AE-RED: A Hyperspectral Unmixing Framework Powered by Deep Autoencoder and Regularization by Denoising

Min Zhao, Jie Chen, Nicolas Dobigeon

AE-RED: A Hyperspectral Unmixing Framework Powered by Deep Autoencoder and Regularization by Denoising

Min Zhao, *Student Member, IEEE,* Jie Chen, *Senior Member, IEEE* and Nicolas Dobigeon, *Senior Member, IEEE*

***Abstract*—Spectral unmixing has been extensively studied with a variety of methods and used in many applications. Recently, data-driven techniques with deep learning methods have obtained great attention to spectral unmixing for its superior learning ability to automatically learn the structure information. In particular, autoencoder based architectures are elaborately designed to solve blind unmixing and model complex nonlinear mixtures. Nevertheless, these methods perform unmixing task as black-boxes and lack interpretability. On the other hand, conventional unmixing methods carefully design the regularizer to add explicit information, in which algorithms such as plug-and-play (PnP) strategies utilize off-the-shelf denoisers to plug powerful priors. In this paper, we propose a generic unmixing framework to integrate the autoencoder network with regularization by denoising (RED), named AE-RED. More specially, we decompose the unmixing optimized problem into two subproblems. The first one is solved using deep autoencoders to implicitly regularize the estimates and model the mixture mechanism. The second one leverages the denoiser to bring in the explicit information. In this way, both the characteristics of the deep autoencoder based unmixing methods and priors provided by denoisers are merged into our well-designed framework to enhance the unmixing performance. Experiment results on both synthetic and real data sets show the superiority of our proposed framework compared with state-of-the-art unmixing approaches.**

***Index Terms*—Hyperspectral unmixing, deep learning, autoencoder, plug-and-play, image denoising, RED.**

I. Introduction

Hyperspectral imaging has been a widely explored imaging technique during recent years and is still receiving a growing attention in various applicative fields [1], [2]. Benefiting from a rich spectral information, hyperspectral images enable the analysis of fine materials in the observed scenes to tackle various challenging tasks such as target detection and classification [3], [4]. However, due to the limitations of the imaging acquisition devices, there is an insurmountable trade-off between the collected spectral and spatial information, which limits the spatial resolution of the hyperspectral sensors. As a consequence, a pixel observed by a hyperspectral sensor may encompass several materials. In particular when observing complex scenes, the spectrum is assumed to be a mixture of several elementary spectral signatures. To overcome this limitation, hyperspectral unmixing (HU) aims at decomposing the ith observed pixel spectrum $\mathbf{y}_i \in \mathbb{R}^B$ into a set of R spectral signatures of so-called endmembers collected in the matrix $\mathbf{S} = [\mathbf{s}_1, \ldots, \mathbf{s}_R] \in \mathbb{R}^{B\times R}$ and their associated fractions or abundances $\mathbf{a}_i \in \mathbb{R}^R$ [5]–[7]. For the sake of physical interpretability, the abundances are subject to two constraints, namely abundance sum-to-one constraint (ASC), $\mathbf{1}_R^\top \mathbf{a}_i = 1$, and abundance nonnegativity constraint (ANC), $\mathbf{a}_i \geq \mathbf{0}$. The endmember spectral signatures are constrained to be nonnegative (ENC), $\mathbf{S} \geq \mathbf{0}$.

TABLE I
Notations.

x, X	scalar
$\mathbf{x}$	column vector
$\mathbf{X}$	matrix
B	number of spectral bands
N	number of pixels
R	number of endmembers
$\mathbf{y}_i \in \mathbb{R}^B$	spectrum of the ith observed pixel
$\mathbf{Y} \in \mathbb{R}^{B\times N}$	an observed hyperspectral image
$\mathbf{a}_i \in \mathbb{R}^R$	abundance vector of the ith pixel
$\mathbf{A} \in \mathbb{R}^{R\times N}$	abundance matrix of all pixels
$\mathbf{S} \in \mathbb{R}^{B\times R}$	endmember matrix with R spectral signatures
$\mathbf{1}$	all one vector or matrix
$\mathbf{0}$	all zero vector or matrix
$\cdot \geq \cdot$	elementwise inequality between vectors or matrices

Many methods have been proposed in the literature to address the HU problem [8]–[12]. Considering a set of N observed pixels $\mathbf{Y} = [\mathbf{y}_1, \ldots, \mathbf{y}_N] \in \mathbb{R}^{B\times N}$ sharing the same endmembers, the canonical formulation of HU is written as an optimization problem, which aims at estimating the endmembers $\mathbf{S}$ and the abundances $\mathbf{A}$ jointly, i.e.,

$$\begin{aligned} \min_{\mathbf{S},\mathbf{A}} \quad & \sum_{i=1}^{N} \mathcal{D}\left[\mathbf{y}_i \middle|\middle| \mathcal{M}(\mathbf{S}, \mathbf{a}_i)\right] + \mathcal{R}(\mathbf{S}, \mathbf{A}) \\ \text{s.t.} \quad & \mathbf{1}_R^\top \mathbf{A} = \mathbf{1}_N^\top, \ \mathbf{A} \geq \mathbf{0}, \text{ and } \mathbf{S} \geq \mathbf{0} \end{aligned} \tag{1}$$

where

- $\mathcal{D}[\cdot, \cdot]$ stands for a discrepancy measure (e.g., divergence),

M. Zhao and J. Chen are with School of Marine Science and Technology, Northwestern Polytechnical University, Xi'an 710072, China (e-mail: minzhao@mail.nwpu.edu.cn; dr.jie.chen@ieee.org).

N. Dobigeon is with University of Toulouse, IRIT/INP-ENSEEIHT, CNRS, 2 rue Charles Camichel, BP 7122, 31071 Toulouse Cedex 7, France (e-mail: Nicolas.Dobigeon@enseeiht.fr).

The work of Jie Chen was supported in part by Shenzhen Science and Technology Program under Grant JCYJ20220530161606014 and Grant JCYJ20230807145600001, in part by the Department of Natural Resources of Guangdong Province under Grant GDNRC[2023]47, and in part by the TCL Science and Technology Innovation Fund. The work of Nicolas Dobigeon was supported by the Artificial Natural Intelligence Toulouse Institute (ANITI) under Grant ANR-19-PI3A-0004.

- $\mathcal{M}_{\boldsymbol{\Theta}}(\cdot,\cdot)$ describes the mixture model which relates the endmembers and the abundances to the measurements,
- $\mathcal{R}(\cdot,\cdot)$ acts as a regularization term that encodes prior information regarding the endmembers $\mathbf{S}$ and the abundances $\mathbf{A}$.

The regularization $\mathcal{R}(\cdot,\cdot)$ is often designed to be separable with respect to the abundances and endmembers,

$$\mathcal{R}(\mathbf{S},\mathbf{A}) = \mathcal{R}_{\mathrm{e}}(\mathbf{S}) + \mathcal{R}_{\mathrm{a}}(\mathbf{A}), \tag{2}$$

where the endmember and abundance prior information is encoded in $\mathcal{R}_{\mathrm{e}}(\cdot)$ and $\mathcal{R}_{\mathrm{a}}(\cdot)$, respectively. For instance, geometry-based penalizations, such as minimum volume [13], are often chosen as endmember regularizers. Sparsity-based [14], low-rankness [15] or spatial regularizers, such as total variation (TV) [16], are usually utilized to promote expected properties of the abundances. This work specifically focuses on the design of the abundance regularization.

As for the mixing process, typical methods rely on an explicit mathematical expression for $\mathcal{M}(\cdot,\cdot)$ to describe the mixture mechanism. For example, the linear mixing model (LMM) is by far the most used in the literature since it provides a generally admissible first-order approximation of the mixing processes, and assumes that the incident light comes in and only reflects once on the ground before reaching the hyperspectral sensor. Besides, bilinear models consider second-order reflections, for instance in the case of multiple vegetation layers [1], [17]. These explicit models are usually designed by describing the path of the light, along with its scattering and the interaction mechanisms among the materials. They are thus generally referred to as physics-based models. However, in some acquisition scenarios, they may fail to accurately account for real complex scenes. Data-driven methods have been thus proposed to implicitly learn the mixing mechanism from the observed data. Nevertheless, if not carefully designed, a data-driven method may overlook the physical mixing process and require abundant training data [18].

A. Motivation

Numerous methods cope with the HU problem by carefully designing the data-fitting and regularization terms [19], [20]. To reduce the computational complexity, most HU methods are based on the LMM. It may be not sufficient to account for spectral variability and endmember nonlinearity. On the other hand, designing a relevant regularizer is not always trivial and is generally driven by an empirical yet limited knowledge. For these reasons, research works have been devoted to the design of deep learning based HU approaches. Among them, autoencoders (AEs) become increasingly popular for unsupervised HU, which exhibit several advantages: *i)* they can embed a physical-based mixing model into the structure of the decoder, *ii)* they implicitly incorporate data-driven image priors and *iii)* the unmixing procedure can benefit from powerful optimizers, such as Adam [21] and SGD [22]. However, these deep architectures behave as black boxes, and the results lack interpretation. Motivated by these findings, this paper attempts to answer the following question: is it possible to design an unsupervised HU framework which combines the advantages of AE-based unmixing while leveraging on explicit priors?

B. Contributions

This paper derives a novel HU framework which answers this question affirmatively. More precisely, it introduces an AE-based unmixing strategy while incorporating an explicit regularization of the form of RED. To solve the resulting optimization problem, an alternating direction method of multiplier (ADMM) is implemented with the great advantages of decomposing the initial problem into several simpler subproblems. One of these subproblems can be interpretated as a standard training task associated with an AE. Another is a standard denoising problem. The main advantages of the proposed frameworks are threefold:

- This framework combines the deep AE with RED priors for unsupervised HU. By leveraging the benefits of these two ingredients, the framework provides accurate unmixing results.
- The optimization procedure splits the unmixing task into two main subtasks. The first subtask involves training an AE to learn the mixing process and estimate a latent representation of the image as abundance maps and the weights of a specific layer as endmembers. In the second subtask, a denoising step is applied to improve the estimation of the latent representation.
- The proposed framework is highly versatile and can accommodate various architectures for the encoder, and the decoder can be tailored to mimic any physics-based mixing model, such as the LMM, nonlinear mixing models, and spectral variability-aware mixing models.

This paper is organized as follows. Section II provides a concise overview of related HU algorithms, with a particular focus on the design of regularizations and AE-based unmixing methods. It also describes some technical ingredients necessary to build the proposed framework. In Section III, the proposed generic framework is derived, and details about particular instances of this framework are given. Section IV reports the results obtained from extensive experiments conducted on synthetic and real datasets to demonstrate the superiority of the proposed framework. Finally, Section V concludes the paper.

II. Related works and background

This section first draws brief literature overviews on two aspects related to this work, namely regularization designs in HU and AE-based unmixing. Then it provides the technical background on which the proposed framework is built.

A. Related works

1) Regularization design: Efficient algorithms for HU often require effective regularizations that incorporate prior knowledge about the abundances and constrain the range of the admissible solutions. Traditional model-based regularizations can be roughly divided into two main families. Some promote

expected spatial properties of the abundance maps. Others exploit the fact that only a few materials generally contribute to the mixture in a given pixel to derive sparsity-based regularizers. In [16], TV is combined with an ℓ_1-norm regularization to simultaneously promote similarity between neighboring pixels while ensuring sparse representations of the measured spectral signatures. Since the ℓ_1-norm is inconsistent with the ASC, ℓ_p-norms with $0 < p < 1$ [23] and reweighting strategies [24] have been also considered as alternatives to promote sparse estimates. In [25], a weighted average is applied to all pixels to exploit non-local spatial information. Sparsity-based spatial regularizations informed by image segmentation such as SLIC have been designed in [26] and [27]. In [28], a cofactorization model is used to jointly exploit spectral and spatial information, while the work of [29] introduces an adaptive graph to automatically determine the best neighbor points of pixels and assign corresponding weights. However, these traditional model-based regularizations are generally motivated by empirical choices, which may hardly capture the complexities of spatial contents inherent to most remote sensing images. Moreover, they all require to derive and implement dedicated optimization algorithms which can be computationally intensive when handling large images.

