2],

UnDIP [34] and others [14], [36], [37], under both linear and nonlinear unmixing cases.

The paper is structured as follows: Section II introduces related works. Section III describes the construction of each module of the proposed networks and how to train the apparatus in an entirely unsupervised manner. Section IV presents extensive experimental comparisons with competing methods. Finally, Section V concludes the paper.

II. Related Work

We now overview some related works.

A. Linear Mixing Model (LMM)

The linear mixing model (LMM) is one of the most popular unmixing models in hyperspectral unmixing literature, which assumes that, for each pixel, the reflectance spectrum is a linear combination of the spectrum of the endmembers weighted by the corresponding abundances [1], [4]. This model can be described as follows:

$$\mathbf{Y} = \mathbf{E}\mathbf{A} + \mathbf{N} \tag{1}$$

where $\mathbf{Y} = [\mathbf{y}_1, ..., \mathbf{y}_n] \in \mathbb{R}^{p\times n}$ is a HSI data cube containing the reflectance spectra of n pixels across p spectral bands, i.e.,, $\mathbf{y}_i \in \mathbb{R}^{p\times 1}$ is the spectra of i^{th} pixel; $\mathbf{E} = [\mathbf{e}_1, ..., \mathbf{e}_r] \in \mathbb{R}^{p\times r}$ is the endmember matrix containing the spectral signatures of r endmembers across p spectral bands, i.e., $\mathbf{e}_i \in \mathbb{R}^{p\times 1}$ models the spectra signature of the i^{th} endmember ($i = 1, ..., r$); $\mathbf{A} = [\mathbf{a}_1, ..., \mathbf{a}_n] \in \mathbb{R}^{r\times n}$ is the corresponding fractional abundance matrix, i.e., $\mathbf{a}_i \in \mathbb{R}^{r\times 1}$ is the abundance vector containing the abundances of r different endmembers present in the i^{th} pixel; and $\mathbf{N} \in \mathbb{R}^{p\times n}$ represents additive white gaussian noise. It should be noted that we discuss with flattened HSI image $\mathbf{Y} \in \mathbb{R}^{p\times n}$, flattened abundance map $\mathbf{A} \in \mathbb{R}^{r\times n}$, and flattened noise $\mathbf{N} \in \mathbb{R}^{p\times n}$ for the purpose of simpler notation, but we actually work with the HSI image of size $n_1 \times n_2$, i.e., $\mathbf{Y} \in \mathbb{R}^{p\times n_1\times n_2}$, $\mathbf{A} \in \mathbb{R}^{r\times n_1\times n_2}$, and $\mathbf{N} \in \mathbb{R}^{p\times n_1\times n_2}$.

Generally, the abundance is subjected to the non-negative constraint (ANC) and sum-to-one constraint (ASC), i.e., all the elements are equal to or greater than zero, $\mathbf{A} \geq \mathbf{0}$ [1] and the abundance of each pixel should sum up to one, $\mathbf{A}^T\mathbf{1}_r = \mathbf{1}_n$, where $\mathbf{1}_r$ is the all one vector with size $r \times 1$. Similarly, the endmember matrix is also subjected to non-negative constraint (ENC), $\mathbf{E} \geq \mathbf{0}$, in order to be physically meaningful.

The goal of blind linear unmixing is to estimate $\mathbf{E}$ and $\mathbf{A}$ given only $\mathbf{Y}$. A popular approach to address this type of problem involves solving [38]:

$$\begin{aligned} \hat{\mathbf{E}}, \hat{\mathbf{A}} = arg\min_{\mathbf{E},\mathbf{A}} \frac{1}{2}\|\mathbf{Y} - \mathbf{E}\mathbf{A}\|_F^2 + R(\mathbf{A}) \\ s.t., \mathbf{E} \geq \mathbf{0}, \mathbf{A} \geq \mathbf{0}, \mathbf{A}^T\mathbf{1}_r = \mathbf{1}_n \end{aligned} \tag{2}$$

where, $R(\mathbf{A})$ is a regularizer depending on abundance matrix $\mathbf{A}$, such as total variation (TV) [38]. Generally, the choice of R is heavily dependent on the prior knowledge about the task at hand. The optimisation problem (2) is typically solved by the multiplicative update rule [15] or a two-stage cyclic descent method [38], which alternates between optimising $\mathbf{A}$ for fixed $\mathbf{E}$, and vice versa.

Despite its popularity and simplicity, the linear model can not handle more complex hyperspectral unmixing scenarios [2] such as those depicted in Fig. 1b and Fig. 1c.

B. Nonlinear Mixing Model

As an alternative to LMM, the nonlinear mixing model (NLMM) [11], [12], [39], [40] accounts for the presence of nonlinear effects by introducing additional nonlinear interaction terms in the LMM. Generally, a NLMM can be expressed as follows:

$$\mathbf{Y} = \mathbf{E}\mathbf{A} + \mathbf{O} + \mathbf{N} \tag{3}$$

where $\mathbf{Y}, \mathbf{E}, \mathbf{A}, \mathbf{N}$ are akin to the same quantities appearing in (1) and $\mathbf{O}$ denotes the additional term accounting for nonlinear mixing effects. (3) is also known as robust LMM (rLMM), and $\mathbf{O}$ is used to denote the outlier terms in [14]. NLMM has been widely proposed in hyperspectral imaging literature, that differ from one another depending on how $\mathbf{O}$ is modelled.

For example, the Bilinear model [39] is one of the most popular variants of NLMM, which models the observed spectra $\mathbf{y}_k$ for the k^{th} pixel as follows:

$$\mathbf{y}_k = \mathbf{E}\mathbf{a}_k + \sum_{i=1}^{r-1}\sum_{j=i+1}^{r} \beta_{i,j,k}\mathbf{e}_i \odot \mathbf{e}_j + \mathbf{n}_k \tag{4}$$

where, $\mathbf{y}_k, \mathbf{a}_k, \mathbf{n}_k$ correspond to the k^{th} column vector of $\mathbf{Y}, \mathbf{A}, \mathbf{N}$ in (3). $\odot$ is the element-wise (Hadamard) product. The coefficient $\beta_{i,j,k}$ captures the degree of nonlinear interactions between the endmember $\mathbf{e}_i$ and $\mathbf{e}_j$. For more comprehensive discussions regarding the Bilinear model, please refer to the Supplementary.

The goal of blind nonlinear unmixing is also to estimate $\mathbf{E}$ and $\mathbf{A}$ (sometimes the outlier/nonlinear terms $\mathbf{O}$) given only $\mathbf{Y}$. Recently, this problem has been solved in [14] via minimising the following objective:

$$\begin{aligned} \min_{\mathbf{E},\mathbf{A},\mathbf{O}} \frac{1}{2}\|\mathbf{Y} - \mathbf{E}\mathbf{A} + \mathbf{O}\|_F^2 + R(\mathbf{O}) \\ s.t., \mathbf{E} \geq \mathbf{0}, \mathbf{O} \geq \mathbf{0}, \mathbf{A} \geq \mathbf{0}, \mathbf{A}^T\mathbf{1}_r = \mathbf{1}_n \end{aligned} \tag{5}$$

The objective (5) defines a robust nonnegative matrix factorisation (NMF) problem and can be solved via an iterative block-coordinate descent algorithm [14], which update each of the parameters $\mathbf{E}, \mathbf{A}, \mathbf{O}$ in turn while the other parameters are fixed.

C. Deep Image Prior (DIP)

DIP [35] was originally proposed to solve inverse problems such as denoising, given by:

$$\boldsymbol{x}^* = arg\min_{\boldsymbol{x}} \|\boldsymbol{x} - \boldsymbol{x}_0\|_2^2 + R(\boldsymbol{x}) \tag{6}$$

where, $\boldsymbol{x}_0$ is a noisy image, and R is a regularizer explicitly capturing the prior information about the clean image $\boldsymbol{x}$. Many contributions have concentrated on designing good regularisers, for example, Total Variation (TV) makes regions in the

[1] In this work, $\mathbf{A} \geq \mathbf{0}$ is used to denote that $\mathbf{A}$ is a matrix in which all elements are nonnegative.

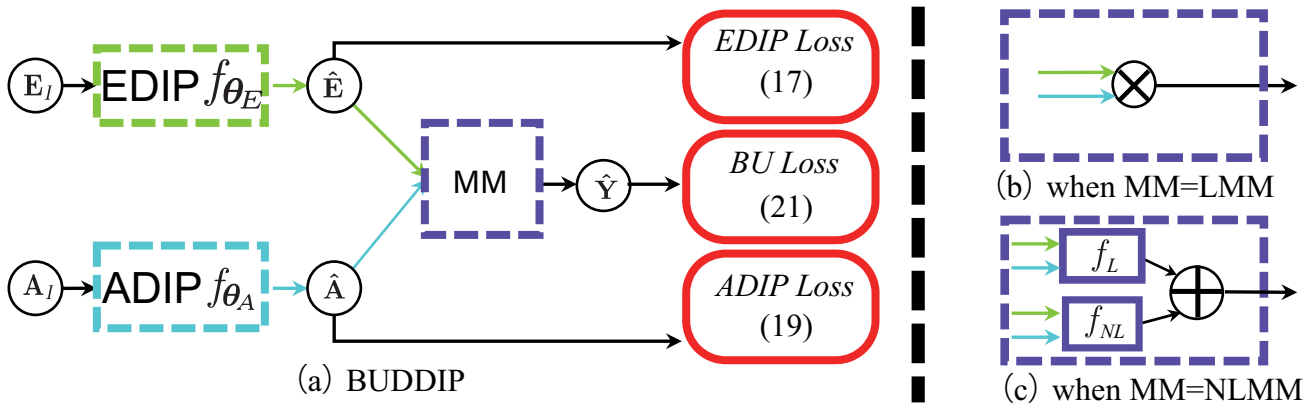


Fig. 2. (a) The general architecture of the proposed BUDDIP. It consists of three modules: endmember estimation DIP (EDIP), abundance estimation DIP (ADIP) and mixing module (MM). The output of EDIP is denoted by the green arrow while that of ADIP by the blue arrow. (b) when MM performs LMM as defined in (12), we coined the whole model in (a) as L-BUDDIP, which can solve the linear blind unmixing problem. (c) While MM performs NLMM as defined in (13), we coined the whole model in (a) as NL-BUDDIP, which can solve the nonlinear blind unmixing problem. f_L and f_{NL} are defined in (14) and (15), respectively.

image more uniform and plug-and-play prior [41] connects image inverse problems with well-developed image denoiser. DIP however solves the optimization problem given by:

$$\boldsymbol{\theta}^* = arg\min_{\boldsymbol{\theta}} \|f_{\boldsymbol{\theta}}(\boldsymbol{z}) - \boldsymbol{x}_0\|_2^2 \tag{7}$$

where, $f_{\boldsymbol{\theta}}(\boldsymbol{z})$ is a neural network parameterized by $\boldsymbol{\theta}$, with random input $\boldsymbol{z}$. That is, DIP effectively replaces the regularizer R in (6) with a neural network. According to the DIP techniques [34], [35], the optimisation problem (7) can be solved by randomly initialising the parameter $\boldsymbol{\theta}$, and using a network optimiser such as gradient descent to update $\boldsymbol{\theta}$ iteratively until a predetermined number of iterations is reached. After learning, the network parameterization would implicitly capture the prior and output the restored image given by $\boldsymbol{x}^* = f_{\boldsymbol{\theta}^*}(\boldsymbol{z})$.

