Model-Based Deep Autoencoder Networks for Nonlinear Hyperspectral Unmixing

Haoqing Li, Ricardo A. Borsoi, *Graduate Student Member, IEEE*, Tales Imbiriba,
Pau Closas, *Senior Member, IEEE*, José C. M. Bermudez, *Senior Member, IEEE*, and
Deniz Erdoğmuş, *Senior Member, IEEE*

***Abstract*— Autoencoder (AEC) networks have recently emerged as a promising approach to perform unsupervised hyperspectral unmixing (HU) by associating the latent representations with the abundances, the decoder with the mixing model, and the encoder with its inverse. AECs are especially appealing for nonlinear HU since they lead to unsupervised and model-free algorithms. However, existing approaches fail to explore the fact that the encoder should invert the mixing process, which might reduce their robustness. In this letter, we propose a model-based AEC for nonlinear HU by considering the mixing model a nonlinear fluctuation over a linear mixture. Different from previous works, we show that this restriction naturally imposes a particular structure to both the encoder and decoder networks. This introduces prior information in the AEC without reducing the flexibility of the mixing model. Simulations with synthetic and real data indicate that the proposed strategy improves nonlinear HU.**

***Index Terms*— Autoencoder (AEC), deep neural networks (NNs), hyperspectral data, nonlinear unmixing.**

I. Introduction

Hyperspectral unmixing (HU) consists of unveiling the spectral signatures of pure materials, called endmembers (EMs), and the proportions (also called *abundances*) with which they appear at every pixel of a hyperspectral image (HI) [1]. Although some HU methods assume the EMs spectra to be known *a priori* [2], [3], most applications require unsupervised algorithms, which estimates the EMs from the HI [4], [5]. The linear mixing model (LMM) represents the reflectance of an observed pixel as a linear combination of the reflectance of the spectral signatures of the EMs, weighted by their corresponding abundance proportion. However, the LMM fails to account for nonlinear interactions between different materials commonly seen in real scenes due to complex radiation scattering among several EMs [1].

HU strategies considering nonlinear mixtures can be generally divided into model-based and model-free methods. Model-based nonlinear HU assumes that the mixing process is known *a priori*. Common examples include algorithms based on, e.g., the bilinear mixing model (BLMM) or the post-nonlinear mixing model (PNMM) [1]. However, the mixing mechanisms can be complex in practice, and an appropriate model is rarely available. This motivated the consideration of model-free nonlinear HU, which employs more flexible nonlinear mixing models that are able to represent the mixing process in practice and can be learned directly from the observed HI [1], [6]. Examples include the use of graph-based approximate geodesic distances [7] and kernel-based algorithms [6], [8], the latter of which provides nonparametric function spaces that can represent arbitrary nonlinear mixtures.

Recently, the use of unsupervised neural networks (NNs) based on autoencoders (AECs) has become widespread in HU [9]–[11]. AECs consist of encoder–decoder structured NNs originally devised for nonlinear dimensionality reduction. By associating the low-dimensional latent representation of the input pixels with the abundances and the decoder structure of the network with the mixing model, HU can be performed by training the AEC on the observed HI [10]. The learned encoder is then applied to each image pixel to compute the abundances. Several AEC-based strategies have been proposed for linear HU, using denoising AECs to reduce noise and outliers [9], [12], [13], exploring sparsity constraints [11], [14], using a near-orthogonality prior over the abundances [15], [16] and convolutional architectures for spectral–spatial data processing [17], or using AECs as generative models to account for the spectral variability of the EMs [18], [19].

More recently, AEC architectures have also shown promising performance in nonlinear HU, leading to algorithms that are unsupervised and model-free. For instance, in [20], a decoder network was proposed as the composition of an EM matrix and nonlinear NN layers to learn postnonlinear mixtures. In [16], an AEC was proposed to account for bilinear mixtures by representing the abundances as the Hadamard product of two NN-generated estimates. However, the connection to the BLMM is not clear in this architecture. In [21], another architecture was presented by considering a decoder composed of a sum of a linear transformation and a multilayer NN to account for other types of nonlinearity in the mixture.