More recently, the idea of PnP has been introduced to exploit the intrinsic properties of hyperspectral images. These methods use generic denoisers that act as implicit or explicit regularizers. In [30], an HU method based on ADMM is introduced to plug denoising priors. The work of [31] proposes a nonlinear unmixing method with prior information provided by denoisers. However, these methods have been designed to handle only one fixed specific mixing model. Generalizing these methods to handle other mixing models would require to completely redesign the overall resolution algorithmic scheme. Conversely, the work reported in this paper introduces a general framework whose AE can learn the mixing process and leverages an RED approach, which has been shown to outperform PnP.

2) Deep AE-based unmixing methods: Elegant neural network structures have been proposed to formulate the HU task as a simple training process. Early works used fully connected layers to design the network, such as [12] and [32]. However, these networks process the pixels independently and ignore the spatial correlation intrinsic to the image. To overcome this limitation, some AE-based methods include spatial regularizations, such as TV, in the loss function [33]. More recently, convolutional neural networks (CNNs) have been used to perform HU and have shown promising performance. CNNs convolve the input data with filter kernels to capture spatial information [10], [34]. Recurrent neural networks (RNNs), which embed memory cells, implement a sequential process with hidden states that depend on the previous states [33]. Hyperspectral images are often corrupted by noise or outliers, which can dramatically decrease the unmixing performance. To address this issue, denoising-oriented architectures have been proposed [32]. Some works have also proposed variants of encoders. In [35], a dual-branch AE network is designed to leverage multiscale spatial contextual information.

Most AE-based HU methods use a fully connected linear layer in the decoder to mimic the LMM. However, considering the physical interactions between multiple materials and the ability of deep networks to model nonlinear processes, some works have focused on the design of structured decoders to ensure the interpretability of the nonlinear model inherent in the mixing process [33], [34], [36], [37]. The work of [36] introduces a nonlinear decoder. The decoder contains two parts: the linear part is considered as a rough approximation of the mixture and then it is complemented by the nonlinear part. However, this post-nonlinear model-based decoder may not be sufficient to represent complex nonlinear cases. Some works have investigated the nonlinear fluctuation part of the decoder [33], [34], [37]. For example, the method in [37] designs a special layer to capture the second-order interaction, similar to the bilinear models [38]. Moreover, spectral variability can also be addressed by using deep generative decoders [39], [40].

Recently, deep unfolding techniques have been used to unroll iterative unmixing algorithms into deep networks. This approach allows physically grounded and interpretable findings to be invoked when designing the network layers. They can also avoid the painful tuning of some hyperparameters by learning them from the data [41]. In [42], an iterative shrinkage-thresholding algorithm (ISTA)-inspired network layer is applied to build an AE-based unmixing architecture. The work of [43] unrolls a sparse non-negative matrix factorization (NMF)-based algorithm with an ℓ_p-norm regularizer to integrate prior knowledge into the unmixing network. An ADMM solver with a sparse regularizer is also unrolled to build an AE-like unmixing architecture. However, these methods do not utilize spatial consistency information in the design of the network, which may limit their unmixing performance.

B. Background

1) Autoencoder-based unmixing: As highlighted in the previous section, AEs have demonstrated to be a powerful tool to conduct unsupervised unmixing. An AE typically consists of an encoder and a decoder. The encoder, represented by $\mathsf{E}_{\boldsymbol{\Theta}_\mathsf{E}}(\cdot)$, aims at learning a nonlinear mapping from input data, denoted as $\mathbf{w}_i$, to their corresponding latent representations, denoted as $\mathbf{v}_i$. This can be expressed as follows:

$$\mathbf{v}_i = \mathsf{E}_{\boldsymbol{\Theta}_\mathsf{E}}(\mathbf{w}_i), \tag{3}$$

where $\boldsymbol{\Theta}_\mathsf{E}$ gathers all parameters of the encoder. The input $\mathbf{W} = [\mathbf{w}_1, \ldots, \mathbf{w}_N]$ depends on the architecture chosen for the encoder network. For instance, when dealing with the specific task of HU, the input can be chosen as the image pixels $\mathbf{Y} = [\mathbf{y}_1, \ldots, \mathbf{y}_N]$ or random noise realizations $\mathbf{Z} = [\mathbf{z}_1, \ldots, \mathbf{z}_N]$ with $\mathbf{z}_i \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$. The decoder, denoted by $\mathsf{D}_{\boldsymbol{\Theta}_\mathsf{D}}(\cdot)$, is responsible for reconstructing the data, or at least an approximation $\hat{\mathbf{y}}_i$, from the latent feature $\mathbf{v}_i$ provided by the encoder. This can be expressed as follows:

$$\hat{\mathbf{y}}_i = \mathsf{D}_{\boldsymbol{\Theta}_\mathsf{D}}(\mathbf{v}_i), \tag{4}$$

where $\boldsymbol{\Theta}_\mathsf{D}$ parameterizes the decoder. Under this paradigm, adjusting the encoder and decoder parameters $\boldsymbol{\Theta}_\mathsf{E}$ and $\boldsymbol{\Theta}_\mathsf{D}$ is generally achieved by minimizing the empirical expectation of

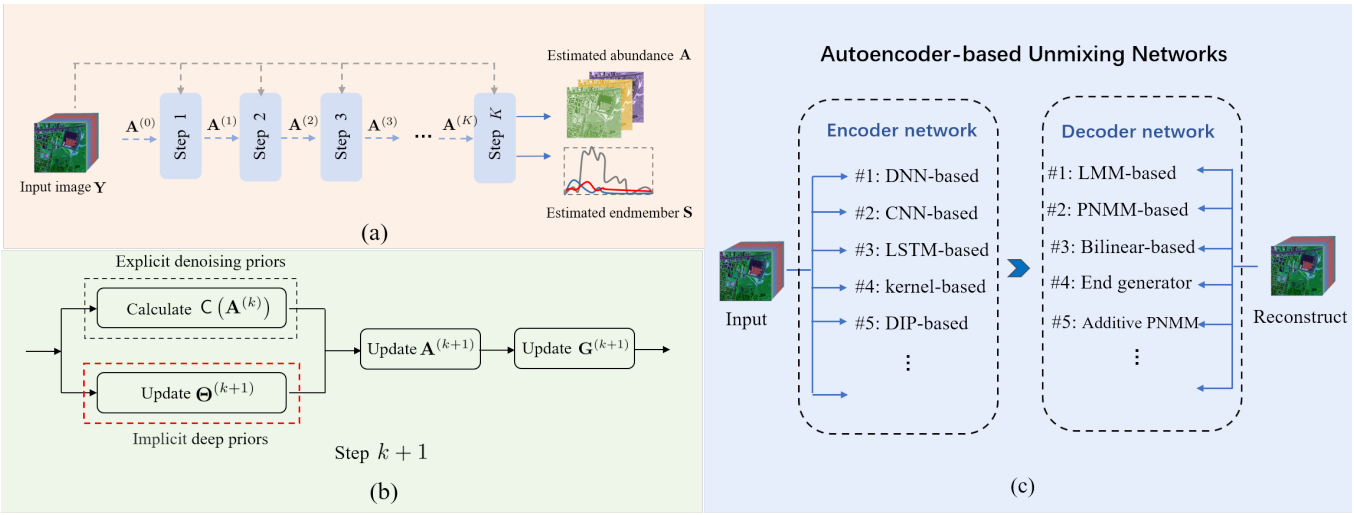

Fig. 1. Framework of the proposed AE-RED. (a) The scheme of the proposed framework. (b) Flowchart of the $(k+1)$th ADMM step: the denoising operator is applied in parallel to the update of $\boldsymbol{\Theta}$ to speed up calculations. (c) An overview and some instances of AE-based unmixing networks, where the encoder embeds deep priors for abundance estimation, and the decoder can model the mixture mechanism and extract the endmembers. The choice of the encoder and decoder is let to the end-user.

a discrepancy measure between the input data $\mathbf{y}_1, \ldots, \mathbf{y}_N$ and their corresponding approximation $\hat{\mathbf{y}}_1, \ldots, \hat{\mathbf{y}}_N$, i.e.,

$$\mathcal{L}(\boldsymbol{\Theta}_{\mathsf{E}}, \boldsymbol{\Theta}_{\mathsf{D}}) = \frac{1}{N} \sum_{i=1}^{N} \mathcal{D}\left[\mathbf{y}_i || \hat{\mathbf{y}}_i\right] \tag{5}$$

with $\hat{\mathbf{y}}_i = \mathsf{D}_{\boldsymbol{\Theta}_{\mathsf{D}}}\left(\mathsf{E}_{\boldsymbol{\Theta}_{\mathsf{E}}}\left(\mathbf{w}_i\right)\right)$. This reconstruction loss function can be complemented with additional terms to account for any desired properties regarding the network parameters and the latent representation.

Drawing a straightforward analogy with the problem (1), AE-based unmixing frameworks generally assume that the latent variable $\mathbf{V} = [\mathbf{v}_1, \ldots, \mathbf{v}_N]$ is an estimate of the abundance matrix $\mathbf{A}$. The encoder can thus be considered as a regularization for abundance estimation. Its architecture should be chosen to be able to extract key spatial features from the input data. Several choices are possible and will be discussed as archetypal examples later in Section III-B. The decoder can then be designed to mimic the mixing process $\mathcal{M}(\cdot, \cdot)$ in (1). The endmember signatures to be recovered are part of the decoder parameters, i.e., $\boldsymbol{\Theta}_{\mathsf{D}} = \left\{\tilde{\boldsymbol{\Theta}}_{\mathsf{D}}, \mathbf{S}\right\}$ where $\tilde{\boldsymbol{\Theta}}_{\mathsf{D}}$ are intrinsic network parameters. For instance, when the decoder is designed according to a physics-based nonlinear mixing model prescribed beforehand, $\tilde{\boldsymbol{\Theta}}_{\mathsf{D}}$ gathers the nonlinearity parameters. In the simplistic assumption of the LMM, the decoder does not depend on any additional intrinsic parameters and $\boldsymbol{\Theta}_{\mathsf{D}} = \mathbf{S}$.

2) Regularization by denoising priors: Various model-free regularizers have been considered to design the term $\mathcal{R}_{\mathrm{a}}(\cdot)$. Among them, a powerful strategy consists in resorting to off-the-shelf denoisers to implicitly or explicitly regularize inverse problems. The first approach, referred to as PnP, is a flexible and generic framework that naturally emerges when resorting to splitting-based optimization procedures, such as half-quadratic splitting or ADMM. After augmenting the initial optimization problem with auxiliary variables, the resolution algorithmic scheme can be decomposed into several steps. The only step which depends on the regularization boils down to performing a denoising task, which can be achieved by any denoiser. This strategy has been effectively used when tackling many imaging inverse problems, such as super-resolution and inpainting [44], [45]. Under this PnP paradigm, the regularization never needs to be specified and is only implicitly defined through the use of the denoiser. More recently, a second approach, referred to as RED, also leverages the genericity of denoising but with an explicit image-adaptive Laplacian-based regularization defined as

$$\mathcal{R}_{\mathrm{a}}(\mathbf{A}) = \frac{1}{2}\mathbf{A}^{\top}\left(\mathbf{A} - \mathsf{C}\left(\mathbf{A}\right)\right), \tag{6}$$

where $\mathsf{C}(\cdot)$ is a denoiser [46]. This framework has demonstrated superior performance with respect to the original PnP approach.