The DIP technique has been used to solve linear unmixing problems in UnDIP [34], where the endmember matrix is estimated via the existing method SiVM, which is used to guide the training of an abundance estimation network using a DIP network. However, UnDIP itself does not provide endmember estimation and the quality of its abundance estimation is limited by the quality of the guidance. Another problem of DIP techniques is that they often suffered from overfitting, which is usually solved by regularisation techniques such as the early stopping technique [35] and exponentially weighted averaging over the outputs from different runs [34], respectively. We next describe how we adapt the DIP technique to solve these issues.

III. BUDDIP

We now introduce the proposed general framework for blind hyperspectral linear and nonlinear unmixing tasks using the double deep image prior (BUDDIP).

The proposed BUDDIP network is a self-supervised end-to-end network, which consists of three modules: EDIP, ADIP and MM, as shown in Fig. 2(a). EDIP is responsible for endmember estimation, whereas ADIP is responsible for abundance estimation. After obtaining the estimation of endmembers and abundances, the output of EDIP and ADIP modules will then be injected into MM to generate a reconstruction of the observed hyperspectral spectrum.

The ability to choose the MM module leads to the flexibility of the proposed BUDDIP. When MM is chosen to be an LMM, the proposed BUDDIP becomes linear BUDDIP (L-BUDDIP), which is capable of solving linear unmixing tasks. Similarly, when MM is chosen to be an NLMM, the proposed BUDDIP becomes non-linear BUDDIP (NL-BUDDIP), which can be used to solve non-linear unmixing tasks. In order to generate meaningful unmixing results, the proposed BUDDIP is trained with the guidance of endmembers and abundances, which are generated using any existing unmixing methods such as SiVM [7]+FCLS [8]. However, different from other unmixing networks trained with guidance such as UnDIP [34] or EGU-Net [2], the performance of the proposed BUDDIP can surpass that of the guidance.

A. Endmember Estimation using DIP

We now introduce how to solve the endmember estimation problem using DIP. Like the two-stage cyclic descent method [38], let us first assume that, at the endmember estimation stage, we are given access to an estimate of the abundance matrix $\mathbf{A}_G$. This could be achieved by using some existing algorithms such as FCLS [8]. Then, the optimization problem (2) would reduce to:

$$\hat{\mathbf{E}} = arg\min_{\mathbf{E}} \frac{1}{2}\|\mathbf{Y} - \mathbf{E}\mathbf{A}_G\|_F^2 \quad s.t., \mathbf{E} \geq \mathbf{0} \tag{8}$$

In this work, following the idea of DIP, we propose to estimate the endmembers using a DIP network (EDIP) $f_{\boldsymbol{\theta}_E}$ with learnable parameters $\boldsymbol{\theta}_E$ and an input $\mathbf{T}_E$. This leads to the optimization problem given by:

$$\boldsymbol{\theta}_E^* = arg\min_{\boldsymbol{\theta}_E} \frac{1}{2}\|\mathbf{Y} - f_{\boldsymbol{\theta}_E}(\mathbf{T}_E)\mathbf{A}_G\|_F^2 \tag{9}$$

In the original DIP [35] and unmixing work using DIP UnDIP [34], the network is fed with a random input, i.e., $\mathbf{T}_E = \boldsymbol{z}_E$, where $\boldsymbol{z}_E$ is Gaussian noise in [34] and uniform

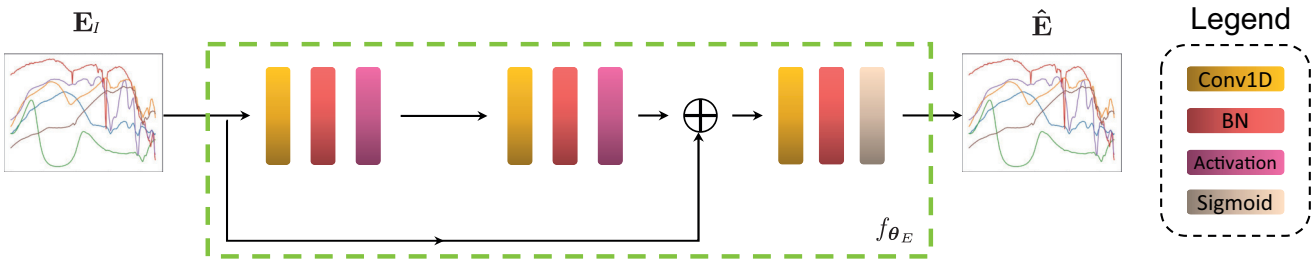


Fig. 3. The architecture of the proposed EDIP. We propose to give a meaningful input $\mathbf{T}_E = \mathbf{E}_I$, where $\mathbf{E}_I$ is an estimation of endmember generated by existing methods, such as SiVM [7]. The outputs of the main branch and the skipped input are added and forwarded to the final output block.

noise between zero and 0.1 in [35]. However, there are significant drawbacks to using random input as it does not contain any relevant information about the task or data that the network is designed to process. Another observation in [42] is that, when the network is given simultaneously a noisy observation and random noise, the network tends to ignore the noise. To address these drawbacks, we propose to give the DIP network a more meaningful input. Since there are many impressive unmixing works in the hyperspectral literature, we propose to use existing unmixing algorithms, such as SiVM [7], to generate an estimate of endmembers, $\mathbf{E}_I$, and use it as input to the proposed EDIP network, $\mathbf{T}_E = \mathbf{E}_I$. This can be viewed as a noisy estimation of the ground truth endmembers.

With the proposed input strategy, the network would already have a reasonable starting point and only need to learn the difference between the desired output and the input. This further motivates the design of the architecture of the proposed EDIP network $f_{\boldsymbol{\theta}_E}$. In particular, we use the ResNet [43] like structure, because the skip-connections proposed in ResNet can force the network to learn the difference between the input and output. Different from the common ResNet structure used in DIP and UnDIP, in this paper, we use a simpler network shown in Fig. 3.

The proposed EDIP network $f_{\boldsymbol{\theta}_E}$ is built upon a block that consists of a Convolutional layer, a Batch normalisation layer, and an Activation layer. This type of block is very popular in neural network architectures like ResNet in literature [35], [44]. However, different from the common 2D convolutional layer used in literature, we use the 1D convolutional layer in the proposed EDIP, because 2D convolutional networks are commonly used to tackle image related problems, where the image is represented as a 3D tensor. However, in the proposed EDIP network, the input and output are the endmember signatures represented as a 2D matrix. The 1D convolution will be performed over the spectral band dimension p in order to capture the spectral information as there is no spatial information in the endmember matrix. This block is repeated twice in the middle. As mentioned before, we want the network to learn the difference between the input and desired output, which is achieved via a skip connection shown in the side branch of Fig. 3. The output of the main branch and the skip connection will be added and injected into the final output block. The structure of the output block is the same as before except that the activation layer is replaced with the Sigmoid activation layer in order to meet the ENC constraint. After learning the parameter $\boldsymbol{\theta}_E^*$, the estimated endmember is given by $\hat{\mathbf{E}} = f_{\boldsymbol{\theta}_E^*}(\mathbf{E}_I)$.

B. Abundance Estimation using DIP

We now introduce how to derive the DIP network for abundance estimation. Let us now assume that, at the abundance estimation stage, we are given access to an estimate of the endmembers $\mathbf{E}_G$, using some algorithm such as SiVM [7]. Then, the optimization problem (2) would reduce to:

$$\begin{aligned} \hat{\mathbf{A}} = arg\min_{\mathbf{A}} \frac{1}{2}\|\mathbf{Y} - \mathbf{E}_G\mathbf{A}\|_F^2 + R(\mathbf{A}) \\ s.t., \mathbf{A} \geq \mathbf{0}, \mathbf{A}^T\mathbf{1}_r = \mathbf{1}_n \end{aligned} \tag{10}$$

We also propose to generate the estimation of the abundances using a DIP network (ADIP) $f_{\boldsymbol{\theta}_A}$ with an input $\mathbf{T}_A$ and learnable parameters $\boldsymbol{\theta}_A$. This leads to the optimization problem given by:

$$\boldsymbol{\theta}_A^* = arg\min_{\boldsymbol{\theta}_A} \frac{1}{2}\|\mathbf{Y} - \mathbf{E}_G f_{\boldsymbol{\theta}_A}(\mathbf{T}_A)\|_F^2 \tag{11}$$

Similar to EDIP, we give the proposed ADIP a more meaningful input instead of random noise $\boldsymbol{z}_A$. We propose to use existing unmixing algorithms to generate an estimate of abundance $\mathbf{A}_I$, and use it as the input of the proposed ADIP network, $\mathbf{T}_A = \mathbf{A}_I$.

Akin to EDIP, the core of designing the architecture of the ADIP network is to force the network to learn the difference between the input and desired output. As a result, we use a similar structure as EDIP with several modifications, which are shown in Fig. 4. First of all, the ADIP network is used to estimate the abundances, which is represented as a 3D tensor like an image. Thus, the convolutional layer in the ADIP network is a 2D convolution, which can capture spatial information via various convolutional kernels. Because the abundance maps are more complex than the endmember matrix, we use four blocks to capture the rich information in abundance maps. Another difference is that, in order to meet ASC and ANC, we use softmax as the output layer of the ADIP network. After learning the parameter $\boldsymbol{\theta}_A^*$, the estimated abundance is given by $\hat{\mathbf{A}} = f_{\boldsymbol{\theta}_A^*}(\mathbf{A}_I)$.

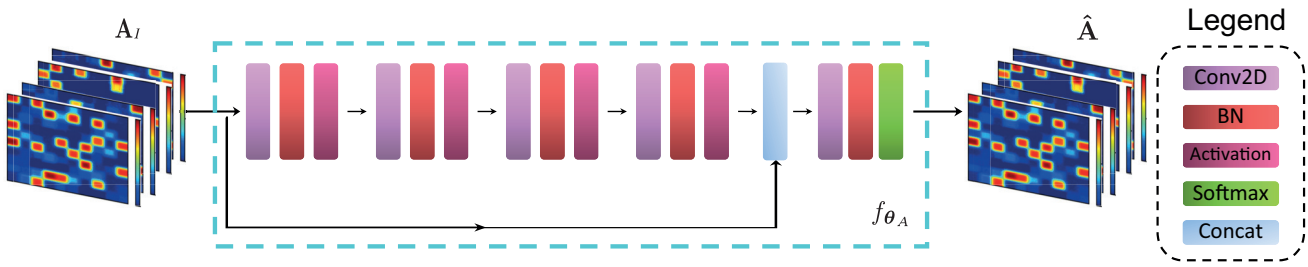


Fig. 4. The architecture of the proposed ADIP. We propose to give a meaningful input $\mathbf{T}_A = \mathbf{A}_I$, where $\mathbf{A}_I$ is an estimation of abundances generated by existing methods, such as FCLS [8]. The outputs of the skip connection and the main branch are concatenated and forwarded to the final output block.