Despite achieving good performance, existing AEC-based nonlinear HU algorithms fail to properly explore the fact that the encoder should invert the mixing process. This may reduce their robustness, especially when nonlinear NNs with many degrees of freedom are considered. In this letter, we propose a model-based AEC for nonlinear HU by considering the mixing

Manuscript received February 16, 2021; revised April 4, 2021; accepted April 16, 2021. Date of publication May 4, 2021; date of current version December 28, 2021. This work was supported in part by the National Science Foundation under Award CNS-1815349 and Award ECCS-1845833; and in part by the National Council for Scientific and Technological Development (CNPq) under Grant 304250/2017-1, Grant 409044/2018-0, Grant 141271/2017-5, and Grant 204991/2018-8. *(Corresponding author: Haoqing Li.)*

Haoqing Li, Tales Imbiriba, Pau Closas, and Deniz Erdoğmuş are with the ECE Department, Northeastern University, Boston, MA 02115 USA (e-mail: li.haoq@northeastern.edu; t.imbiriba@northeastern.edu; closas@northeastern.edu; d.erdogmus@northeastern.edu).

Ricardo A. Borsoi is with EEL–UFSC, Florianópolis 88040-370, Brazil, and also with the Lagrange Laboratory (CNRS, OCA), Université Côte d'Azur, 6108 Nice, France (e-mail: raborsoi@gmail.com).

José C. M. Bermudez is with EEL–UFSC, Florianópolis 88040-370, Brazil.

Digital Object Identifier 10.1109/LGRS.2021.3075138

model a nonlinear fluctuation over a linear mixture. Different from previous works, we show that this restriction naturally imposes a particular structure to both the encoder and the decoder networks. This introduces important prior information into the AEC without reducing the flexibility of the mixing model. Simulations with synthetic and real data indicate that the proposed strategy significantly improves the performance of nonlinear HU compared to other state-of-the-art algorithms.

II. Problem Formulation

The LMM assumes that each L-band pixel $\boldsymbol{y}_n \in \mathbb{R}^L$, $n = 1, \ldots, N$, of an N-pixel HI, can be modeled as [1]

$$\boldsymbol{y}_n = \boldsymbol{M}\boldsymbol{a}_n + \boldsymbol{e}_n\,, \quad \text{s. t.} \quad \boldsymbol{a}_n \in \mathcal{S}^1 \tag{1}$$

where $\boldsymbol{M} \in \mathbb{R}^{L\times P}$ is a matrix whose columns are the P EM spectral signatures $\boldsymbol{m}_k$, $\boldsymbol{a}_n$ is the abundance vector, $\mathcal{S}^1 = \{\boldsymbol{x} \in \mathbb{R}^P : \boldsymbol{x} \geq \boldsymbol{0},\, \boldsymbol{1}^\top\boldsymbol{x} = 1\}$ denotes the unity simplex, and $\boldsymbol{e}_n$ is an additive noise term.

Despite its popularity, the LMM fails to represent nonlinear interactions between the different materials in the scene, which are commonly observed in real HIs [1]. This requires NLMMs, which can generally be represented as

$$\boldsymbol{y}_n = \boldsymbol{f}(\boldsymbol{M}, \boldsymbol{a}_n) + \boldsymbol{e}_n\,, \quad \text{s. t.} \quad \boldsymbol{a}_n \in \mathcal{S}^1 \tag{2}$$

in which $\boldsymbol{f} : \mathbb{R}^{L\times P} \times \mathbb{R}^P \to \mathbb{R}^L$ stands for different models for the interactions between the materials. Prominent examples include the BLMM (for macroscopic interactions), the PNMM (e.g., nonlinearities occurring between the scene and the sensor), and Hapke (for intimate mixtures) models [1].