Both PnP and RED share some similarities: *i)* they allow an inverse problem to be regularized without resorting to an image-flavored model-based penalization, *ii)* they finally rely on the use of an off-the-shelf denoiser whose choice can be let to the end-user. However, their respective foundations are significantly different. Within the PnP framework, the regularization $\mathcal{R}_a(\cdot)$ can be not specified explicitly. Instead, this denoising step implicitly arises in the optimization scheme. Conversely, RED exploits the expected properties of any denoiser to explicitly define the regularization. Indeed, it relies on the inner-product between the solution $\mathbf{A}$ and its post-denoising residual $\mathbf{A} - \mathsf{C}\left(\mathbf{A}\right)$. Interestingly, this definition makes the regularization to be small when the solution or the corresponding residual follow two expected behaviors. First, the regularization is small when the residual as well, i.e., the solution can be considered as a fixed-point of the denoiser (the solution does not need to be denoised further). Second, the regularization is small when the (empirical) cross-correlation of the residual to the image is small, i.e., when the residual

is decorrelated from the noise-free image [47]. Besides, RED exhibits several key advantages. First, similar to PnP, it does not need to prescribe a particular model-based prior of the image. Instead, it only relies on the ability of performing an image denoising task. Second, any existing denoiser available from the literature can be implemented. In particular, it can embed any data-informed denoiser which has been trained on an appropriate training set beforehand. Third, under some reasonable and mild assumptions on $\mathsf{C}(\cdot)$, its derivative with respect to $\mathbf{A}$ is simple and given as the denoising residual, i.e., $\nabla \mathcal{R}_a(\mathbf{A}) = \mathbf{A} - \mathsf{C}(\mathbf{A})$ [46], which avoids differentiating the denoiser function. Finally, for a large class of denoisers, it is a convex function. It can be readily utilized in first-order optimization solvers, e.g., gradient descent and fixed-point strategies.

III. Proposed method

A. Generic framework

The generic unmixing framework proposed in this paper, referred to as AE-RED hereafter, formulates the HU problem as the training of an AE while leveraging the RED paradigm. Adopting a conventional Euclidean divergence for $\mathcal{D}(\cdot,\cdot)$, the HU problem (1) is now specified as

$$\begin{aligned} \min_{\boldsymbol{\Theta}} \; & \|\mathbf{Y} - \mathsf{D}_{\boldsymbol{\Theta}_\mathsf{D}}(\mathsf{E}_{\boldsymbol{\Theta}_\mathsf{E}}(\mathbf{W}))\|_\mathrm{F}^2 \\ & + \lambda \mathsf{E}_{\boldsymbol{\Theta}_\mathsf{E}}(\mathbf{W})^\top (\mathsf{E}_{\boldsymbol{\Theta}_\mathsf{E}}(\mathbf{W}) - \mathsf{C}(\mathsf{E}_{\boldsymbol{\Theta}_\mathsf{E}}(\mathbf{W}))) \\ \text{s.t. } & \mathbf{1}_R^\top \mathsf{E}_{\boldsymbol{\Theta}_\mathsf{E}}(\mathbf{W}) = \mathbf{1}_N^\top,\ \mathsf{E}_{\boldsymbol{\Theta}_\mathsf{E}}(\mathbf{W}) \geq \mathbf{0} \text{ and } \mathbf{S} \geq \mathbf{0} \end{aligned} \tag{7}$$

with $\boldsymbol{\Theta} = \{\boldsymbol{\Theta}_\mathsf{E}, \boldsymbol{\Theta}_\mathsf{D}\}$. As stated in the previous section, the endmembers are part of the set of decoder parameters, and $\mathbf{A} = \mathsf{E}_{\boldsymbol{\Theta}_\mathsf{E}}(\mathbf{W})$. This formulation of the unmixing task leverages a combination of the AE modeling and RED, providing two main benefits. First, the AE is effective in handling the mixture mechanism and learning underlying information. Second, RED provides a flexible and efficient way to encode image priors.

Solving the minimization problem (7) with deep learning-flavored black-box optimizers is challenging if not infeasible, in particular because back-propagating $\boldsymbol{\Theta}_\mathsf{E}$ would require differentiating the denoising function $\mathsf{C}(\cdot)$. For most denoisers, this differentiation is not straightforward and may need a huge amount of computations. However, it is worth noting that one of the great advantages of RED is that its derivative can be directly calculated. To benefit from this property, one simple strategy consists in reintroducing the abundance matrix $\mathbf{A}$ explicitly as an auxiliary variable and then reformulating (7) as a constrained problem

$$\begin{aligned} \min_{\boldsymbol{\Theta},\mathbf{A}} \; & \|\mathbf{Y} - \mathsf{D}_{\boldsymbol{\Theta}_\mathsf{D}}(\mathsf{E}_{\boldsymbol{\Theta}_\mathsf{E}}(\mathbf{W}))\|_\mathrm{F}^2 + \lambda \mathbf{A}^\top (\mathbf{A} - \mathsf{C}(\mathbf{A})) \\ \text{s.t. } & \mathbf{1}_R^\top \mathsf{E}_{\boldsymbol{\Theta}_\mathsf{E}}(\mathbf{W}) = \mathbf{1}_N^\top,\ \mathsf{E}_{\boldsymbol{\Theta}_\mathsf{E}}(\mathbf{W}) \geq \mathbf{0},\ \mathbf{S} \geq \mathbf{0} \\ & \text{and } \mathbf{A} = \mathsf{E}_{\boldsymbol{\Theta}_\mathsf{E}}(\mathbf{W}). \end{aligned} \tag{8}$$

To solve (8), a common yet efficient strategy boils down to splitting the initial problems into several simpler subproblems following an ADMM.

The main steps of the resulting ADMM algorithmic scheme write

$$\begin{aligned} \boldsymbol{\Theta}^{(k+1)} = & \arg\min_{\boldsymbol{\Theta}} \|\mathbf{Y} - \mathsf{D}_{\boldsymbol{\Theta}_\mathsf{D}}(\mathsf{E}_{\boldsymbol{\Theta}_\mathsf{E}}(\mathbf{W}))\|_\mathrm{F}^2 \\ & + \mu \|\mathbf{A}^{(k)} - \mathsf{E}_{\boldsymbol{\Theta}_\mathsf{E}}(\mathbf{W}) - \mathbf{G}^{(k)}\|_\mathrm{F}^2 \\ & \text{s.t. } \mathbf{1}_R^\top \mathsf{E}_{\boldsymbol{\Theta}_\mathsf{E}}(\mathbf{W}) = \mathbf{1}_N^\top,\ \mathsf{E}_{\boldsymbol{\Theta}_\mathsf{E}}(\mathbf{W}) \geq \mathbf{0} \text{ and } \mathbf{S} \geq \mathbf{0} \end{aligned} \tag{9}$$

$$\begin{aligned} \mathbf{A}^{(k+1)} = & \arg\min_{\mathbf{A}} \lambda \mathbf{A}^\top (\mathbf{A} - \mathsf{C}(\mathbf{A})) \\ & + \mu \|\mathbf{A} - \mathsf{E}^{(k+1)}_{\boldsymbol{\Theta}_\mathsf{E}}(\mathbf{W}) - \mathbf{G}^{(k)}\|_\mathrm{F}^2 \end{aligned} \tag{10}$$

$$\mathbf{G}^{(k+1)} = \mathbf{G} - \mathbf{A}^{(k+1)} + \mathsf{E}^{(k+1)}_{\boldsymbol{\Theta}_\mathsf{E}}(\mathbf{W}) \tag{11}$$

where μ is a penalty parameter and $\mathbf{G}$ is the dual variable. The framework of the proposed AE-RED is summarized in Fig. 1. It embeds a data-driven AE with a model-free RED. The algorithmic scheme is shown to be a convenient way to fuse the respective advantages of these two approaches. Note that, since the AE-based formulation is nonlinear, providing convergence guarantees about the resulting optimization scheme is not trivial. However, the experimental results reported in Section IV show that the proposed method is able to provide consistent performance. Finally, without loss of generality, detailed technical implementations of the first two steps (9) and (10) are discussed in the following paragraphs for specific architectures of the AE.

B. Updating $\boldsymbol{\Theta}$

At each iteration, the set of parameters $\boldsymbol{\Theta}$ of the AE is updated through the rule (9). This can be achieved by training the network with the criterion in (9) as the loss function. The first term measures the data fit while the second acts as a regularization to enforce the representation $\mathsf{E}_{\boldsymbol{\Theta}_\mathsf{E}}(\mathbf{W})$ in the latent space to be close to a corrected version $\mathbf{A} - \mathbf{G}$ of the abundance. Regarding the ASC, ANC and ENC constraints, they can be ensured by an appropriate design of the network. In practice, Adam is used to train the AE.

Various AE architectures can be envisioned and the encoder and the decoder can be chosen by the end-user with respect to the targeted applicative context. Some archetypal examples of possible elements composing these architectures (non-exhaustively) are listed in Fig. 1(c). The encoder $\mathsf{E}_{\boldsymbol{\Theta}_\mathsf{E}}(\cdot)$ aims at extracting relevant features to be incorporated into the estimated abundances. Training a network based on a single spectrum at a time ignores the spatial information. Therefore, patch-wise or cube-wise encoders are generally preferred to jointly capture the information across the image dimensions. A popular choice consists in adopting a CNN-based architecture where the input is the observed image. Another promising approach leverages on the more recent concept of deep image prior (DIP) with a random noise as input. These two particular choices will be discussed later in this section. Regarding the decoder $\mathsf{D}_{\boldsymbol{\Theta}_\mathsf{D}}(\cdot)$, it generally mimics the mixing process and the endmembers usually define the weights of one specially designed linear layer. Again, the proposed AE-RED framework is sufficiently flexible to host various architectures and to handle various spectral mixing models. A popular strategy is to design the decoder such that it combines physics-based and data-driven strategies to

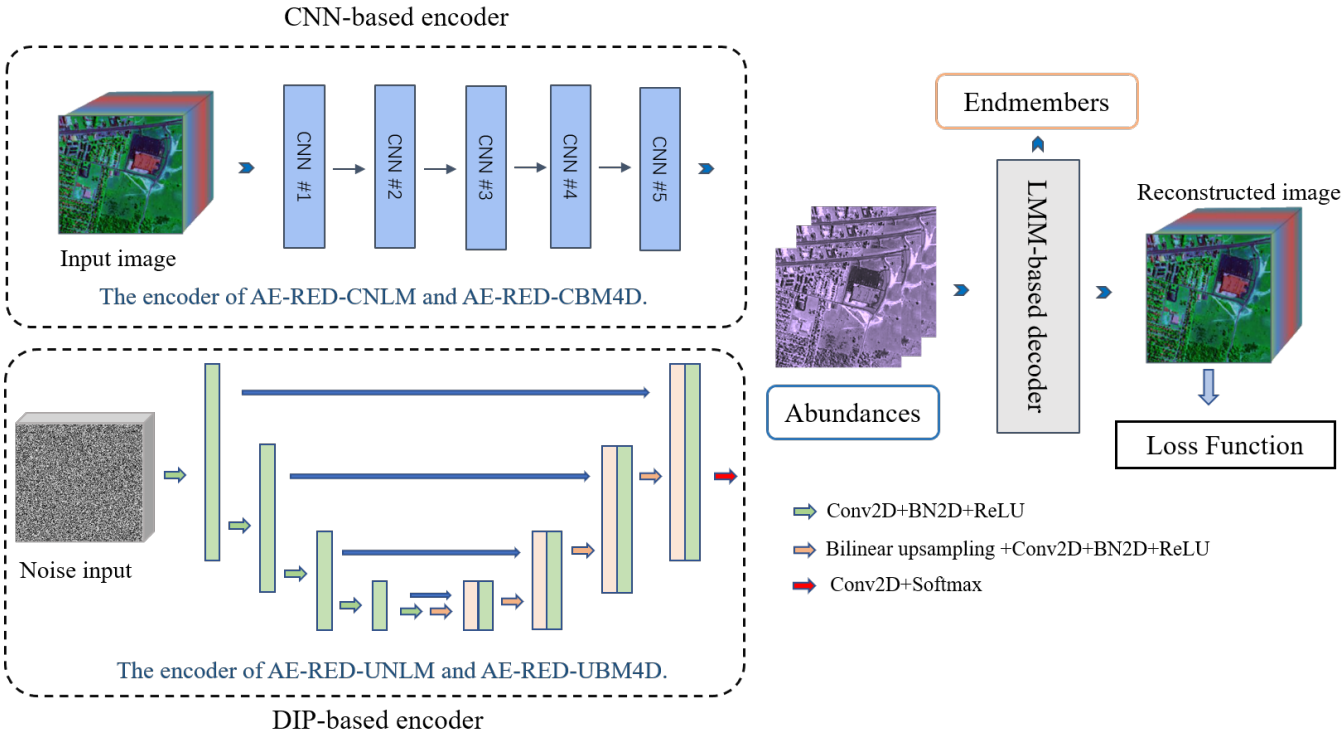

Fig. 2. Left: the architectures of CNN-based and DIP-based networks used as particular instances of the proposed method. Right: particular instance of the decoder to mimic the LMM.

account for complex nonlinearities or spectral variabilities. For instance, additive nonlinear and post-nonlinear models have been extensively investigated [33], [34], [37] as well as spectral variability-aware endmember generators [39], [40].