C. Blind Unmixing using Double DIP

Finally, we will introduce how to construct the proposed general framework for blind unmixing using double DIP (BUDDIP).

After obtaining an estimation of endmember and abundance, $\hat{\mathbf{E}}$ and $\hat{\mathbf{A}}$, using EDIP and ADIP respectively, we can immediately generate a reconstruction of the observed HSI image by feeding $\hat{\mathbf{E}}$ and $\hat{\mathbf{A}}$ into a Mixture Module (MM). The structure of the MM, which we will introduce later, is very flexible and dependent on which physical model is selected. After combining MM with EDIP and ADIP, the architecture of the proposed general BUDDIP framework is shown in Fig. 2. This general framework can be used to solve both linear and nonlinear unmixing problems, by applying different MM.

1) Linear BUDDIP (L-BUDDIP): In the LMM case, according to (1), the MM would perform the linear mixing process as follows:

$$\hat{\mathbf{Y}}^M = \hat{\mathbf{E}}\hat{\mathbf{A}} \tag{12}$$

where $\hat{\mathbf{Y}}^M$ is the reconstructed HSI observation. This is illustrated in Fig.2(a)&(b). We coined it L-BUDDIP.

2) Nonlinear BUDDIP (NL-BUDDIP): In the NLMM case, according to the model (3), we can generate a reconstruction of the observed HSI image by feeding $\hat{\mathbf{E}}$, $\hat{\mathbf{A}}$ into the MM, which would perform the nonlinear mixing process as follows:

$$\hat{\mathbf{Y}}^M = f_L(\hat{\mathbf{E}}, \hat{\mathbf{A}}) + f_{NL}(\hat{\mathbf{E}}, \hat{\mathbf{A}}) \tag{13}$$

where,

$$f_L(\hat{\mathbf{E}}, \hat{\mathbf{A}}) = \hat{\mathbf{E}}\hat{\mathbf{A}} \tag{14}$$

is responsible for the linear effects in NLMM, and

$$f_{NL}(\hat{\mathbf{E}}, \hat{\mathbf{A}}) = \hat{\mathbf{O}} \tag{15}$$

would account for the nonlinear effects in NLMM, where $\hat{\mathbf{O}} = [\hat{\mathbf{o}}_1, ..., \hat{\mathbf{o}}_n] \in \mathbb{R}^{p \times n}$. The choice of f_{NL} is very flexible and dependent on which nonlinear model is selected. For example, in the FM model [45], according to (4), f_{NL} would output $\hat{\mathbf{O}}$ as follows:

$$\hat{\mathbf{o}}_k = \sum_{i=1}^{r-1} \sum_{j=i+1}^{r} a_{i,k} a_{j,k} \mathbf{e}_i \odot \mathbf{e}_j \tag{16}$$

An illustration of the proposed NL-BUDDIP is shown in Fig. 2(a)&(c). Note that for other models such as GBM with extra parameters, those parameters can be learned via another DIP network according to [44].

TABLE I
SUMMARY OF SOME NOTATIONS.

$\mathbf{E}_I$, $\mathbf{A}_I$	meaningful input for EDIP and ADIP respectively
$\mathbf{E}_G$, $\mathbf{A}_G$	training guidance for ADIP and EDIP respectively
$\hat{\mathbf{E}}$, $\hat{\mathbf{A}}$	estimation of endmembers and abundances by the proposed network
$\mathbf{Y}$, $\hat{\mathbf{Y}}^M$, $\hat{\mathbf{Y}}^E$,$\hat{\mathbf{Y}}^A$	HSI observation and its reconstruction by the MM, EDIP, ADIP module, respectively
$\boldsymbol{\theta}_E$, $\boldsymbol{\theta}_A$	learnable parameters of proposed EDIP and ADIP network

D. Training Details

We now elaborate on how we optimize the various parameters associated with BUDDIP.

1) Loss function: According to the discussion above, the network has generated three estimations: $\hat{\mathbf{E}}$ for endmembers, $\hat{\mathbf{A}}$ for abundances, and $\hat{\mathbf{Y}}^M$ for reconstructed HSI observation, as shown in Fig. 2. Correspondingly, we propose to train our BUDDIP network using six loss functions (two for each output). To make the notations used later clear, we summarise some notations in Table. I.

First, for endmember estimation $\hat{\mathbf{E}}$ by EDIP $f_{\boldsymbol{\theta}_E}(\mathbf{E}_I)$, motivated by the optimization problem (9) and the angle distance loss [37], we propose to use the loss function given by:

$$L_{EDIP}(\boldsymbol{\theta}_E) = \alpha_1 \cdot L_{EMSE} + \alpha_2 \cdot L_{EAng} \tag{17}$$

where,

$$\begin{aligned} L_{EMSE} &= \frac{1}{2}\|\mathbf{Y} - \hat{\mathbf{Y}}^E\|_F^2 \\ L_{EAng} &= \frac{1}{n}\sum_{k=1}^{n} \frac{180}{\pi} \cos^{-1}\left(\frac{\mathbf{y}_k^T \hat{\mathbf{y}}_k^E}{\|\mathbf{y}_k\|_2 \|\hat{\mathbf{y}}_k^E\|_2}\right) \\ \hat{\mathbf{Y}}^E &= f_{\boldsymbol{\theta}_E}(\mathbf{E}_I)\mathbf{A}_G \end{aligned} \tag{18}$$

α_1 and α_2 in (17) are the hyperparameters that control the relative importance of the corresponding loss term. $\hat{\mathbf{Y}}^E$ is the HSI reconstruction by EDIP $f_{\boldsymbol{\theta}_E}(\mathbf{E}_I)$, given the guidance $\mathbf{A}_G$. $\mathbf{y}_k, \hat{\mathbf{y}}_k^E$ are the k^{th} pixel of HSI observation $\mathbf{Y}$ and HSI reconstruction $\hat{\mathbf{Y}}^E$, respectively. The proposed loss function in (17) is an extension of (9) that includes an additional angle distance loss. The first loss term, L_{EMSE}, in (17) minimizes the discrepancy between $\mathbf{Y}$ and $\hat{\mathbf{Y}}^E$ in Euclidean distance, similar to the optimization problem in Eq. (9). The second

loss term, L_{EAng}, provides a measure of disparity from a geometric perspective. As demonstrated in [37], the inclusion of this additional angle loss term can enhance the performance of the unmixing network.

As discussed before, in (18), the EDIP network $f_{\boldsymbol{\theta}_E}$ is fed with a meaningful input $\mathbf{E}_I$, where $\mathbf{E}_I$ is an estimate of endmembers generated via some existing unmixing algorithms. Similarly, $\mathbf{A}_G$ is an estimate of abundances generated by some existing unmixing algorithms, which would serve as the guide to the training of the EDIP network. The guidance $\mathbf{A}_G$ would guarantee the EDIP network to yield a meaningful estimation of endmembers. In this paper, $\mathbf{E}_I$ and $\mathbf{A}_G$ are generated by existing unmixing methods such as SiVM [7] and FCLS respectively.

Similarly, for abundance estimation $\hat{\mathbf{A}}$ by ADIP $f_{\boldsymbol{\theta}_A}(\mathbf{A}_I)$, we propose to use the loss function given by:

$$L_{ADIP}(\boldsymbol{\theta}_A) = \alpha_3 \cdot L_{AMSE} + \alpha_4 \cdot L_{AAng} \tag{19}$$

where,

$$\begin{aligned} L_{AMSE} &= \frac{1}{2}\|\mathbf{Y} - \hat{\mathbf{Y}}^A\|_F^2 \\ L_{AAng} &= \frac{1}{n}\sum_{k=1}^{n} \frac{180}{\pi} \cos^{-1}\left(\frac{\mathbf{y}_k^T \hat{\mathbf{y}}_k^A}{\|\mathbf{y}_k\|_2 \|\hat{\mathbf{y}}_k^A\|_2}\right) \\ \hat{\mathbf{Y}}^A &= \mathbf{E}_G f_{\boldsymbol{\theta}_A}(\mathbf{A}_I) \end{aligned} \tag{20}$$

and α_3 and α_4 are loss weights of L_{AMSE} and L_{AAng}. $\hat{\mathbf{Y}}^A$ is the HSI reconstruction by ADIP $f_{\boldsymbol{\theta}_A}(\mathbf{A}_I)$, given the guidance $\mathbf{E}_G$. $\mathbf{y}_k, \hat{\mathbf{y}}_k^A$ are the k^{th} pixel of HSI observation $\mathbf{Y}$ and HSI reconstruction $\hat{\mathbf{Y}}^A$, respectively. Again, in (20), we need a pair of estimations $\mathbf{A}_I$ and $\mathbf{E}_G$ in order to provide ADIP network with a meaningful input and training guidance. Like before, $\mathbf{A}_I$ and $\mathbf{E}_G$ are generated by existing unmixing methods such as SiVM [7] and FCLS respectively.

Moreover, for reconstructed HSI observation $\hat{\mathbf{Y}}^M$, which is defined in (12) and (13), we also impose an additional loss function – the blind unmixing (BU) loss – given by:

$$L_{BU}(\boldsymbol{\theta}_E, \boldsymbol{\theta}_A) = \alpha_5 \cdot L_{BUMSE} + \alpha_6 \cdot L_{BUAng} \tag{21}$$

where,

$$\begin{aligned} L_{BUMSE} &= \frac{1}{2}\|\mathbf{Y} - \hat{\mathbf{Y}}^M\|_F^2 \\ L_{BUAng} &= \frac{1}{n}\sum_{k=1}^{n} \frac{180}{\pi} \cos^{-1}\left(\frac{\mathbf{y}_k^T \hat{\mathbf{y}}_k^M}{\|\mathbf{y}_k\|_2 \|\hat{\mathbf{y}}_k^M\|_2}\right) \end{aligned} \tag{22}$$

and α_5 and α_6 are the loss weights. $\mathbf{y}_k, \hat{\mathbf{y}}_k^M$ are the k^{th} pixel of HSI observation $\mathbf{Y}$ and HSI reconstruction $\hat{\mathbf{Y}}^M$, respectively. This additional loss L_{BU} is necessary because otherwise EDIP and ADIP would yield endmember and abundance estimates close to guidance $\mathbf{E}_G$ and $\mathbf{A}_G$, respectively.