In the main body of this paper, for illustration purpose but without loss of generality, two particular architectures are discussed and then instantiated, as shown in Fig. 2. Both consider an LMM-based decoder composed of a convolutional layer with a filter size of $1 \times 1 \times B$. For this particular instance of LMM-based decoder, the optimization problem (9) can be rewritten as

$$\begin{aligned} \{\boldsymbol{\Theta}_{\mathsf{E}}, \mathbf{S}\} \in \arg\min_{\boldsymbol{\Theta}_{\mathsf{E}}, \mathbf{S}} & \|\mathbf{Y} - \mathbf{S}\mathsf{E}_{\boldsymbol{\Theta}_{\mathsf{E}}}(\mathbf{W})\|_{\mathrm{F}}^2 \\ & + \mu\|\mathbf{A} - \mathsf{E}_{\boldsymbol{\Theta}_{\mathsf{E}}}(\mathbf{W}) - \mathbf{G}\|_{\mathrm{F}}^2 \\ \text{s.t. } & \mathbf{1}_R^\top \mathsf{E}_{\boldsymbol{\Theta}_{\mathsf{E}}}(\mathbf{W}) = \mathbf{1}_N^\top,\ \mathsf{E}_{\boldsymbol{\Theta}_{\mathsf{E}}}(\mathbf{W}) \geq \mathbf{0} \text{ and } \mathbf{S} \geq \mathbf{0}. \end{aligned} \tag{12}$$

However, it is worth noting that Appendix A considers the case of a nonlinear model-based decoder to demonstrate the flexibility of the proposed. The two examples of AE considered in what follows differ by the architecture of the encoder. They have been chosen because they have been often used to perform unmixing. The first network is composed of a CNN-based encoder, which mainly consists of convolutional filters to extract and thus exploit the spatial features of the hyperspectral image. The second network is a DIP-based encoder. By generating output maps from an input noise, the image prior is implicitly encoded in the network parameters. More details about these two choices are given below.

1) CNN-based encoder: The architecture of the CNN-based encoder is shown in Fig. 2. The whole image $\mathbf{Y}$ is used here as the input to extract the structure information from the hyperspectral image. Another choice would consist in considering over-lapping patches as the input. The encoder is composed of 5 blocks. The first two blocks implement 3×3 convolution filters to learn the spatial consistency information. The next two blocks apply 1×1 convolution operators (i.e., fully connected layers) to model the spectral priors. Moreover, to satisfy the ANC and ASC, the conventional LeakyReLU activation function of the last block is replaced by a Softmax function. The output dimensions of each block are narrowly diminished to compress the input pixels into the abundance domain. Considering the optimization function defined in (12), the objective function to train this model is expressed as

$$\mathcal{L}_{\mathrm{AE}}(\boldsymbol{\Theta}) = \|\mathbf{Y} - \mathbf{S}\mathsf{E}_{\boldsymbol{\Theta}_{\mathsf{E}}}(\mathbf{Y})\|_{\mathrm{F}}^2 + \mu\|\mathbf{A} - \mathsf{E}_{\boldsymbol{\Theta}_{\mathsf{E}}}(\mathbf{Y}) - \mathbf{G}\|_{\mathrm{F}}^2 + \alpha\|\boldsymbol{\Theta}\|_{\mathbf{F}}^2. \tag{13}$$

An ℓ_2-norm is introduced in the loss function to penalize model weights and thereby reduce overfitting, and α is the penalty parameter. The resulting unmixing method will be denoted as AE-RED-C in the sequel.

2) Deep image prior-based encoder: Another architecture considered in this paper exploits the DIP strategy to implicitly learn the priors of hyperspectral image. Unlike conventional AE-based unmixing methods which use spectral signatures as input for training, this network applies a Gaussian noise image $\mathbf{Z}$ of size of the abundance matrix $\mathbf{A}$ as input to generate the hyperspectral image. The encoder can be a U-net like architecture to extract the features from different levels. In this work the encoder has been designed with an encoder-decoder structure for abundance estimation. The inner

encoder is composed of 4 down-sampling to compress the features. Each down-sampling block consists of three layers, namely a convolution layer with a filter of size 3×3, a batch normalization layer, and an ReLU nonlinear activation layer. The inner decoder is composed of 5 up-sampling blocks. Each of the first 4 blocks has 4 layers: a bilinear up-sampling layer, a convolution layer, a batch normalization layer and an ReLU nonlinear activation layer. The last block has two layers, namely a convolution layer and a Softmax nonlinear activation layer to generate the estimated abundances while satisfying the ANC and ASC. Skip connections relate the encoder and decoder which are used to fuse the low-level and high-level features and to obtain multiscale information. The objective function to train this deep model is also defined as (13) where $\mathsf{E}_{\boldsymbol{\Theta}_\mathsf{E}}(\mathbf{Y})$ is replaced by $\mathsf{E}_{\boldsymbol{\Theta}_\mathsf{E}}(\mathbf{Z})$. The proposed method with this architecture is denoted as AE-RED-U.

Remark (On the choice of the AE block number). The detailed design of any network architecture generally follows some empirical principles, subsequently validated by the reached performance face to a given task. Thus there is no universal rules to determine the number of blocks composing an encoder, as it depends on a variety of external factors, such as the size and the complexity of the input data. For the proposed architectures, adjusting the number of blocks can be guided by monitoring the unmixing performance as a function of the number of blocks.

Remark (On possible overfitting issues). During the numerical validation of the proposed approach, no overfitting issue has been experienced. This may be explained by the following four aspects which tend to prevent such shortcomings: *i)* during the training stage, the weights of the networks follow a weight decay strategy, i.e., they are granted with an ℓ_2-norm regularization, *ii)* the RED prior contributes to reducing the influence of the noise into the model, *iii)* by design, the proposed framework embeds various constraints imposed to the abundances (i.e., ASC and ANC) and the endmembers (ENC), which directly reduces the range of admissible solutions and *iv)* as in [48], training an AE-based unmixing network follows an iterative process which does not rely on a training set but rather considers a fixed input and iteratively adjusts the weights by assessing the quality of the network output.

C. Updating $\mathbf{A}$

The abundance matrix $\mathbf{A}$ is updated by solving (10). This problem is a standard RED objective function and can be interpreted as a denoising of $\mathsf{E}_{\boldsymbol{\Theta}_\mathsf{E}}(\mathbf{W}) + \mathbf{G}$. The seminal paper [46] discusses two algorithmic schemes to solve this problem, namely fixed-point and gradient-descent strategies. In this work we derive a fixed-point algorithm by setting the gradient of the objective function to $\mathbf{0}$,

$$\lambda \left(\mathbf{A} - \mathsf{C}\left(\mathbf{A}\right)\right) + \mu \left(\mathbf{A} - \mathsf{E}_{\boldsymbol{\Theta}_\mathsf{E}}(\mathbf{W}) - \mathbf{G}\right) = \mathbf{0}. \tag{14}$$

Algorithm 1 The proposed unmixing framework AE-RED

Input: Hyperspectal image $\mathbf{Y}$; Regularization parameter λ; ADMM coefficient μ; Denoiser $\mathsf{C}(\cdot)$; Outer and inner iteration numbers K and J; Training parameters (learning rate, epochs, batch size, α).

Initialization: $\boldsymbol{\Theta}$ randomly, $\mathbf{A}$ and $\mathbf{G}$ with $\mathbf{0}$, $\mathbf{S}$ with VCA.

% ADMM iterations

for $k = 1, \cdots, K$ **do**

% Updating $\boldsymbol{\Theta}$

  for $i = 1, \cdots,$ epochs **do**

    Update $\mathsf{E}_{\boldsymbol{\Theta}_\mathsf{E}}(\mathbf{W})$ via forward propagation,

    Compute the loss function by (13),

    Update $\boldsymbol{\Theta}^{(k)}$ via retropropagation,

  end for

% Updating $\mathbf{A}$

  Set $\mathbf{A}^{(k-1,0)} = \mathbf{A}^{(k-1)}$

  for $j = 1, \cdots, J$ **do**

    Update $\mathbf{A}^{(k-1,j)}$ with (15),

  end for

  Set $\mathbf{A}^{(k)} = \mathbf{A}^{(k-1,J)}$

% Updating $\mathbf{G}$

  Update $\mathbf{G}^{(k)}$ with (11);

end for

Output: Estimated abundances $\mathbf{A}$ and endmembers $\mathbf{S}$.

Then, at the $(k+1)$th iteration of the ADMM, the jth inner iteration of the fixed-point algorithm can be summarized as

$$\mathbf{A}^{(k+1,j+1)} = \frac{1}{\lambda + \mu}\left[\lambda \mathsf{C}\left(\mathbf{A}^{(k+1,j)}\right) + \mu\left(\mathsf{E}^{(k+1)}_{\boldsymbol{\Theta}_\mathsf{E}}(\mathbf{W}) + \mathbf{G}^{(k)}\right)\right]. \tag{15}$$

For illustration, we consider two particular denoisers $\mathsf{C}(\cdot)$, namely nonlocal means (NLM) [49] and block-matching and 4-D filtering (BM4D) [50]. NLM is a 2D denoiser and should be applied on each spectral bands independently, while BM4D is a 3D-cube based denoiser. Depending on the architecture chosen for the encoder (see Section III-B), the corresponding instances of the proposed framework are named as AE-RED-CNLM, AE-RED-CBM4D, AE-RED-UNLM and AE-RED-UBM4D, respectively. It is worth noting that the two denoisers considered in this work do not require any training procedure. Conversely, they are described by explicit parametric models to leverage the universal property of image self-similarity. However, the proposed framework is sufficiently flexible to embed other denoisers, in particular pretrained models described by deep neural networks [30], [31]. Simultaneously training a deep denoiser alongside the unmixing process would significantly increase the computational complexity of the proposed algorithm, without bringing noticeable performance improvements. In practical scenarios, various factors such as the available computing time and resources can heavily guide the choice of a denoiser. The end-user should take all the factors into consideration to make informed decisions.

TABLE II
SYNTHETIC DATA: PERFORMANCE OF ABUNDANCE AND ENDMEMBER ESTIMATIONS IN TERMS OF RMSES ($\times 10^{-2}$) AND MSADS ($\times 10^{-2}$), RESPECTIVELY. BEST RESULTS ARE REPORTED IN BOLD AND UNDERLINED NUMBERS DENOTE THE SECOND BEST RESULTS.

	5dB		10dB		20dB		30dB	
	RMSE	mSAD	RMSE	mSAD	RMSE	mSAD	RMSE	mSAD
SUnSAL-TV	11.12 ± 0.45	10.13 ± 0.31	8.04 ± 0.38	6.23 ± 0.21	2.84 ± 0.17	1.73 ± 0.08	1.04 ± 0.06	0.52 ± 0.03
PnP-NMF	10.29 ± 0.81	8.55 ± 0.62	7.11 ± 0.36	5.33 ± 0.19	3.11 ± 0.16	1.81 ± 0.09	1.17 ± 0.05	0.83 ± 0.03
CNNAE	10.78 ± 0.63	8.11 ± 0.51	6.82 ± 0.42	4.81 ± 0.28	2.92 ± 0.28	1.62 ± 0.11	1.27 ± 0.07	0.45 ± 0.03
UnDIP	14.69 ± 1.45	9.77 ± 0.20	8.54 ± 0.73	6.85 ± 0.16	2.80 ± 0.33	1.93 ± 0.13	1.00 ± 0.04	0.57 ± 0.03
SNMF	12.07 ± 1.32	8.52 ± 0.67	9.06 ± 0.96	5.95 ± 0.39	3.13 ± 0.42	1.13 ± 0.11	1.12 ± 0.11	0.43 ± 0.06
CyCU-Net	11.50 ± 0.83	8.26 ± 0.51	7.08 ± 0.74	5.69 ± 0.43	2.96 ± 0.69	1.46 ± 0.13	1.39 ± 0.58	0.69 ± 0.05
AE-RED-CNLM	9.43 ± 0.49	7.69 ± 0.23	6.40 ± 0.24	4.37 ± 0.19	2.61 ± 0.13	$\mathbf{1.03 \pm 0.09}$	0.97 ± 0.05	0.41 ± 0.01
AE-RED-CBM4D	10.09 ± 0.60	7.70 ± 0.26	6.65 ± 0.29	$\mathbf{4.30 \pm 0.16}$	$\mathbf{2.35 \pm 0.14}$	1.05 ± 0.08	$\mathbf{0.93 \pm 0.04}$	0.42 ± 0.03
AE-RED-UNLM	$\mathbf{9.19 \pm 0.40}$	$\mathbf{7.67 \pm 0.20}$	6.02 ± 0.31	4.34 ± 0.17	2.41 ± 0.19	1.08 ± 0.08	0.95 ± 0.06	0.40 ± 0.02
AE-RED-UBM4D	9.72 ± 0.35	7.68 ± 0.27	$\mathbf{5.85 \pm 0.32}$	4.33 ± 0.18	2.51 ± 0.14	1.07 ± 0.09	0.94 ± 0.05	$\mathbf{0.39 \pm 0.01}$

TABLE III
SYNTHETIC DATA: PERFORMANCE OF ENDMEMBER ESTIMATION AND IMAGE RECONSTRUCTION IN TERMS OF MSIDS ($\times 10^{-2}$) AND PSNR, RESPECTIVELY. BEST RESULTS ARE REPORTED IN BOLD AND UNDERLINED NUMBERS DENOTE THE SECOND BEST RESULTS.

	5dB		10dB		20dB		30dB	
	mSID	PSNR	mSID	PSNR	mSID	PSNR	mSID	PSNR
SUnSAL-TV	3.91 ± 0.14	30.83 ± 2.17	1.20 ± 0.07	35.35 ± 0.94	0.13 ± 0.02	43.94 ± 0.63	0.02 ± 0.00	54.43 ± 0.34
PnP-NMF	1.95 ± 0.13	31.68 ± 1.60	1.00 ± 0.07	36.29 ± 0.95	0.11 ± 0.02	44.35 ± 0.57	0.02 ± 0.00	54.65 ± 0.61
CNNAE	4.32 ± 0.16	31.45 ± 1.62	0.69 ± 0.08	35.25 ± 1.22	0.13 ± 0.02	43.35 ± 0.60	0.03 ± 0.00	50.94 ± 0.47
UnDIP	6.50 ± 0.32	30.30 ± 1.63	1.30 ± 0.14	34.82 ± 1.27	0.23 ± 0.04	44.31 ± 0.92	$\mathbf{0.01 \pm 0.00}$	54.70 ± 0.25
SNMF	13.69 ± 0.66	28.14 ± 2.67	1.12 ± 0.29	32.22 ± 1.89	0.07 ± 0.02	41.25 ± 0.81	$\mathbf{0.01 \pm 0.00}$	51.40 ± 0.41
CyCU-Net	4.47 ± 0.21	30.82 ± 2.50	0.52 ± 0.16	35.48 ± 1.97	0.14 ± 0.05	42.69 ± 0.93	0.03 ± 0.00	50.15 ± 0.53
AE-RED-CNLM	$\mathbf{1.84 \pm 0.15}$	$\mathbf{32.49 \pm 1.69}$	0.38 ± 0.04	36.89 ± 0.92	$\mathbf{0.05 \pm 0.01}$	44.41 ± 0.37	$\mathbf{0.01 \pm 0.00}$	54.70 ± 0.15
AE-RED-CBM4D	1.89 ± 0.10	31.71 ± 1.21	$\mathbf{0.36 \pm 0.03}$	36.13 ± 0.73	0.06 ± 0.02	$\mathbf{45.32 \pm 0.34}$	$\mathbf{0.01 \pm 0.00}$	54.82 ± 0.16
AE-RED-UNLM	1.95 ± 0.13	32.02 ± 1.48	0.37 ± 0.04	36.68 ± 0.87	0.07 ± 0.01	44.49 ± 0.40	$\mathbf{0.01 \pm 0.00}$	55.03 ± 0.21
AE-RED-UBM4D	1.87 ± 0.14	32.28 ± 1.98	0.37 ± 0.05	$\mathbf{36.90 \pm 0.93}$	0.06 ± 0.02	44.53 ± 0.37	$\mathbf{0.01 \pm 0.00}$	$\mathbf{55.24 \pm 0.17}$

IV. EXPERIMENTAL RESULTS

This section presents experiments conducted to evaluate the effectiveness of the proposed unmixing framework. These experiments have been conducted on synthetic and real data sets to quantitatively assess the unmixing results and to demonstrate the effectiveness of our proposed method in real applications, respectively (see Sections IV-A and IV-B).

Compared methods – Several state-of-the-art methods have been compared. A first family of unmixing algorithms are conventional methods. SUnSAL-TV [16] leverages on a handcrafted TV-term to regularize the optimization function. PnP-NMF [9] is an NMF-based unmixing method, and denoisers are embedded as PnP to introduce prior information. A second family of compared methods is based on deep learning. CNNAE [10] is a deep AE-based unmixing method where convolutional filters capture spatial information. UnDIP [48] is a DIP-based unmixing method which uses a convolutional network. A geometric endmember extraction method is applied to estimate endmembers. SNMF [43] is a deep unrolling algorithm, which unfolds the ℓ_p-sparsity constrained NMF model into trainable deep architectures. CyCU-Net [11] proposes a cascaded AEs for unmixing with a cycle-consistency loss to enhance the unmixing performance. For all methods, the endmembers have been initialized with the signatures extracted by a popular dedicated algorithm, namely vertex component analysis (VCA) [51], since it has empirically shown to provide the most consistent results. The other network parameters have been initialized randomly, while the abundance matrix $\mathbf{A}$ and the dual variables $\mathbf{G}$ have been initialized with zeros.

Hyperparameter settings – Regarding the experiments conducted on synthetic data, the hyperparameters have been adjusted following a grid search strategy to obtain the best unmixing results and to conduct fair comparisons. For example, the number of blocks of the encoder has been progressively increased to reach the best unmixing performance. Regarding the experiments on real data sets, due to the absence of available ground truth, these hyperparameters have been adjusted in the same ranges of values obtained on the synthetic data by empirically inspecting the unmixing results. The values are reported in Appendix B.

The learning rate to train the deep networks is set to 1×10^{-3}, and set to 1×10^{-4} when fine-tuning the decoder

weights. For the proposed CNN-based encoder, the number K of ADMM iterations is set to 15, the number of epochs is set to 250 and the number of inner iterations when updating the abundances is set to $J = 1$. As for the proposed DIP-based encoder, K, the number of epochs and J are respectively set to 10, 2300 and 1.

Performance metrics – The root mean square error (RMSE) is used to evaluate the abundance estimation performance, which can be expressed by

$$\text{RMSE} = \sqrt{\frac{1}{NR}\sum_{i=1}^{N}\|\mathbf{a}_i - \hat{\mathbf{a}}_i\|^2}, \tag{16}$$

where $\mathbf{a}_i$ is the actual abundance of the ith pixel, and $\hat{\mathbf{a}}_i$ is the corresponding estimate. The lower the RMSE, the better the abundance estimates. The endmember estimation is assessed by computing the mean spectral angle distance (mSAD) and the mean spectral information divergence (mSID) given by

$$\text{mSAD} = \frac{1}{R}\sum_{r=1}^{R}\arccos\left(\frac{\mathbf{s}_r^\top\hat{\mathbf{s}}_r}{\|\mathbf{s}_r\|\|\hat{\mathbf{s}}_r\|}\right) \tag{17}$$

and

$$\text{mSID} = \frac{1}{R}\sum_{r=1}^{R}\mathbf{p}_r\log\left(\frac{\mathbf{p}_r}{\hat{\mathbf{p}}_r}\right), \tag{18}$$

where $\mathbf{s}_r$ and $\hat{\mathbf{s}}_r$ are the actual and estimate of the rth endmember, respectively, $\mathbf{p}_r = \mathbf{s}_r/\mathbf{1}^\top\mathbf{s}_r$ and $\hat{\mathbf{p}}_r = \hat{\mathbf{s}}_r/\mathbf{1}^\top\hat{\mathbf{s}}_r$. The smaller the mSAD and mSID, the better the endmember estimates. Finally, the peak signal-to-noise ratio (PSNR) is used to evaluate the image denoising and reconstruction, which is defined by

$$\text{PSNR} = 10\times\log_{10}\left(\frac{\text{MAX}^2}{\text{MSE}}\right) \tag{19}$$

where MAX is the maximum pixel value of the reconstructed image $\hat{\mathbf{Y}}$ and MSE is the mean square error between the reconstructed image and the noise-free image. The higher the PSNR, the better the reconstruction.

To assess the statistical significance of the reported experimental results, these metrics have been averaged over 10 Monte Carlo runs.

A. Experiments on synthetic data sets

Data description – The synthetic images are composed of 100×100 pixels. Abundance maps are generated using the method of the Hyperspectral Imagery Synthesis tools[1] to mimic the spatial content exhibited by remote sensing hyperspectral images. The ground-truth abundance maps are shown in Fig. 3 (1st column). Sets of $R = 5$ endmembers are randomly selected from the U.S. Geological Survey (USGS) spectral library with a number of spectral bands of $B = 224$. These endmembers are mixed according to the LMM and an additive zero-mean Gaussian noise is considered with variances adjusted according to 4 signal-to-noise ratios (SNRs), i.e., $\text{SNR} \in \{5\text{dB}, 10\text{dB}, 20\text{dB}, 30\text{dB}\}$.

[1]http://www.ehu.es/ccwintco/index.php/Hyperspectral Imagery Synthesis tools for MATLAB

Results – Tables II-III report the estimation results obtained by the compared algorithms in terms of RMSE for the abundance estimation, mSAD and mSID for the endmember estimation and PSNR for the reconstruction. Conventional unmixing methods, such as SUnSAL-TV and PnP-NMF, achieve good unmixing results, demonstrating the usefulness of the explicit prior provided by manually designed regularization. Deep learning-based methods, such as CNNAE, SNMF and CyCU-Net, they can obtain suitable unmixing results and better endmember estimation results compared with the conventional methods, illustrating the ability of deep networks to embed prior information. These results also show that the proposed AE-RED framework outperforms the compared state-of-the-art methods, across all performance metrics and the noise levels. Fig. 3 depicts the estimated abundance maps associated with the synthetic data set with SNR= 10dB. It can be observed that the abundance maps estimated by the AE-RED framework exhibit better agreement with the ground-truth, whatever the implementations (architectures and denoisers). Fig. 4 shows the endmembers estimated by the proposed framework on the synthetic data set with SNR = 10 dB, which are close to the ground-truth.

Sensitivity analysis – Fig. 5 shows how the parameters λ, μ, learning rate and epoch impact the performance of AE-RED-CNLM with synthetic data (SNR = 10dB).

Ablation study – An ablation study has been conducted to evaluate the effectiveness of each component of the proposed framework. First, counterparts of AE-RED-CNLM and AE-RED-UNLM, referred to as AE-C and AE-U, respectively, do not include RED as a regularization. Second, the two proposed methods are instantiated without deep AEs but only RED as regularization, and directly optimize abundances and endmembers with gradient descent algorithm, with the denoiser chosen as NLM (method referred to as RED-NLM) or BM4D (method referred to as RED-BM4D). Table IV provides the performance of abundance and endmember estimations in terms of RMSE and mSAD provided by these four depreciated methods. These results are significantly worse than those initially reported in Table II corresponding to the proposed methods combining both RED and deep AEs. This demonstrates the relevance of the adopted strategy.