The final loss function for linear blind unmixing is a combination of the above losses, as follows:

$$L(\boldsymbol{\theta}_E, \boldsymbol{\theta}_A) = L_{EDIP} + L_{ADIP} + L_{BU} \tag{23}$$

where, $L_{EDIP}, L_{ADIP}, L_{BU}$ are defined in (17), (19) and (21). From the perspective of unmixing, the terms L_{EDIP} and L_{ADIP} in the composite loss function ensure that the network produces meaningful endmembers and abundances, in the sense that $\hat{\mathbf{E}}, \hat{\mathbf{A}}$ cannot deviate too much from $\mathbf{E}_G, \mathbf{A}_G$. At the same time, L_{BU} allows the network to have the freedom to search for better estimations than the guidance $\mathbf{E}_G, \mathbf{A}_G$. From the perspective of network training, L_{EDIP} and L_{ADIP} can be interpreted as regularizations on the outputs of the BUDDIP network, $\hat{\mathbf{E}}$ and $\hat{\mathbf{A}}$, with L_{BU} serving as the data fidelity term. This composite loss function can therefore alleviate the need for techniques such as early stopping [35] and exponentially averaging over different runs [34]. By properly choosing the hyperparameters $\alpha_{1\sim6}$, the network outputs will eventually reach an equilibrium state between inducing small fitting error (L_{BU}), and regularisation penalty, (L_{EDIP} and L_{ADIP}).

In this paper, the proposed loss function (23) is applied to both linear and nonlinear scenarios. While it is possible to modify the form of $\hat{\mathbf{Y}}^A$ and $\hat{\mathbf{Y}}^E$ to accommodate nonlinear reconstruction of hyperspectral data, we choose to retain the linear form of these terms in the nonlinear case for the sake of simplicity and consistency between the linear and nonlinear cases. However, the purpose of L_{EDIP} and L_{ADIP} is to ensure that the network produces meaningful output, which is still valid under nonlinear conditions without the need for modification. This is because the nonlinear model (3) still contains a linear component. For this reason, the meaningful input $\mathbf{E}_I, \mathbf{A}_I$, and training guidance $\mathbf{E}_G, \mathbf{A}_G$ are generated using the same unmixing algorithms (SiVM+FCLS) for both linear and nonlinear cases.

Different from the two-stage-cyclic descent method [38], the proposed network is trained in an end-to-end manner. Given only the HSI image $\mathbf{Y}$, the learnable parameters $\{\boldsymbol{\theta}_E, \boldsymbol{\theta}_A\}$ are learned by minimizing the composite loss in (23), using a variant of gradient descent optimizer, ADAM [46].

Algorithm 1 Adaptive Loss Weight Strategy.

INPUT :

- $\alpha_{1\sim6}^{init}$ - the initial value of loss weights
- γ_1, γ_2 - the rate of updating loss weights
- $\alpha_{min}, \alpha_{max}$ - the boundary of loss weights
- g - update gap
- J - the number of training epochs.

TRAIN :

```
Initialise α_{1∼6} ← α_{1∼6}^{init}
for i = 0 : J do
    train the proposed network with α_{1∼6};
    if i mod g == 0 then
        [α_{1∼4}, α_{5∼6}] ← [α_{1∼4} * γ_1, α_{5∼6}/γ_2]
        α_{1∼6} ← Clip (α_{1∼6}, α_min, α_max)
    end if
end for
return
```

2) Adaptive loss weight strategy for NL-BUDDIP: To further improve the nonlinear unmixing performance, we also propose an adaptive loss weight strategy. If $\alpha_{1\sim4}$ are relatively larger than $\alpha_{5\sim6}$, the network would simply mimic the guidance $\mathbf{E}_G$ and $\mathbf{A}_G$. On the contrary, if $\alpha_{5\sim6}$ is larger, the network would simply output meaningless endmembers

and abundances. Thus, for the nonlinear case, we propose an adaptive loss weight strategy. The motivation is that, at the beginning of the training, we would like the network to converge to the guidance fast, so the loss weight of EDIP and ADIP, $\alpha_{1\sim4}$ should be larger. After approaching the guidance, the network should have more freedom to explore for a better solution, so we need to reduce $\alpha_{1\sim4}$ whilst increasing $\alpha_{5\sim6}$. Finally, to avoid the weight exploding and vanishing problems, we need to set a threshold for the loss weights. This strategy is summarised in Alg. 1.

IV. Experiments

We now conduct various experiments on both synthetic and real datasets, for linear and nonlinear unmixing tasks, to show the effectiveness of the proposed methods. In view of the space constraints, we have included more experiments in the supplementary materials.

A. Performance Metrics

We adopt some of the most popular metrics in literature to evaluate the unmixing performance of various algorithms. In particular, we employ root mean square error (RMSE), and abundance angle distance (AAD) [16], between the true abundance vector $\mathbf{a}_k$ for k^{th} pixel and the corresponding estimation $\hat{\mathbf{a}}_k$, to measure the quality of abundance estimation. These metrics are given by:

$$RMSE_k = \sqrt{\frac{1}{r}\sum_{i=1}^{r}(a_{i,k} - \hat{a}_{i,k})^2} \tag{24}$$

$$AAD_k = \frac{180}{\pi}\cos^{-1}\left(\frac{\mathbf{a}_k^T\hat{\mathbf{a}}_k}{\|\mathbf{a}_k\|_2\|\hat{\mathbf{a}}_k\|_2}\right) \tag{25}$$

These metrics are then further averaged over the number of pixels to yield the final scalar quantities.

As for the endmember estimation, we also adopt the well-known spectral angle distance (SAD) to measure the dissimilarity between a true endmember signature $\mathbf{e}_i$ and the corresponding estimate $\hat{\mathbf{e}}_i$, where SAD is given by:

$$SAD_i = \frac{180}{\pi}\cos^{-1}\left(\frac{\mathbf{e}_i^T\hat{\mathbf{e}}_i}{\|\mathbf{e}_i\|_2\|\hat{\mathbf{e}}_i\|_2}\right) \tag{26}$$

This metric is instead averaged over the various endmembers to give the final scalar quantity.

B. Data

We also use both synthetic and real datasets to evaluate the performance of various unmixing algorithms.

1) Synthetic Dataset 1: We adopt the popular procedure in [27] to generate the synthetic HSI dataset for the linear unmixing problem. For the nonlinear unmixing case, we only need to modify the final linear mixing process with the corresponding nonlinear mixing counterpart. The procedure is as follows:

- *Endmember generation.* The endmembers are generated by selecting mineral signatures from the famous USGS spectral library known as splib06 [47]. The library contains values of spectral reflectance associated with different minerals over 224 channels, ranging from 0.4 μm to 2.5 μm. Six spectral signatures are randomly selected from the library, resulting in a 224×6 endmember matrix. These endmember signatures are shown in Fig. 5.

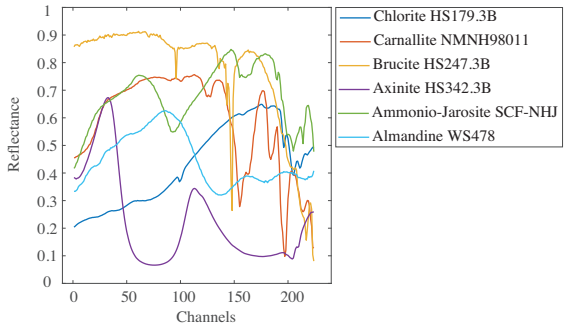


Fig. 5. Endmember signatures for synthetic data.

- *Abundance generation.* The abundances underlying the synthetic data is generated as follows. First, we divide a synthetic image of size $a^2 \times a^2$ pixels into a^2 disjoint patches of size $a \times a$ pixels. Second, for all pixels of a given patch, we randomly select two endmembers out of the six spectral signatures, and assign them with fractions γ and $1-\gamma$, while the remaining four endmembers are assigned with value 0. Finally, the abundance map is convolved with a Gaussian filter of size $(a+1)\times(a+1)$, with variance set to be 2, followed by pixel-wisely re-scaling to meet the ASC constraint. In this paper, we set $a = 10$ and $\gamma = 0.8$.
- *Mixing process.* According to different mixing models (1) or (3), we generate correspondingly linear or nonlinear mixing synthetic data.
- *Noise contamination.* Finally, the generated HSI data is contaminated with additive white gaussian noise (AWGN). The signal-to-noise ratio (SNR) is defined as $SNR = 10\log_{10}\left(E[\mathbf{x}^T\mathbf{x}]/E[\mathbf{n}^T\mathbf{n}]\right)$, where $\mathbf{x}$ is the clean HSI data, and $\mathbf{n}$ is the noise.

2) Synthetic Dataset 2: To test the effectiveness of the proposed methods when there is no pure pixel in HSI dataset, we also adopt the synthetic procedure in [23], [48]–[50], where a purity measure for an observed pixel $\mathbf{y}_k$ is defined as $\rho_k = \|\mathbf{a}_k\|_2 \in [1/\sqrt{r}, 1]$ (due to the ANC and ASC constraint). The larger ρ_k is, the higher the purity of the pixel $\mathbf{y}_k$ is. A set of n observed pixels $\{\mathbf{y}_k\}_{k=1}^n$ with $\rho - 0.1 \leq \rho_k \leq \rho$ is called a dataset with purity level of ρ. The synthetic generation procedure is basically the same as the procedure of synthetic dataset 1 except for the abundance generation, which is as follows:

- Randomly sampling $K = 10n$ abundance vectors from a Dirichlet distribution $D(\mu)$ where $\mu = (1/r)\mathbf{1}_r$, that is,

$$\Omega = \{\mathbf{a}_k | \mathbf{a}_k \sim D(\mu), \forall k = 1, ..., K\} \tag{27}$$

and calculate the corresponding purity ρ_k for all k.

- Construct a set of n abundance vectors with purity level ρ by randomly choosing n samples from Ω subject to $\rho_k \in [\rho - 0.1, \rho]$.

3) Real Dataset: We also adopt the commonly used real HSI datasets Jasper Ridge [51], which contains 512×614 pixels associated with 224 spectral bands ranging from 380 nm up to 2500 nm with a spectral resolution of 9.46 nm. There are four endmembers in the scene: Road, Soil, Water, and Tree. A 100×100 pixels sub-images of the original images is considered in this article to lift the computational burden and deploy faster experimental studies. Due to dense water vapour and atmospheric effects, We also eliminate 26 spectral bands: channels 1-3, 108-112, 154-166 and 220-224, leaving only 198 out of the 224 bands for unmixing purposes. A representative image – associated with the 80th spectral band – is shown in Fig. 6.

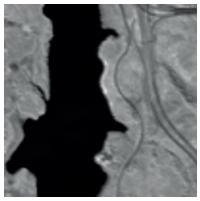


Fig. 6. HSI image Jasper Ridge at 80th channel.