Convergence analysis – Because of the use of highly nonlinear operators (i.e., deep AEs), the convergence of the proposed method can be hardly assessed theoretically. Instead, this convergence has been empirically monitored by evaluating RMSEs as functions of the algorithm iterations. Fig. 6 depicts the curves obtained by the four proposed methods when analysis the synthetic data with SNR = 10dB. These curves confirm appropriate behaviors of the algorithms.

Complexity analysis – The overall computational complexity of the proposed method can be analyzed with respect to

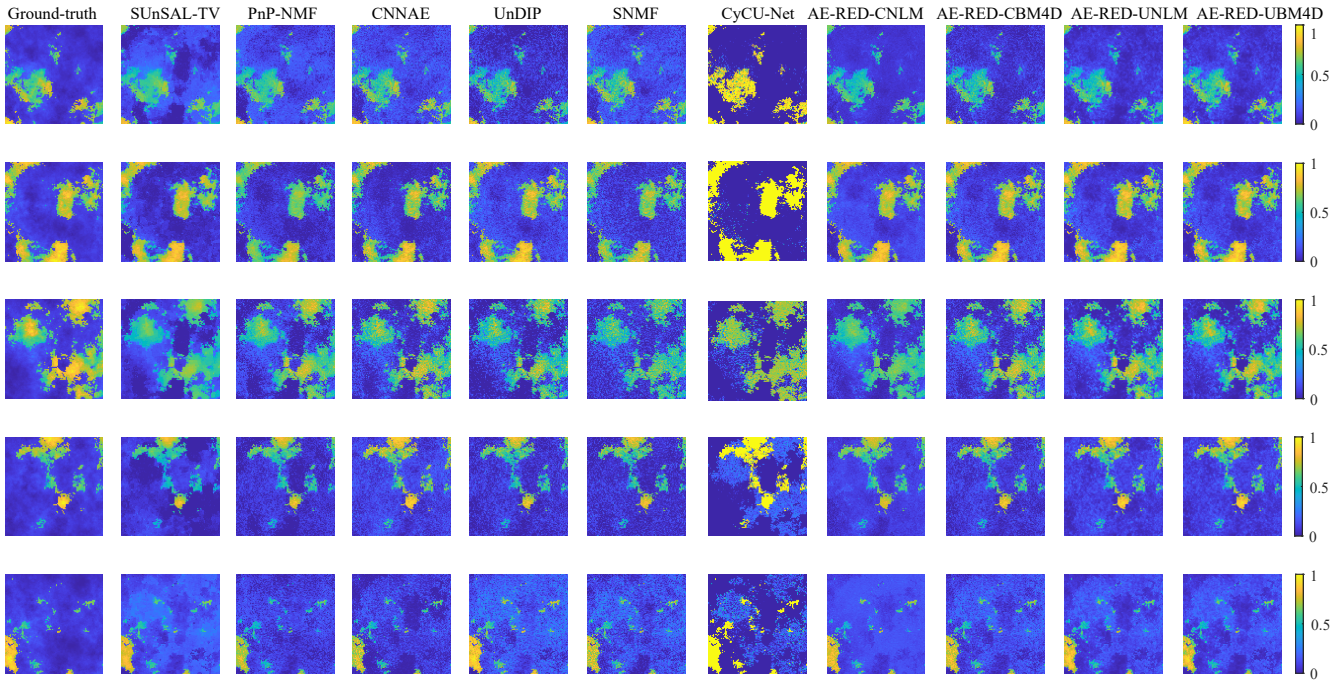

Fig. 3. Synthetic data, SNR $= 10$dB: estimated abundance maps.

TABLE IV
SYNTHETIC DATA, ABLATION STUDY: PERFORMANCE OF ABUNDANCE AND ENDMEMBER ESTIMATIONS IN TERMS OF RMSES ($\times 10^{-2}$) AND MSADS ($\times 10^{-2}$), RESPECTIVELY.

	5dB		10dB		20dB		30dB	
	RMSE	mSAD	RMSE	mSAD	RMSE	mSAD	RMSE	mSAD
AE-C	11.12 ± 0.62	8.86 ± 0.27	6.86 ± 0.34	5.66 ± 0.26	3.00 ± 0.21	1.32 ± 0.08	1.05 ± 0.05	0.43 ± 0.03
AE-U	11.67 ± 0.60	9.22 ± 0.26	6.94 ± 0.35	5.83 ± 0.21	2.82 ± 0.15	1.95 ± 0.08	1.01 ± 0.05	0.44 ± 0.03
RED-NLM	10.24 ± 0.71	9.18 ± 0.32	6.84 ± 0.36	5.50 ± 0.20	3.10 ± 0.24	1.34 ± 0.09	1.11 ± 0.05	0.48 ± 0.02
RED-BM4D	10.15 ± 0.64	8.93 ± 0.21	6.81 ± 0.29	5.36 ± 0.20	3.03 ± 0.23	1.65 ± 0.08	1.09 ± 0.05	0.42 ± 0.02

two essential building blocks, namely the AE and the RED components. Regarding the complexity of the AEs, because of the independence between the samples, their training can be globally evaluated with respect to the number of samples. During the inference step (i.e., once trained), their computational burden depends on the architecture and can be evaluated by forward inference floating point operations (FLOPs), 9.62G for CNN-based network and 8.22G for DIP-based network. Regarding the use of RED, when the number of inner loop is fixed to $J = 1$, its complexity is the same as the one imposed by conventional PnP frameworks [46].

The computational burdens of the compared method have been also evaluated in terms of execution times, reported in Table V. For all deep learning-based methods, they correspond to both training and test stages. The execution time required by the proposed framework is shown to depend on the chosen deep architecture. More precisely, when using CNN-based encoders, the execution times of the proposed framework are of the same order as those of UnDIP or SNMF. These times are significantly longer when using DIP as encoders.

To conclude, it is fair noting that the versatility and the accuracy of the proposed framework come at the price of a heavier computational burden. However some strategies have been deployed to make the proposed methods scalable. First, the optimization strategy detailed in Section III follows a variable splitting scheme (i.e., ADMM), which is known to converge significantly faster than first-order methods. Second, thanks to this splitting scheme, updating the encoder parameters $\boldsymbol{\Theta}_{\mathsf{E}}$ and applying the denoiser $\mathsf{C}(\cdot)$ have been achieved in parallel by exploiting a multi-core processing strategy, as already suggested in Fig. 1(b). Finally, as for most deep learning-based numerical solutions, the use of GPUs to train the AEs is a true asset.

B. Experiments on real data sets

Data description – Finally, experiments conducted on two real data sets are discussed. Firstly, one considers the Samson data set, which was acquired by the SAMSON observer and contains $B = 156$ spectral channels ranging from 400nm to

TABLE V
SYNTHETIC DATA, SNR = 10DB: COMPUTATIONAL TIMES (S).

SUnSAL-TV	PnP-NMF	CNNAE	UnDIP	SNMF	CyCU-Net	AE-RED-CNLM	AE-RED-CBM4D	AE-RED-UNLM	AE-RED-UBM4D
10.7	7.33	33.78	218.53	61.57	29.56	117.53	113.24	553.11	587.51

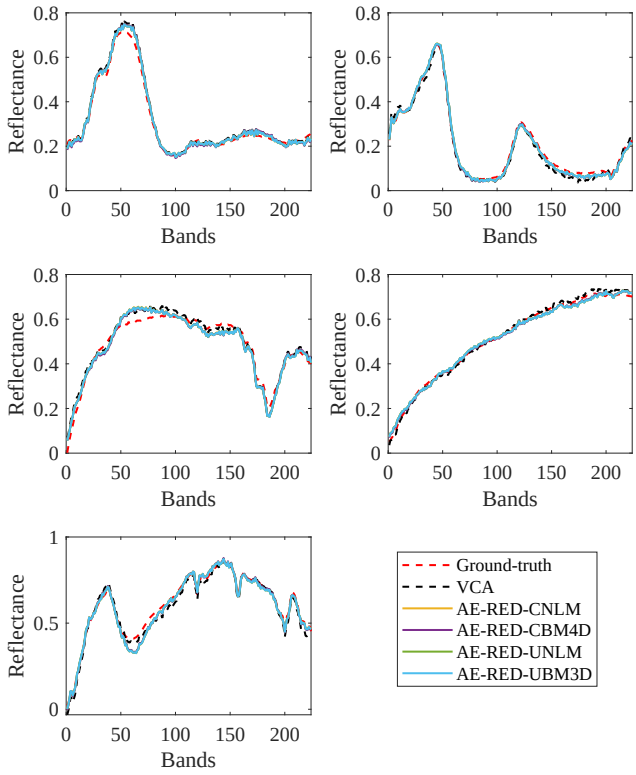

Fig. 4. Synthetic data, SNR = 10dB: estimated endmembers.

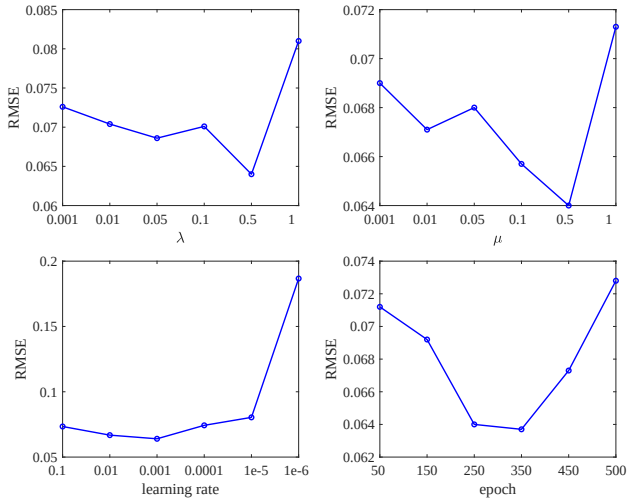

Fig. 5. Synthetic data, SNR = 10dB: RMSE as functions of the regularization parameters λ, μ, learning rate and epoch for AE-RED-CNLM.

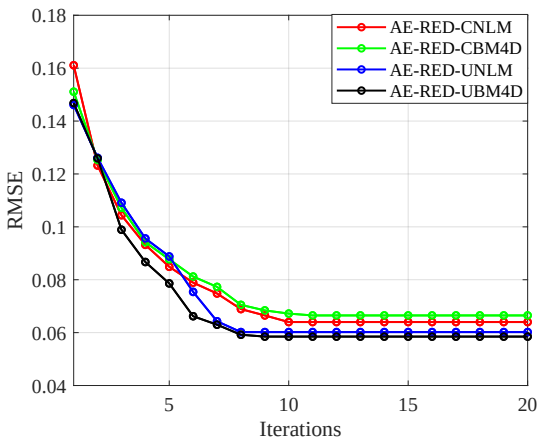

Fig. 6. Synthetic data, SNR = 10dB: RMSEs as functions of iterations.

889nm. The original image is of size of 952×952 pixels, and a subimage of 95×95 pixels is cropped for the experiments. This image contains three endmembers, namely "water", "tree" and "soil". The second real data set used in these experiments is known as the Jasper Ridge image. It was acquired by Analytical Imaging and Geophysics (AIG) in 1999 with $B = 224$ spectral bands covering a spectral range from 380nm to 2500nm. One considers a subimage of size of 100×100 pixels and $B = 198$ channels after removing the bands affected by water vapor and atmospheric effects. It contains $R = 4$ endmembers, namely "water", "soil", "tree" and "road".

Results – As there is no available ground-truth for these real data sets, a quantitative evaluation of abundance and endmember estimations cannot be provided. One alternative consists in conducting qualitative evaluation by visual inspection. Fig. 7 shows the abundance maps estimated by the compared methods for the Samson data set. The proposed AE-RED framework can successfully separate the materials and provide sharp abundance estimates. Fig. 8 depicts the abundance maps estimated by all compared methods for the Jasper Ridge data set. Some of them, such as UnDIP, fail to recover the road. Due to the learning ability of deep networks, most deep learning based methods are able to distinguish the individual materials. Finally the proposed AE-RED framework provides abundance maps with more detailed information and sharper boundaries.

V. CONCLUSION

This paper proposed a generic unmixing framework to embed RED within an AE. By carefully designing the encoder and the decoder, the AE was able to provide estimated abundance maps and endmember spectra. In particular, for

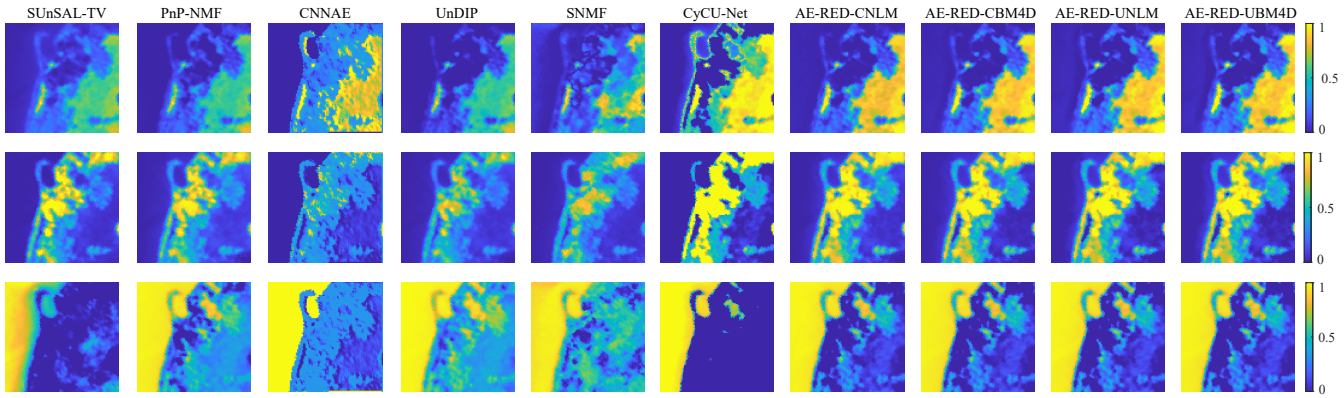

Fig. 7. Samson data set: estimated abundance maps.

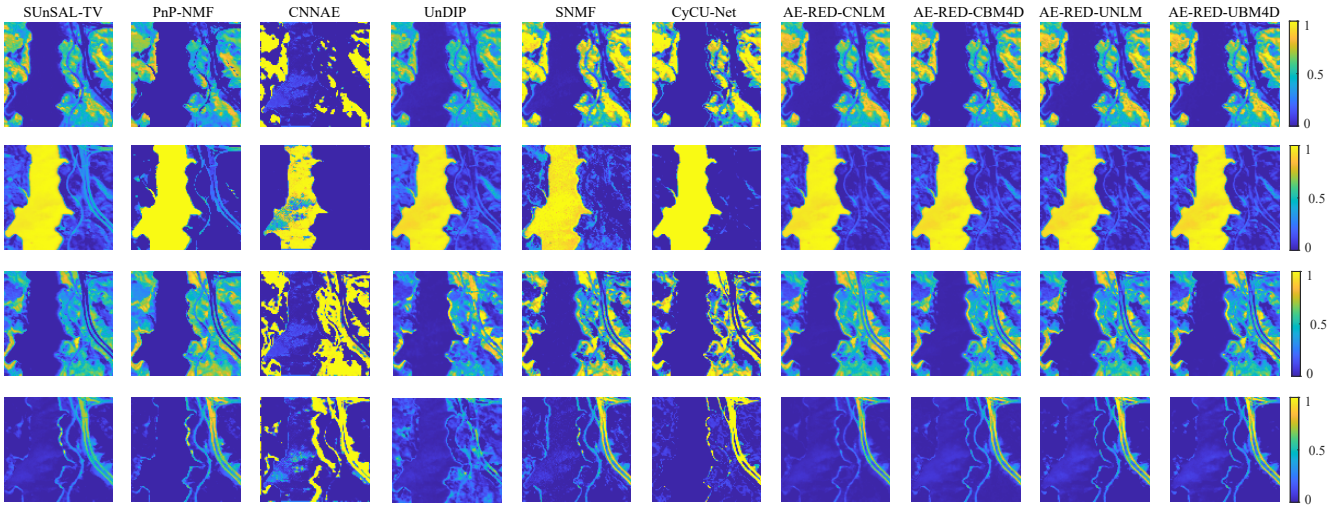

Fig. 8. Jasper Ridge data set: estimated abundance maps.

illustration purpose, two different encoder architectures are considered, namely a CNN and a DIP. Moreover the decoder could be chosen according to a particular mixture model. Leveraging ADMM scheme, the resulting optimization problem was split into simpler subproblems. The first one was described by an objective function composed of a data-fitting term and a quadratic regularization. It was solved through the training of an AE. The second subproblem was a standard RED objective function and solved by a fixed-point strategy. Two denoisers were considered, namely NLM and BM4D. The effectiveness of the proposed framework was evaluated through experiments conducted on synthetic and real data sets. The results showed that the proposed framework outperformed state-of-the-art methods. Future works include considering explicit endmember priors within the proposed framework, automatically selecting mixing model, and deriving some online learning strategies to extend the proposed framework for real-time processing.

APPENDIX A
HANDLING NONLINEAR MIXING MODELS

To illustrate the versatility of the proposed framework, it has been instantiated to handle nonlinear mixtures. More precisely, the LMM-based decoder initially considered in Section III-B has been replaced by the additive post-nonlinear decoder proposed in [33]. A synthetic data set has been generated using the bilinear model and post-nonlinear mixture model (PPNM) defined as

$$\mathbf{y} = \mathbf{S}\mathbf{a} + \sum_{i=1}^{R-1} \sum_{j=i+1}^{R} a_i a_j \left(\mathbf{s}_i \odot \mathbf{s}_j\right) + \mathbf{n} \tag{20}$$

and

$$\mathbf{y} = \mathbf{S}\mathbf{a} + \mathbf{S}\mathbf{a} \odot \mathbf{S}\mathbf{a} + \mathbf{n}. \tag{21}$$

The endmembers and abundances are set as the same as those used in the experiments described in Section IV-A, with $\mathrm{SNR} = 10\mathrm{dB}$. The proposed PPNM-based instance of the AE-RED framework has been compared to two state-of-the-art methods. The first method is the robust NMF (rNMF) proposed in [52]. It is a standard matrix factorization model complemented with an additional spatially sparse term to fit

any nonlinearities, here considered as outliers. The second compared method, referred to as LSTM-DNN, is an AE-based nonlinear unmixing method with recurrent neural network layers as encoder and a PPNM-based decoder [33].

The unmixing results obtained by the compared methods are reported in Table VI. They show that the proposed method is able to handle nonlinearly mixed pixels successfully. Again, compared to rNMF, deep learning-based unmixing methods achieve better results, which shows the ability of deep networks to learn image features. It is worth noting that the proposed and LSTM-DNN share the same decoder structure. However, the former gets better results, which indicates the interest of combining AE and RED priors.

APPENDIX B
HYPERPARAMETER SETTING

The values of the hyperparameters adjusted for the experiments conducted on the synthetic and real data sets are reported in Tables VII and VIII, respectively.

REFERENCES

[1] N. Dobigeon, L. Tits, B. Somers, Y. Altmann, and P. Coppin, "A comparison of nonlinear mixing models for vegetated areas using simulated and real hyperspectral data," *IEEE J. Sel. Topics Appl. Earth Observations Remote Sensing*, vol. 7, no. 6, pp. 1869–1878, June 2014.

[2] P. W. Yuen and M. Richardson, "An introduction to hyperspectral imaging and its application for security, surveillance and target acquisition," *The Imaging Science Journal*, vol. 58, no. 5, pp. 241–253, 2010.

[3] Y. Duan, H. Huang, and T. Wang, "Semisupervised feature extraction of hyperspectral image using nonlinear geodesic sparse hypergraphs," *IEEE Trans. Geosci. Remote Sens.*, 2021.

[4] F. Luo, L. Zhang, B. Du, and L. Zhang, "Dimensionality reduction with enhanced hybrid-graph discriminant learning for hyperspectral image classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 58, no. 8, pp. 5336–5353, 2020.

[5] 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.

[6] 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 Proc. Mag.*, vol. 31, no. 1, pp. 82–94, 2013.

[7] R. A. Borsoi, T. Imbiriba, J. C. M. Bermudez, C. Richard, J. Chanussot, L. Drumetz, J.-Y. Tourneret, A. Zare, and C. Jutten, "Spectral variability in hyperspectral data unmixing: A comprehensive review," *IEEE Geosci. Remote Sens. Mag.*, vol. 9, no. 4, pp. 223–270, 2021.

[8] L. Dong, Y. Yuan, and X. Luxs, "Spectral–spatial joint sparse NMF for hyperspectral unmixing," *IEEE Trans. Geosci. Remote Sens.*, vol. 59, no. 3, pp. 2391–2402, 2020.

[9] M. Zhao, T. Gao, J. Chen, and W. Chen, "Hyperspectral unmixing via nonnegative matrix factorization with handcrafted and learned priors," *IEEE Geosci. Remote Sens. Lett.*, vol. 19, pp. 1–5, 2021.

[10] 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.

[11] L. Gao, Z. Han, D. Hong, B. Zhang, and J. Chanussot, "CyCU-Net: Cycle-consistency unmixing network by learning cascaded autoencoders," *IEEE Trans. Geosci. Remote Sens.*, 2021.

[12] 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, 2018.

[13] 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.

[14] M.-D. 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, 2013.

[15] P. V. Giampouras, K. E. Themelis, A. A. Rontogiannis, and K. D. Koutroumbas, "Simultaneously sparse and low-rank abundance matrix estimation for hyperspectral image unmixing," *IEEE Trans. Geosci. Remote Sens.*, vol. 54, no. 8, pp. 4775–4789, 2016.

[16] M.-D. Iordache, J. M. Bioucas-Dias, and A. Plaza, "Total variation spatial regularization for sparse hyperspectral unmixing," *IEEE Trans. Geosci. Remote Sens.*, vol. 50, no. 11, pp. 4484–4502, 2012.

[17] 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.

[18] J. Chen, M. Zhao, X. Wang, C. Richard, and S. Rahardja, "Integration of physics-based and data-driven models for hyperspectral image unmixing: A summary of current methods," *IEEE Signal Proc. Mag.*, vol. 40, no. 2, pp. 61–74, 2023.

[19] X. Zhang, Y. Yuan, and X. Li, "Sparse unmixing based on adaptive loss minimization," *IEEE Trans. Geosci. Remote Sens.*, vol. 60, pp. 1–14, 2022.

[20] J. Peng, W. Sun, H.-C. Li, W. Li, X. Meng, C. Ge, and Q. Du, "Low-rank and sparse representation for hyperspectral image processing: A review," *IEEE Geosci. Remote Sens. Mag.*, 2021.

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

[22] I. Sutskever, J. Martens, G. Dahl, and G. Hinton, "On the importance of initialization and momentum in deep learning," in *International conference on machine learning*. PMLR, 2013, pp. 1139–1147.

[23] J. Sigurdsson, M. O. Ulfarsson, and J. R. Sveinsson, "Hyperspectral unmixing with ℓ_p regularization," *IEEE Trans. Geosci. Remote Sens.*, vol. 52, no. 11, pp. 6793–6806, 2014.

[24] T. Ince and N. Dobigeon, "Weighted residual NMF with spatial regularization for hyperspectral unmixing," *IEEE Geosci. Remote Sens. Lett.*, vol. 19, no. 6010705, June 2022.

[25] ——, "A fast spatial-spectral NMF for hyperspectral unmixing," *IEEE Geoscience and Remote Sensing Letters*, vol. 20, no. 5505305, June 2023.

[26] X. Wang, Y. Zhong, L. Zhang, and Y. Xu, "Spatial group sparsity regularized nonnegative matrix factorization for hyperspectral unmixing," *IEEE Trans. Geosci. Remote Sens.*, vol. 55, no. 11, pp. 6287–6304, 2017.