C. Evaluation on synthetic data

1) Linear Unmixing: We now compare the performance of our approaches under the linear mixing model using synthetic datasets, with that of some state-of-art learning-based methods, UnDIP [34], EGU-Net-ss [2], MNN-BU-2 [36], U-ADMM-BUNet-II [37], CNNAEU [52] and MiSiCNet [53]. We also include the traditional method SiVM [7]+FCLS [8], HyperCSI [23], HiSun [22], and EDAA [17], where SiVM+FCLS is used to generate the training guide for some of the learning-based methods. Unless explicitly mentioned, the experiments are conducted under the default setting, where the structure of the proposed network is summarised in Table. II. The proposed network is trained using the Adam optimizer with a learning rate set to $5e-3$, and the number of epochs set to 6000. The synthetic HSI image dataset for both synthetic dataset 1&2 is constructed under the linear mixing model, which consists of 100×100 pixels. We also contaminate these HSI reflectances with AWGN leading to $SNR = 30$ dB.

Hyperparameter Tuning: In this section, we evaluate the impact of hyperparameters $\alpha_{1\sim6}$ on unmixing performance using synthetic dataset 1. We retain the default experiment settings except that the hyperparameters vary in $\{0.0, 0.001, 0.01, 0.1, 1.0\}$. The results are shown in Fig. 7. It is clear that, when $\alpha_{1\sim4} = 0$, that is L_{EDIP} and L_{ADIP} are deactivated, the network generates meaningless unmixing results. When $\alpha_5 = \alpha_6 = 0$ and any of $\alpha_{1\sim4}$ is 1, the network would simply output unmixing results close to the corresponding guidance provided by SiVM+FCLS. When $\alpha_2 = \alpha_4 = \alpha_6 = 0, \alpha_1 = 0.1, \alpha_3 = 0.01, \alpha_5 = 1$, i.e., $L_{EAng}, L_{AAng}, L_{BUAng}$ are deactivated and $L_{EMSE}, L_{AMSE}, L_{BUMSE}$ are activated, the proposed method generate better unmixing performance with abundance AAD of 5.67 and endmember SAD of 1.88. Finally, when all the six loss terms are activated, that is, $\alpha_2 = 0.001, \alpha_4 = 0.01, \alpha_6 = 0.1, \alpha_1 = \alpha_3 = \alpha_5 = 1$, the proposed method delivers the best unmixing performance with AAD of 4.65 and SAD of 1.68. Consequently, we have employed these configurations as the default settings for the subsequent experiments conducted on the synthetic dataset. This experiment also indicates that the proposed method is very sensitive to hyperparameters $\alpha_{1\sim6}$ as they control the relative importance of regularisation L_{EDIP}, L_{ADIP} and data fidelity L_{BU}.

TABLE II
HYPERPARAMETERS OF BUDDIP STRUCTURE.

EDIP					
	In Channel	Out Channel	kernel size	stride	pad
Conv1D	p	256	3	1	same
	256	p	3	1	same
	p	p	1	1	same
ADIP					
	In Channel	Out Channel	kernel size	stride	pad
Conv2D	r	32	3	1	same
	32	64	3	1	same
	64	64	3	1	same
	64	r	3	1	same
	$2r$	r	1	1	same
Activation		LeakyReLU		negative_slope=0.1	

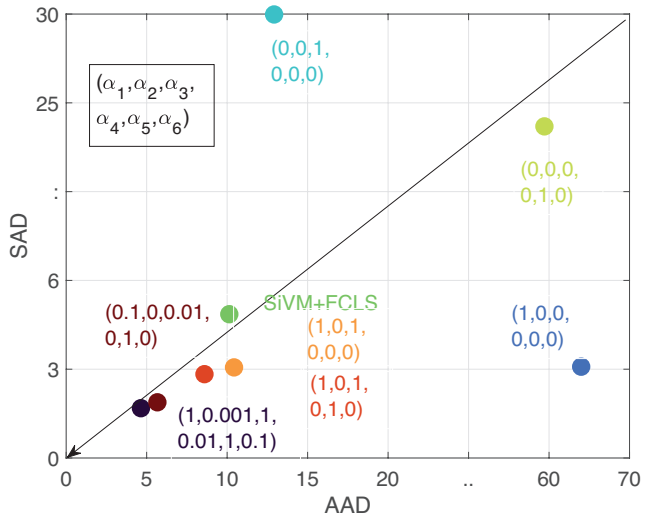
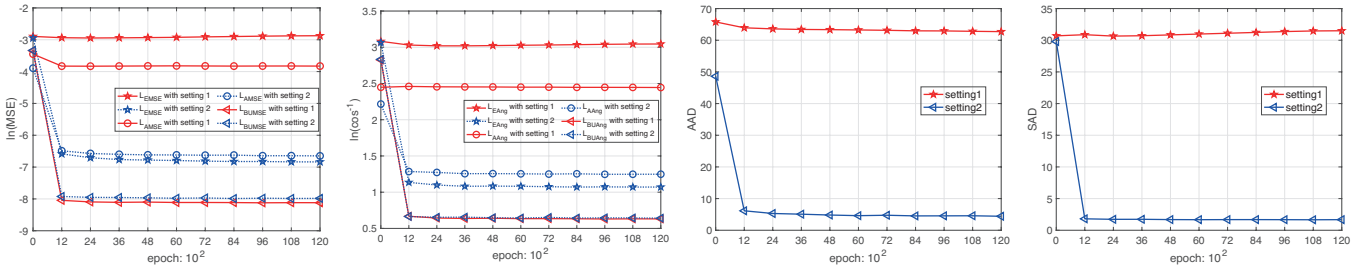


Fig. 7. The impact of hyperparameter on linear unmixing performance.

Convergence Analysis: This experiment presents an analysis of the convergence behavior of the proposed method, by comparing the training process and blind unmixing performance under two different settings. In the first setting, denoted as "setting 1", the network is trained without any regularization or guidance by deactivating the L_{EDIP} and L_{ADIP} terms and setting $\alpha_{1\sim4} = 0$ and $\alpha_5 = 1$, $\alpha_6 = 0.1$. In the second setting, denoted as "setting 2", the L_{EDIP} and L_{ADIP} terms are activated by setting $\alpha_1 = \alpha_3 = \alpha_5 = 1.0$, $\alpha_2 = 0.001$,



(a) various MSE loss versus epoch (b) various angle distance loss versus epoch (c) Abundance AAD versus epoch (d) Endmember SAD versus epoch

Fig. 8. Convergence analysis. In setting 1, we train the network with $\alpha_5 = 1, \alpha_6 = 0.1, \alpha_{1\sim4} = 0$, that is, the regularisation/guidance L_{EDIP} and L_{ADIP} are deactivated. In setting 2, we activate the regularisation/guidance L_{EDIP} and L_{ADIP} by using $\alpha_1 = \alpha_3 = \alpha_5 = 1.0, \alpha_2 = 0.001, \alpha_4 = 0.01, \alpha_6 = 0.1$. (a) various MSE loss versus epochs, (b) various angle distance loss versus epochs, (c) Abundance AAD versus epochs. (d) Endmember SAD versus epoch.

$\alpha_4 = 0.01$, and $\alpha_6 = 0.1$. The number of epochs is set to 12000 for analysis purposes, while the remaining settings are kept as default.

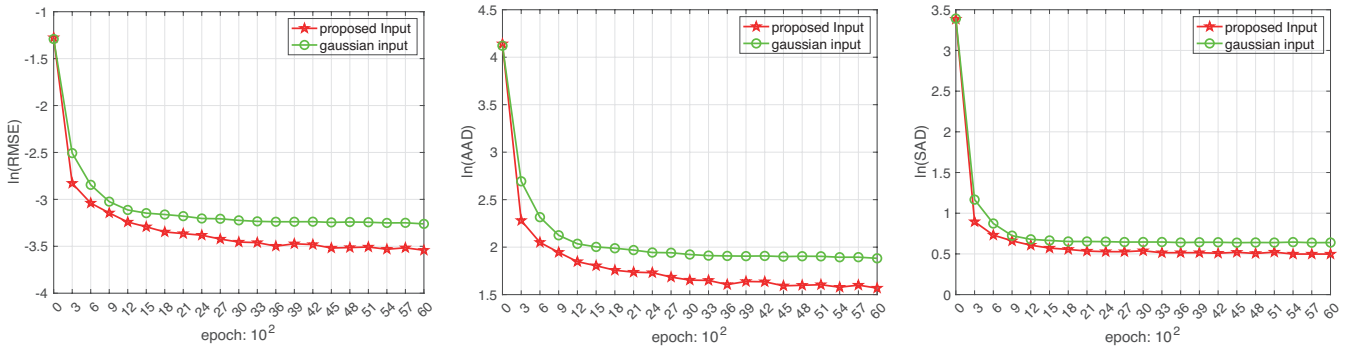
The training process and the corresponding unmixing performance are plotted in Fig. 8 as a function of the number of epochs. In setting 1, the network converges at around 1200 epochs and achieves the smallest fitting error L_{BUMSE}, but the worst unmixing performance, as evidenced by the high AAD and SAD values shown in Fig. 8c and Fig. 8d, respectively. Furthermore, the endmember SAD values increase with more epochs, indicating overfitting. Moreover, the regularisation penalties L_{EMSE}, L_{AMSE}, L_{EAng}, and L_{AAng} in Fig. 8a and Fig. 8b are also high.

In contrast, in setting 2 where all the regularization terms are activated, the network reaches an equilibrium state at around epoch 6000 as shown in Fig. 8, where it minimizes both the fitting error (L_{BUMSE} and L_{BUAng}) and the regularization penalties (L_{EMSE}, L_{AMSE}, L_{EAng}, and L_{AAng}). This setting also yields improved unmixing performance, with an abundance AAD of approximately 4.4 and an endmember SAD of 1.67. Therefore, the proposed method does not require the use of techniques such as early stopping [35] or exponentially weighted averaging of outputs from multiple runs [34], since the L_{EDIP} and L_{ADIP} terms serve as effective regularizations. We choose 6000 epochs as a trade-off between training efficiency and unmixing quality.

Comparison of different input: We now evaluate the effectiveness of the proposed input strategy using synthetic dataset 1. We retain the default settings for BUDDIP except that the network now is fed with two types of input: gaussian input as UnDIP suggested [34], and the proposed input. The unmixing performance with respect to training epochs are shown in Fig. 9. It can be seen that with the proposed input strategy, the network achieves better unmixing when the number of epochs is small, say, 300. We attribute this to the fact that the proposed input can be viewed as a noisy observation of the underlying ground truth. Thus, the task of the network is to generate more refined unmixing results given the noisy input. In comparison, with the gaussian input, the network needs to generate unmixing results given non-informative input. It is also noticeable that, with the proposed input, BUDDIP also delivers better unmixing results when the number of epochs is large.

Performance vs. Purity Level ρ: In this section, we evaluate the impact of purity level ρ on the linear unmixing performance of various unmixing methods using synthetic dataset 2. Specifically, we retain the default settings and the purity ρ varies in $[0.8, 0.9, 1.0]$, where $\rho = 0.8$ indicates the case where the HSI data is highly mixed and $\rho = 1.0$ means there are highly pure pixels. We also train the proposed method with guidances generated from two different methods, SiVM+FCLS and HyperCSI, respectively. The unmixing performance is summarised in Table. III. It is evident that, when the data is highly mixed, i.e., $\rho = 0.8$ and $\rho = 0.9$, HyperCSI performs the best among the various competing methods. Nevertheless, when the proposed method is trained with the guidance from HyperCSI, it achieves almost 1.6 times better performance than HyperCSI. For example, when HyperCSI shows abundance AAD of 2.19 and endmember SAD of 0.80 at purity level 0.8, the proposed method can further improve it to AAD of 1.97 and SAD of 0.49. In the meantime, although SiVM+FCLS failed to deliver satisfactory unmixing performance, the proposed network can use the unmixing results from SiVM+FCLS as the training guidance and further improve the performance by three times, from an abundance AAD of 11.4 to 3.82. At a purity level of $\rho = 1.0$, the proposed L-BUDDIP guided with SiVM has the lowest RMSE and AAD, as well as the second best SAD. Comparing the performance of L-BUDDIP trained with different guidance, it can be seen that the performance of the proposed method is affected by the quality of the guidance used during training. A higher quality guidance leads to better performance of the proposed method. Overall, the proposed L-BUDDIP performs well across different purity levels. Additionally, the proposed method can utilize the unmixing results of the state-of-the-art as training guidance (represented by $\mathbf{E}_G$ and $\mathbf{A}_G$) to further improve its performance.

Processing Time Comparison: In this section, we compare the processing time of various unmixing methods. The test platform is equipped with Intel Xeon Gold 6248 CPU 2.50GHz, Tesla V100 GPU, 503GB RAM. The results for linear unmixing case and nonlinear unmixing case are shown in Table. III and Table. IV, respectively. The quantities are obtained by averaging over ten individual runs. It is evident



(a) Abundance RMSE versus epoch (b) Abundance AAD versus epoch (c) Endmember SAD versus epoch

Fig. 9. Linear unmixing performance of BUDDIP with different input strategies: gaussian input and the proposed input. (a) Abundance RMSE versus epoch. (b) Abundance AAD versus epoch. (c) Endmember SAD versus epoch.

TABLE III
LINEAR UNMIXING PERFORMANCE COMPARISON OF VARIOUS METHODS FOR DIFFERENT PURITY LEVEL ρ. THE BEST RESULTS ARE HIGHLIGHT IN BOLD. THE SECOND BEST ARE DENOTED WITH A STAR.

method	$\rho = 0.8$			$\rho = 0.9$			$\rho = 1.0$			time (:s)
	RMSE	AAD	SAD	RMSE	AAD	SAD	RMSE	AAD	SAD	
SiVM+FCLS	0.0692	11.4335	5.0056	0.0328	4.4675	2.4792	0.0106	1.1287⋆	0.4726	11.9374
HyperCSI	0.0140⋆	2.1875⋆	0.7998⋆	0.0120⋆	1.6817⋆	0.5989⋆	0.0114	1.4366	0.5826	0.1181
HiSun	0.0439	5.6432	4.7341	0.0237	2.7296	2.0948	0.0251	2.3893	0.8225	8.7195
EDAA	0.0633	10.4134	3.5987	0.0701	10.3568	2.0083	0.0316	3.3124	**0.2614**	12.2705
MNN-BU2	0.0614	10.2151	3.3293	0.0831	12.6412	3.2022	0.0256	2.3661	1.1599	0.1926
U-ADMM-BUNet-II	0.0348	5.7705	1.7182	0.0461	6.7399	1.5596	0.0227	2.8388	0.6132	0.1490
EGU-Net-ss	0.0959	13.6120	4.6811	0.0538	6.0256	2.7246	0.0177	1.9359	0.7267	0.3659
UnDIP	0.0720	11.9410	5.0056	0.0345	4.7341	2.4792	0.0139	1.6776	0.4726	0.0086
MiSiCNet	0.0953	14.3151	6.0990	0.0657	8.0372	4.6247	0.0292	3.3278	2.8937	0.0023⋆
CNNAEU	0.3667	65.1646	16.4183	0.4160	69.2101	16.8180	0.4234	70.9834	14.7135	0.0054
L-BUDDIP (SiVM)	0.0236	3.8200	1.5430	0.0161	2.2130	0.9460	**0.0104**	**1.0941**	0.2920⋆	**0.0018**
L-BUDDIP (HyperCSI)	**0.0126**	**1.9681**	**0.4875**	**0.0119**	**1.5704**	**0.4587**	0.0105⋆	1.1408	0.4383	

that the network-based unmixing methods generally outperform traditional unmixing methods in terms of processing time, with the exception of HyperCSI. Among the network-based methods, The proposed method stand out as the fastest unmixing approach.

2) Nonlinear Unmixing: We now evaluate the unmixing performance of the proposed NL-BUDDIP with various state-of-art unmixing methods, such as network based methods UnDIP [34], EGU-Net-ss [2], MNN-BU-2 [36], U-ADMM-BUNet-II [37], CNNAEU [52] and MiSiCNet [53], as well as the traditional methods SiVM [7]+FCLS [8], and rNMF [14], HyperCSI [23], HiSun [22], and EDAA [17], using synthetic dataset under FM model. Correspondingly, in the proposed network, the Mixing Module (MM) would use the FM model [45] as defined in (4), while the structure of EDIP and ADIP module are the same as those summarised in Table. II. We also use the Adam optimiser with learning rate set to $5e-3$ and the number of epochs set to 12000. By default, the synthetic HSI image dataset 1&2 are generated under FM model, which consists of 100×100 pixels. The synthetic dataset is then polluted with AWGN noise leading to $SNR = 30$ dB. The hyperparameters are set to $\alpha_1^{init} = 100, \alpha_2^{init} = \alpha_5^{init} = 1, \alpha_3^{init} = \alpha_4^{init} = 10, \alpha_6^{init} = 0.1, \gamma_1 = 0.8, \gamma_2 = 0.9, \alpha_{min} = 1e-3, \alpha_{max} = 1e+2, g = 300$ by grid search techniques.

Fixed vs. Adaptive Loss Weight Strategy: We now evaluate the impact of the proposed adaptive loss weight strategy using the nonlinear HSI synthetic dataset 1 under FM model. Specifically, we vary the hyperparameters $\alpha_2^{init}, \alpha_4^{init}, \gamma_1, \gamma_2$, while the remaining are fixed as the default setting. The unmixing performance is reported in Fig. 10. The dots with $\gamma_1 = \gamma_2 = 1$ are results of the fixed loss weight strategy, while the others are results of the adaptive loss weight strategy. It is clear that, with the fixed strategy, the network can readily generate unmixing results better than the guidance. Nevertheless, with the proposed adaptive loss weight strategy,

the network achieves the best unmixing performance in terms of both endmember SAD and abundance AAD.

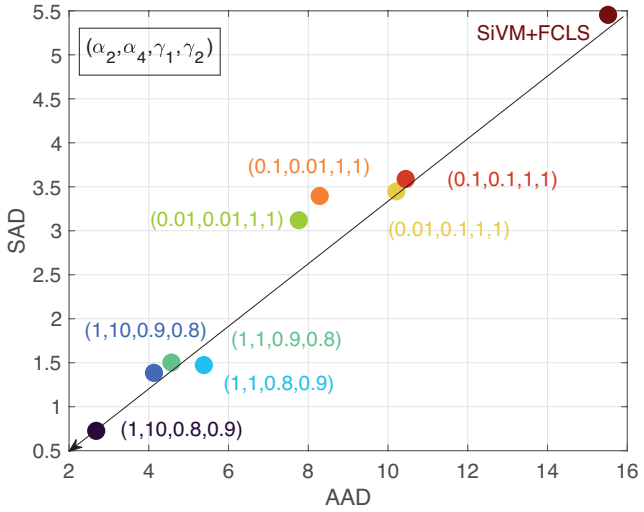


Fig. 10. Fixed versus adaptive loss weight strategy under nonlinear unmixing case. X axis is abundance AAD and Y axis is endmember SAD. The dots with $\gamma_1 = \gamma_2 = \gamma_3 = 1$ are the fixed loss weight strategy, while the remaining are adaptive loss weight strategy.

Performance vs. Purity Level ρ: In this section, we evaluate the impact of purity ρ on the nonlinear unmixing performance of the proposed methods using synthetic HSI dataset 2 under FM model. Specifically, we retain the default settings and the purity ρ varies in $[0.8, 0.9, 1.0]$. The unmixing performance is summarised in Table. IV. It is clear that, at the purity level of $\rho = 0.8$, HyperCSI performs the best among the various competing methods with AAD of 4.6 and SAD of 1.5. On the other hand, although SiVM+FCLS only achieves AAD of 17.96 and SAD of 6.34, the proposed NL-BUDDIP can utilise those unmixing results as guidance and delivers 7 times better performance with AAD of 3.9959 and SAD of 0.8752, which is even better than HyperCSI. When ρ increases to 0.9 and 1.0, the proposed NL-BUDDIP achieves the best performance among all competitors. In general, the proposed NL-BUDDIP shows the state-of-the-art nonlinear unmixing performance across different purity levels.

D. Evaluation on real dataset

We conducted an evaluation of the unmixing performance of various methods on the real dataset Jasper Ridge. Comparisons on other real dataset can be found in the appendix. The methods used in this comparison were traditional linear unmixing methods, namely SiVM [7]+FCLS [8], HyperCSI [23], HiSun [22], and EDAA [17], as well as traditional nonlinear unmixing methods, including rNMF [14]. Additionally, we evaluated learning-based unmixing networks such as MNN-BU-2 [36], U-ADMM-BUNet-II [37], EGU-Net-ss [2], UnDIP [34], CNNAEU [52], MiSiCNet [53], and our proposed L-BUDDIP and NL-BUDDIP. In this experiment, we utilized the MM module in the NL-BUDDIP to function as the FM model. We employed SiVM+FCLS as the default guidance/initialization generator for various methods such as MNN-BU-2, U-ADMM-BUNet-II, EGU-Net-ss, UnDIP, MiSiCNet, and the proposed L-BUDDIP. Finally, we trained both L/NL-BUDDIP using guidance generated from EDAA to demonstrate state-of-the-art performance.