[27] T. Ince and N. Dobigeon, "Fast hyperspectral unmixing using a multiscale sparse regularization," *IEEE Geoscience and Remote Sensing Letters*, vol. 19, no. 6015305, Oct. 2022.

[28] A. Lagrange, M. Fauvel, S. May, and N. Dobigeon, "Matrix cofactorization for joint spatial–spectral unmixing of hyperspectral images," *IEEE Trans. Geosci. Remote Sens.*, vol. 58, no. 7, pp. 4915–4927, 2020.

[29] L. Tong, J. Zhou, B. Qian, J. Yu, and C. Xiao, "Adaptive graph regularized multilayer nonnegative matrix factorization for hyperspectral unmixing," *IEEE J. Sel. Top. Appl. Earth Observat. Remote Sens.*, vol. 13, pp. 434–447, 2020.

[30] M. Zhao, X. Wang, J. Chen, and W. Chen, "A plug-and-play priors framework for hyperspectral unmixing," *IEEE Trans. Geosci. Remote Sens.*, vol. 60, pp. 1–13, 2021.

[31] Z. Wang, L. Zhuang, L. Gao, A. Marinoni, B. Zhang, and M. K. Ng, "Hyperspectral nonlinear unmixing by using plug-and-play prior for abundance maps," *Remote Sensing*, vol. 12, no. 24, p. 4117, 2020.

[32] 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, 2018.

[33] M. Zhao, L. Yan, and J. Chen, "LSTM-DNN based autoencoder network for nonlinear hyperspectral image unmixing," *IEEE J. Sel. Top. Sig. Process.*, vol. 15, no. 2, pp. 295–309, 2021.

[34] 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.*, vol. 60, pp. 1–15, 2021.

[35] Z. Hua, X. Li, Y. Feng, and L. Zhao, "Dual branch autoencoder network for spectral-spatial hyperspectral unmixing," *IEEE Geosci. Remote Sens. Lett.*, vol. 19, pp. 1–5, 2021.

[36] M. Wang, M. Zhao, J. Chen, and R. Susanto, "Nonlinear unmixing of hyperspectral data via deep autoencoder network," *IEEE Geosci. Remote Sens. Lett.*, pp. 1–5, 2019.

[37] K. T. Shahid and I. D. Schizas, "Unsupervised hyperspectral unmixing via nonlinear autoencoders," *IEEE Trans. Geosci. Remote Sens.*, vol. 60, pp. 1–13, 2021.

[38] Y. Altmann, N. Dobigeon, and J.-Y. Tourneret, "Bilinear models for nonlinear unmixing of hyperspectral images," in *Proc. IEEE GRSS Workshop on Hyperspectral Image and SIgnal Processing: Evolution in Remote Sensing (WHISPERS)*, Lisbon, Portugal, June 2011, pp. 1–4.

TABLE VI
SYNTHETIC DATA, NONLINEAR MODEL: PERFORMANCE IN TERMS OF RMSE ($\times 10^{-2}$), MSAD ($\times 10^{-2}$), MSID ($\times 10^{-2}$), AND PSNR. BEST RESULTS ARE REPORTED IN BOLD AND UNDERLINED NUMBERS DENOTE THE SECOND BEST RESULTS.

	RMSE		mSAD		mSID		PSNR	
	Bilinear	PNMM	Bilinear	PNMM	Bilinear	PNMM	Bilinear	PNMM
rNMF	9.65 ± 0.52	6.43 ± 0.77	6.32 ± 0.58	12.31 ± 2.49	3.77 ± 0.48	3.73 ± 0.74	26.88 ± 0.81	27.58 ± 0.94
LSTM-DNN	8.70 ± 0.81	6.98 ± 0.61	5.97 ± 0.51	10.77 ± 1.04	1.33 ± 0.27	1.73 ± 0.64	27.56 ± 0.88	27.39 ± 0.86
AE-RED-CNLM	$\underline{7.17 \pm 0.83}$	$\mathbf{5.69 \pm 0.93}$	$\underline{5.05 \pm 0.49}$	10.37 ± 1.53	1.02 ± 0.39	2.23 ± 0.79	$\mathbf{31.38 \pm 0.83}$	28.22 ± 0.73
AE-RED-CBM4D	7.25 ± 0.53	$\underline{5.95 \pm 0.71}$	5.16 ± 0.50	9.43 ± 0.96	1.01 ± 0.29	1.52 ± 0.63	28.25 ± 0.71	27.88 ± 0.82
AE-RED-UNLM	8.06 ± 0.86	6.01 ± 0.58	5.25 ± 0.24	$\underline{9.39 \pm 0.88}$	$\underline{0.89 \pm 0.13}$	$\mathbf{1.41 \pm 0.71}$	28.67 ± 0.72	$\underline{28.87 \pm 0.78}$
AE-RED-UBM4D	$\mathbf{6.91 \pm 0.75}$	6.34 ± 0.45	$\mathbf{5.04 \pm 0.39}$	$\mathbf{9.34 \pm 0.97}$	$\mathbf{0.83 \pm 0.27}$	$\underline{1.47 \pm 0.69}$	$\underline{30.69 \pm 0.84}$	$\mathbf{29.19 \pm 0.83}$

TABLE VII
SYNTHETIC DATA SETS: HYPERPARAMETER SETTINGS.

	5dB	10dB	20dB	30dB
SUnSAL-TV	$\lambda = 4 \times 10^{-4}$, $\lambda_{\text{TV}} = 1 \times 10^{-3}$	$\lambda = 1 \times 10^{-3}$, $\lambda_{\text{TV}} = 5 \times 10^{-4}$	$\lambda = 1 \times 10^{-3}$, $\lambda_{\text{TV}} = 1 \times 10^{-4}$	$\lambda = 1 \times 10^{-3}$, $\lambda_{\text{TV}} = 1 \times 10^{-4}$
PnP-NMF	$\alpha = 1 \times 10^{-4}$, $\mu = 1 \times 10^{-2}$, $\lambda = 3 \times 10^{-2}$	$\alpha = 1 \times 10^{-4}$, $\mu = 1 \times 10^{-2}$, $\lambda = 5 \times 10^{-2}$	$\alpha = 1 \times 10^{-4}$, $\mu = 1 \times 10^{-3}$, $\lambda = 1 \times 10^{-2}$	$\alpha = 1 \times 10^{-4}$, $\mu = 1 \times 10^{-2}$, $\lambda = 5 \times 10^{-2}$
CNNAE	$\alpha = 3.5$	$\alpha = 3.5$	$\alpha = 3.5$	$\alpha = 3.5$
UnDIP	–	–	–	–
SNMF	$K = 9$	$K = 9$	$K = 9$	$K = 9$
CyCU-Net	$\beta = 0.5$, $\sigma = 1 \times 10^{-2}$, $\gamma = 1 \times 10^{-6}$	$\beta = 0.5$, $\sigma = 1 \times 10^{-2}$, $\gamma = 1 \times 10^{-6}$	$\beta = 0.6$, $\sigma = 1 \times 10^{-2}$, $\gamma = 1 \times 10^{-7}$	$\beta = 0.7$, $\sigma = 1 \times 10^{-2}$, $\gamma = 1 \times 10^{-7}$
AE-RED-CNLM	$\lambda = 0.5$, $\mu = 0.5$	$\lambda = 0.5$, $\mu = 0.5$	$\lambda = 0.1$, $\mu = 0.1$	$\lambda = 0.01$, $\mu = 0.01$
AE-RED-CBM4D	$\lambda = 0.5$, $\mu = 0.5$	$\lambda = 0.5$, $\mu = 0.5$	$\lambda = 0.1$, $\mu = 0.1$	$\lambda = 0.01$, $\mu = 0.01$
AE-RED-UNLM	$\lambda = 0.5$, $\mu = 0.5$	$\lambda = 0.5$, $\mu = 0.5$	$\lambda = 0.1$, $\mu = 0.1$	$\lambda = 0.01$, $\mu = 0.01$
AE-RED-UBM4D	$\lambda = 0.5$, $\mu = 0.5$	$\lambda = 0.5$, $\mu = 0.5$	$\lambda = 0.1$, $\mu = 0.1$	$\lambda = 0.01$, $\mu = 0.01$

TABLE VIII
REAL DATA SETS: HYPERPARAMETER SETTINGS.

	Samson dataset	Jasper Ridge dataset
SUnSAL-TV	$\lambda = 3 \times 10^{-4}$, $\lambda_{\text{TV}} = 0.005$	$\lambda = 0.001$, $\lambda_{\text{TV}} = 0.002$
PnP-NMF	$\alpha = 0.01$, $\mu = 0.001$, $\lambda = 0.1$	$\alpha = 0.01$, $\mu = 0.01$, $\lambda = 0.05$
CNNAE	$\alpha = 3.5$	$\alpha = 3.5$
UnDIP	–	–
SNMF	$K = 7$	$K = 7$
CyCU-Net	$\beta = 0.5$, $\sigma = 1 \times 10^{-3}$, $\gamma = 1 \times 10^{-7}$	$\beta = 2$, $\sigma = 1 \times 10^{-2}$, $\gamma = 1 \times 10^{-7}$
AE-RED-CNLM	$\lambda = 0.01$, $\mu = 0.01$	$\lambda = 0.01$, $\mu = 0.01$
AE-RED-CBM4D	$\lambda = 0.01$, $\mu = 0.01$	$\lambda = 0.01$, $\mu = 0.01$
AE-RED-UNLM	$\lambda = 0.01$, $\mu = 0.01$	$\lambda = 0.01$, $\mu = 0.01$
AE-RED-UBM4D	$\lambda = 0.01$, $\mu = 0.01$	$\lambda = 0.01$, $\mu = 0.01$

[39] R. Borsoi, T. Imbiriba, and J. Bermudez, "Deep generative endmember modeling: An application to unsupervised spectral unmixing," *IEEE Trans. Comput. Imag.*, vol. 6, pp. 374–384, 2019.

[40] S. Shi, M. Zhao, L. Zhang, Y. Altmann, and J. Chen, "Probabilistic generative model for hyperspectral unmixing accounting for endmember variability," *IEEE Trans. Geosci. Remote Sens.*, vol. 60, 2022.

[41] M. Fahes, C. Kervazo, J. Bobin, and F. Tupin, "Unrolling PALM for sparse semi-blind source separation," in *Proc. Int. Conf. Learning Representations (ICLR)*, 2022.

[42] 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.

[43] 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.*, vol. 60, pp. 1–16, 2021.

[44] Z. Lai, K. Wei, and Y. Fu, "Deep plug-and-play prior for hyperspectral image restoration," *Neurocomputing*, vol. 481, pp. 281–293, 2022.

[45] R. Dian, S. Li, and X. Kang, "Regularizing hyperspectral and multispectral image fusion by CNN denoiser," *IEEE Trans. Neur. Net. Lear. Sys.*, vol. 32, no. 3, pp. 1124–1135, 2020.

[46] Y. Romano, M. Elad, and P. Milanfar, "The little engine that could: Regularization by denoising (RED)," *SIAM Journal on Imaging Sciences*, vol. 10, no. 4, pp. 1804–1844, 2017.

[47] P. Milanfar, "A tour of modern image filtering: New insights and methods, both practical and theoretical," *IEEE Signal Process. Mag.*, vol. 30, no. 1, pp. 106–128, 2012.

[48] B. Rasti, B. Koirala, P. Scheunders, and P. Ghamisi, "UnDIP: Hyperspectral unmixing using deep image prior," *IEEE Trans. Geosci. Remote Sens.*, vol. 60, pp. 1–15, 2021.

[49] A. Buades, B. Coll, and J.-M. Morel, "Non-local means denoising," *Image Processing On Line*, vol. 1, pp. 208–212, 2011.

[50] M. Maggioni, V. Katkovnik, K. Egiazarian, and A. Foi, "Nonlocal transform-domain filter for volumetric data denoising and reconstruction," *IEEE Trans. Image Process.*, vol. 22, no. 1, pp. 119–133, 2012.

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

[52] 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.