The proposed L-BUDDIP with guidance from EDAA is trained using the hyperparameters $\alpha_1 = \alpha_3 = \alpha_6 = 0.01$, $\alpha_2 = \alpha_4 = 100$, and $\alpha_5 = 1.0$, while the hyperparameters for NL-BUDDIP with guidance from EDAA are set to $\alpha_1^{init} = \alpha_3^{init} = \alpha_4^{init} = 10$, $\alpha_2^{init} = 100$, $\alpha_5^{init} = 0.1$, $\alpha_6^{init} = 0.01$, $\gamma_1 = \gamma_2 = 0.9$, $\alpha_{min} = 0.01$, $\alpha_{max} = 100$, and $g = 700$. These hyperparameters were selected using grid search techniques. In contrast, the hyperparameters for L-BUDDIP with guidance from SiVM+FCLS were set to $\alpha_1 = 45.25$, $\alpha_2 = 100$, $\alpha_3 = 16.60$, $\alpha_4 = 47.16$, $\alpha_5 = 1.0$, and $\alpha_6 = 0.08$, which were determined using random search techniques. All networks were trained using a learning rate of $5e-3$ and epoch 6000. The qualitative results of estimated endmembers and abundances are shown in Fig. 11 and Fig. 12, respectively. The corresponding quantitative results are shown in Table. V. The results presented in Fig. 11 show that several existing methods, such as SiVM+FCLS, CNNAEU, UnDIP, and EGU-Net, failed to accurately estimate the signatures of the road, resulting in large endmember SAD values as indicated in Table. V. Moreover, both HyperCSI and HiSun failed to unmix the signature of water, while MiSiCNet produce the best estimation for the endmember of Tree for this dataset. In contrast, the proposed L-BUDDIP with guidance from SiVM+FCLS was able to recover more accurate endmember signatures for the road. Specifically, Table V shows that L-BUDDIP with SiVM+FCLS guidance achieved an Endmember SAD of 4.13 for the road, which represents a significant improvement compared to SiVM+FCLS, which achieved an SAD of 15.8 for the road.

Furthermore, when using guidance from EDAA, the proposed methods generated the most visually appealing unmixing results. The quantitative results presented in Table. V demonstrate that the proposed method has the best SAD for the road endmember and the overall average SAD among all endmembers. With improved endmember estimation, the proposed method also achieved the best abundance estimation in terms of RMSE and AAD. Moreover, NL-BUDDIP slightly outperformed L-BUDDIP in terms of abundance estimation. Although EDAA readily delivered good unmixing results with an AAD of 7.66, the proposed method was able to further improve the AAD to 5.2. Finally, the qualitative results presented in Fig. 12 indicate that the proposed method generates the best abundance estimation compared to its competitors.

V. Conclusion

In this work, we have proposed a general neural network structure that is capable of solving both linear and nonlinear hyperspectral blind unmixing problems. Different from the popular autoencoder structure, building upon Deep Image Prior (DIP) techniques, the proposed method consists of three modules: an Endmember estimation module using DIP (EDIP), an Abundance estimation module using DIP (ADIP), and a Mixing Module (MM). The EDIP and ADIP module

TABLE IV
NONLINEAR UNMIXING PERFORMANCE COMPARISON OF VARIOUS METHODS FOR DIFFERENT PURITY LEVEL ρ. THE BEST RESULTS ARE IN BOLD. THE SECOND BEST ARE DENOTED WITH A STAR.

method	$\rho = 0.8$			$\rho = 0.9$			$\rho = 1.0$			time (:s)
	RMSE	AAD	SAD	RMSE	AAD	SAD	RMSE	AAD	SAD	
rNMF	0.3042	50.732	23.3305	0.3353	52.6879	22.4824	0.3665	54.2828	22.4233	3.1484
SiVM+FCLS	0.1002	17.9685	6.3412	0.0465	6.7308	3.4658	0.0192	2.4020	0.6095	10.9354
HyperCSI	0.0302⋆	4.6126⋆	1.5008⋆	0.0237	3.2769	1.1230⋆	0.0176	2.1633	0.7855	0.1551
HiSun	0.0405	5.6666	5.4575	0.0212⋆	2.5992⋆	2.6349	0.0188	2.0754	0.6695	9.5697
EDAA	0.0698	11.7178	3.8387	0.0406	5.4092	1.6947	0.0390	4.3166	**0.2894**	12.3635
MNN-BU2	0.1451	29.0092	12.0890	0.0736	9.6026	5.5778	0.1391	22.4132	3.8198	0.1872
U-ADMM-BUNet-II	0.0543	9.1804	2.6652	0.0550	7.9753	1.7887	0.0312	3.8843	0.9381	0.2729
EGU-Net-ss	0.1022	15.3104	4.4069	0.0499	5.9326	2.4708	0.0164⋆	1.8787⋆	0.625	0.3789
UnDIP	0.1496	28.7185	7.1064	0.0479	6.9371	3.4658	0.0238	3.1231	0.6095	0.0099
MiSiCNet	0.0988	15.7423	5.7003	0.0658	8.5409	4.4252	0.0312	3.6481	2.8217	**0.0026**
CNNAEU	0.3600	66.6197	16.2333	0.3972	69.6882	16.9117	0.4428	72.7636	16.5118	0.0077
NL-BUDDIP	**0.026**	**3.9959**	**0.8752**	**0.0182**	**2.2613**	**0.5897**	**0.0121**	**1.1652**	0.4202⋆	0.0039⋆

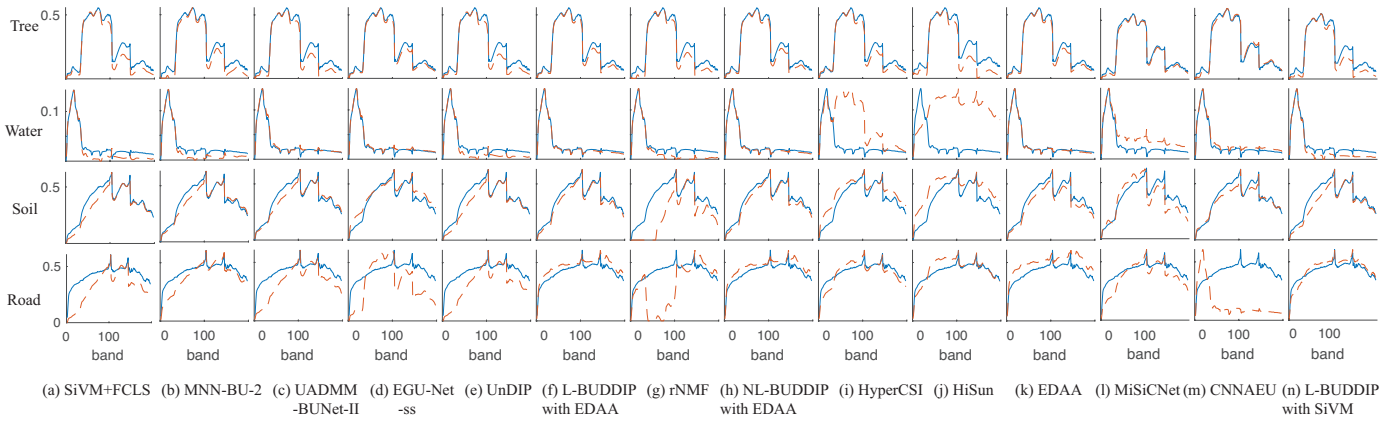


Fig. 11. Endmembers estimated by different methods on Jasper Ridge dataset. Blue solid lines indicate the true value, while red dot lines indicate the scaled estimated value. From top to bottom: Tree, Water, Soil, and Road. (a) SiVM+FCLS. (b) MNN-BU-2. (c) U-ADMM-BUNet-II. (d) EGU-Net-ss. (e) UnDIP. (f) L-BUDDIP with EDAA. (g) rNMF. (h) NL-BUDDIP with EDAA. (i) HyperCSI. (j) HiSun. (k) EDAA. (l) MiSICNet. (m) CNNAEU. (n) L-BUDDIP with SiVM+FCLS.

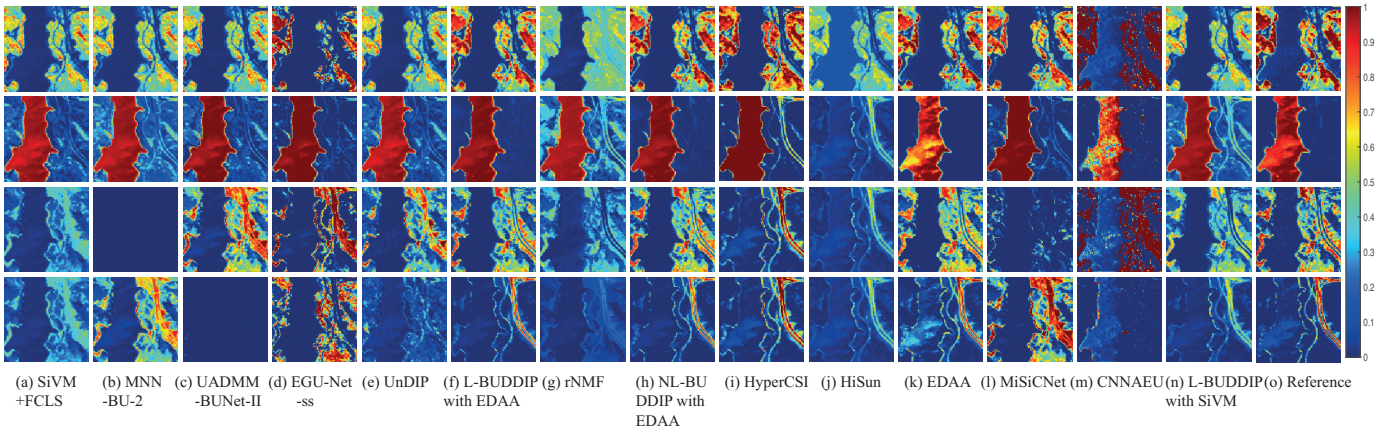


Fig. 12. Abundance maps estimated by different methods on Jasper Ridge dataset. From top to bottom: Tree, Water, Soil, and Road. (a) SiVM+FCLS. (b) MNN-BU-2. (c) U-ADMM-BUNet-II. (d) EGU-Net-ss. (e) UnDIP. (f) L-BUDDIP with EDAA. (g) rNMF. (h) NL-BUDDIP with EDAA. (i) HyperCSI. (j) HiSun. (k) EDAA. (l) MiSICNet. (m) CNNAEU. (n) L-BUDDIP with SiVM+FCLS. (o) Reference.

would output estimations for endmembers and abundances, respectively, whereas the MM, based on the unmixing model, would yield a reconstruction of the observed HSI reflectances. Different from the general noise input used in the DIP

TABLE V
MEAN AND STANDARD DEVIATION OF ABUNDANCE RMSE, AAD (IN DEGREES), ENDMEMBER SAD (IN DEGREES) BY DIFFERENT METHODS ON JASPER RIDGE. THE BEST RESULTS ARE IN BOLD. THE SECOND BEST ARE DENOTED WITH A STAR.

Methods	RMSE	AAD	SAD of Tree	SAD of Water	SAD of Soil	SAD of Road	averaged SAD
HyperCSI	0.2005±0.0000	20.7649±0.0000	5.4103±0.0000	44.3726±0.0000	7.5442±0.0000	6.6429±0.0000	15.9925±0.0000
HiSun	0.3152±0.0000	46.0728±0.0000	14.4662±0.0000	54.6580±0.0000	10.3029±0.0000	5.8153±0.0000	21.3106±0.0000
SiVM+FCLS	0.1480±0.0000	20.7198±0.0000	8.5545±0.0000	14.4877±0.0000	6.5558±0.0000	15.7991±0.0000	11.3493±0.0000
MNN-BU-2	0.2154±0.0007	33.1936±0.0927	7.9771±0.0046	14.4324±0.0117	5.3219±0.0367	8.6187±0.0020	9.0875±0.0117
U-ADMM-Net-II	0.1332±0.0022	17.6087±0.3170	8.6635±0.0160	4.0919±0.1313	5.6216±0.0113	13.6586±0.2183	8.0089±0.0417
EGU-Net-ss	0.2110±0.0019	30.7221±0.2117	2.7626±0.1517⋆	4.5969±0.5298	5.6690±0.0646	22.0731±0.3154	8.7754±0.0242
UnDIP	0.1748±0.0252	25.3249±4.4570	8.5545±0.0000	14.4877±0.0000	6.5558±0.0000	15.7991±0.0000	11.3493±0.0000
L-BUDDIP (EDAA)	0.0445±0.0005⋆	5.2348±0.1152⋆	4.2622±0.0335	2.9074±0.0209	2.7430±0.0206⋆	**2.8273±0.0463**	**3.1850±0.0045**
rNMF	0.1750±0.0002	25.2385±0.0233	8.7639±0.0098	16.1319±0.0122	25.8640±0.0630	32.0590±0.0210	20.7047±0.0148
NL-BUDDIP	**0.0434±0.0005**	**5.1956±0.0722**	4.3011±0.0489	2.8836±0.0352⋆	2.7488±0.0228	2.8532±0.0681⋆	3.1966±0.0078⋆
MiSiCNet	0.1894±0.0000	27.4191±0.0015	**2.4860±0.0010**	16.5820±0.0069	10.0646±0.0007	9.3605±0.0018	9.6233±0.0015
CNNAEU	0.2726±0.0860	36.6164±12.9452	3.8993±0.5724	4.3835±1.2212	5.0614±1.3909	31.3367±16.9217	11.1702±4.0669
EDAA	0.0583±0.0000	7.6572±0.0000	4.2205±0.0000	**2.8038±0.0000**	**2.7262±0.0000**	3.1506±0.0000	3.2253±0.0000
L-BUDDIP (SiVM+FCLS)	0.1238±0.0003	16.2442±0.0447	8.7884±0.0139	13.2621±0.0064	6.4180±0.0212	4.1317±0.0013	8.1501±0.0036

technique, we propose to use the estimations from existing unmixing methods as the input, based upon which, we design a more efficient DIP network structure with less learnable parameters. In order to generate meaningful unmixing results and further improve the unmixing performance, we propose a new composite loss function that is applicable in both linear and nonlinear unmixing cases. For the nonlinear case, we also propose an adaptive loss weight strategy to further improve the performance. Extensive experiments on both linear and nonlinear unmixing cases show that the proposed method can deliver better unmixing performance than the state-of-art unmixing methods such as SiVM+FCLS, rNMF, MNN-BU, U-ADMM-BUNet, EGU-Net, and UnDIP.

ACKNOWLEDGMENTS

This work was supported by EPSRC grant EP/R032785/1 and UCL Project ARTICT. The work of Chao Zhou was supported in part by China Scholarship Council. For the purpose of open access, the author has applied a Creative Commons Attribution (CC BY) licence to any Author Accepted Manuscript version arising.

REFERENCES

[1] 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.

[2] 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.

[3] G. Camps and G. Camps-Valls, *Remote Sensing Image Processing*, ser. Synthesis lectures on image, video, and multimedia processing. Morgan & Claypool Publishers, 2011. [Online]. Available: https://books.google.co.uk/books?id=kjMWGbygJIQC

[4] N. Keshava and J. F. Mustard, "Spectral unmixing," *IEEE Geosci. Remote Sens. Mag.*, vol. 19, no. 1, pp. 44–57, 2002.

[5] Z. Wang, B. Chen, H. Zhang, and H. Liu, "Unsupervised hyperspectral and multispectral images fusion based on nonlinear variational probabilistic generative model," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 33, no. 2, pp. 721–735, 2020.

[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, Apr. 2005.

[7] 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.

[8] 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.

[9] J. M. Bioucas-Dias and M. A. T. Figueiredo, "Alternating direction algorithms for constrained sparse regression: Application to hyperspectral unmixing," in *2010 2nd Workshop on Hyperspectral Image and Signal Processing: Evolution in Remote Sensing*, 2010, pp. 1–4.

[10] M. Iordache, J. M. Bioucas-Dias, and A. Plaza, "Collaborative sparse regression for hyperspectral unmixing," *IEEE Trans. Geosci. Remote Sens.*, vol. 52, no. 1, pp. 341–354, 2014.

[11] A. Halimi, Y. Altmann, N. Dobigeon, and J.-Y. Tourneret, "Nonlinear unmixing of hyperspectral images using a generalized bilinear model," *IEEE Trans. Geosci. Remote Sens.*, vol. 49, no. 11, pp. 4153–4162, 2011.

[12] Y. Ma, C. Li, X. Mei, C. Liu, and J. Ma, "Robust sparse hyperspectral unmixing with $\ell_{2,1}$ norm," *IEEE Trans. Geosci. Remote Sens.*, vol. 55, no. 3, pp. 1227–1239, 2017.

[13] 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.

[14] C. Févotte and N. Dobigeon, "Nonlinear hyperspectral unmixing with

robust nonnegative matrix factorization," *IEEE Trans. Image Process.*, vol. 24, no. 12, pp. 4810–4819, 2015.

[15] 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.

[16] 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.

[17] A. Zouaoui, G. Muhawenayo, B. Rasti, J. Chanussot, and J. Mairal, "Entropic descent archetypal analysis for blind hyperspectral unmixing," *arXiv preprint arXiv:2209.11002*, 2022.

[18] Z. Yang, Y. Xiang, S. Xie, S. Ding, and Y. Rong, "Nonnegative blind source separation by sparse component analysis based on determinant measure," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 23, no. 10, pp. 1601–1610, 2012.

[19] S. E. Selvan, U. Amato, K. A. Gallivan, C. Qi, M. F. Carfora, M. Larobina, and B. Alfano, "Descent algorithms on oblique manifold for source-adaptive ica contrast," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 23, no. 12, pp. 1930–1947, 2012.

[20] A. Chattopadhyay, S. E. Selvan, and U. Amato, "A derivative-free riemannian powell's method, minimizing hartley-entropy-based ica contrast," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 27, no. 9, pp. 1983–1990, 2015.

[21] Z. Yang, Y. Xiang, Y. Rong, and K. Xie, "A convex geometry-based blind source separation method for separating nonnegative sources," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 26, no. 8, pp. 1635–1644, 2014.

[22] C.-H. Lin and J. M. Bioucas-Dias, "Nonnegative blind source separation for ill-conditioned mixtures via john ellipsoid," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 32, no. 5, pp. 2209–2223, 2020.

[23] C.-H. Lin, C.-Y. Chi, Y.-H. Wang, and T.-H. Chan, "A fast hyperplane-based minimum-volume enclosing simplex algorithm for blind hyperspectral unmixing," *IEEE Trans. Signal Process.*, vol. 64, no. 8, pp. 1946–1961, 2015.

[24] M. Zhao, M. Wang, J. Chen, and S. Rahardja, "Hyperspectral unmixing for additive nonlinear models with a 3-d-cnn autoencoder network," *IEEE Trans. Geosci. Remote Sens.*, pp. 1–15, 2021.

[25] 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.

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

[27] C. Zhou and M. R. Rodrigues, "An admm based network for hyperspectral unmixing tasks," in *ICASSP 2021 - 2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2021, pp. 1870–1874.

[28] 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.

[29] 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.

[30] 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.

[31] 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.

[32] 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.

[33] 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.

[34] 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.

[35] V. Lempitsky, A. Vedaldi, and D. Ulyanov, "Deep image prior," in *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2018, pp. 9446–9454.

[36] 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.

[37] C. Zhou and M. R. Rodrigues, "Admm-based hyperspectral unmixing networks for abundance and endmember estimation," *IEEE Trans. Geosci. Remote Sens.*, 2021, early Access.

[38] J. Sigurdsson, M. O. Ulfarsson, and J. R. Sveinsson, "Blind hyperspectral unmixing using total variation and ℓ_q sparse regularization," *IEEE Trans. Geosci. Remote Sens.*, vol. 54, no. 11, pp. 6371–6384, 2016.

[39] N. Dobigeon, J.-Y. Tourneret, C. Richard, J. C. M. Bermudez, S. McLaughlin, and A. O. Hero, "Nonlinear unmixing of hyperspectral images: Models and algorithms," *IEEE Signal Process. Mag.*, vol. 31, no. 1, pp. 82–94, 2014.

[40] M. Zhao, M. Wang, J. Chen, and S. Rahardja, "Hyperspectral unmixing via deep autoencoder networks for a generalized linear-mixture/nonlinear-fluctuation model," *arXiv preprint arXiv:1904.13017*, 2019.

[41] S. V. Venkatakrishnan, C. A. Bouman, and B. Wohlberg, "Plug-and-play priors for model based reconstruction," in *2013 IEEE Global Conference on Signal and Information Processing*. IEEE, 2013, pp. 945–948.

[42] P. Isola, J.-Y. Zhu, T. Zhou, and A. A. Efros, "Image-to-image translation with conditional adversarial networks," in *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017, pp. 5967–5976.

[43] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016, pp. 770–778.

[44] Y. Gandelsman, A. Shocher, and M. Irani, "" double-dip": Unsupervised image decomposition via coupled deep-image-priors," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 11 026–11 035.

[45] W. Fan, B. Hu, J. Miller, and M. Li, "Comparative study between a new nonlinear model and common linear model for analysing laboratory simulated-forest hyperspectral data," *International Journal of Remote Sensing*, vol. 30, no. 11, pp. 2951–2962, 2009.

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

[47] "Usgs library." [Online]. Available: \url{https://www.usgs.gov/labs/spec-lab}

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

[49] A. Ambikapathi, T.-H. Chan, W.-K. Ma, and C.-Y. Chi, "Chance-constrained robust minimum-volume enclosing simplex algorithm for hyperspectral unmixing," *IEEE Trans. Geosci. Remote Sens.*, vol. 49, no. 11, pp. 4194–4209, 2011.

[50] C.-H. Lin, W.-K. Ma, W.-C. Li, C.-Y. Chi, and A. Ambikapathi, "Identifiability of the simplex volume minimization criterion for blind hyperspectral unmixing: The no-pure-pixel case," *IEEE Trans. Geosci. Remote Sens.*, vol. 53, no. 10, pp. 5530–5546, 2015.

[51] 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.

[52] B. Palsson, M. O. Ulfarsson, and J. R. Sveinsson, "Convolutional autoencoder for spectral–spatial hyperspectral unmixing," *IEEE Trans. Geosci. Remote Sens.*, vol. 59, no. 1, pp. 535–549, 2020.

[53] B. Rasti, B. Koirala, P. Scheunders, and J. Chanussot, "Misicnet: Minimum simplex convolutional network for deep hyperspectral unmixing," *IEEE Trans. Geosci. Remote Sens.*, vol. 60, pp. 1–15, 2022.