Since it can be difficult to specify a precise model for $\boldsymbol{f}$ in advance of HU, the use of nonparametric approaches based on, e.g., kernel machines, has received a lot of attention [6], [8], [22]. Kernel methods have flexibility to model arbitrary nonlinear mixtures by learning $\boldsymbol{f}$ directly from the data. In this framework, a particularly interesting approach consists of assuming that the mixing process can be well represented as a nonlinear fluctuation $\boldsymbol{\psi}$ over the LMM [6]

$$\boldsymbol{y}_n = \boldsymbol{M}\boldsymbol{a}_n + \boldsymbol{\psi}(\boldsymbol{M}, \boldsymbol{a}_n) + \boldsymbol{e}_n, \quad \text{s. t.} \quad \boldsymbol{a}_n \in \mathcal{S}^1. \tag{3}$$

While the connection between (2) and (3) may not be straightforward, model (3) allows easier control of the degree of nonlinearity in the model by penalizing the contribution of $\boldsymbol{\psi}$ during the HU process. In [6], an HU methodology was proposed using model (3) by constraining $\boldsymbol{\psi}$ to belong to a reproducing kernel Hilbert space, which allowed an efficient solution as a convex optimization problem (i.e., a least-squares support vector regression problem). However, this methodology is purely supervised and cannot estimate the EMs directly from the HI.

A. AEC-Based Unsupervised Nonlinear HU

More recently, unsupervised approaches based on AECs have been proposed for nonlinear HU. Such approaches differ from the AEC strategies used for linear HU in the sense that the decoder, which is based on the mixing model, must be designed to incorporate the nonlinearity seen in, e.g., (2) or (3). For instance, in [20], a network was proposed based on the postnonlinear model, where the decoder NN is formed by a composition of the EM matrix and nonlinear layers. In [21], a similar idea was used based on model (3), where the decoder NN was a linear combination of a linear layer (representing the EMs) and a nonlinear NN (representing the nonlinear interactions). However, both these works use generic nonlinear network architectures for the encoder, which do not by themselves guarantee a proper inversion of the nonlinear HU process. Being HU an inference problem, the objective is to retrieve both the abundances and EMs by fitting the model to a given HI. This contrasts with typical machine learning applications, where training and test sets are required to be distinct. Hence, the importance of structured models guarantees the physical interpretability of the retrieved latent representations.

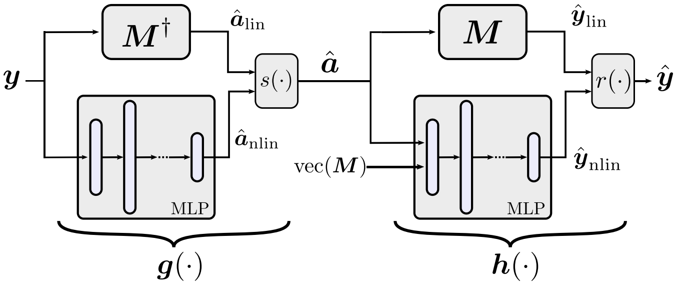

Fig. 1. Proposed model-based AEC solution.

B. Importance of the Encoder for HU

Although some works have explored the influence of the encoder, its potential to improve the quality and robustness of AEC-based HU has not been fully utilized. For instance, early architectures considered tied weights for the encoder and decoder [9], which severely limited the performance. More recent works use almost exclusively untied weights [11], using more flexible NNs as the encoder. Denoising AECs have also been included as the first layers of the decoder in order to act as a preprocessing step before HU and improve the results [9], [12]. However, these works do not fully explore the facts that, while the decoder should match the mixing process, the encoder should be constrained to represent its inverse. In Section III, we will leverage this knowledge to design the proposed model-based AEC.

III. Proposed Solution

In this work, we propose to leverage knowledge about the physical mixing process to derive an AEC architecture, which is better able to represent both the nonlinear mixing process and its inverse to perform HU in real-world applications. Our contributions in this step are twofold, related to the design of both the encoder and the decoder. An illustrative depiction is shown in Fig. 1.

Inspired by (3), we consider an AEC decoder design with an additive linear and nonlinear part, which is similar to the one used in [21]. The NN weights for the linear part of the decoder are directly associated with the EM signatures $\boldsymbol{M}$. Moreover, we introduce both the abundances and the EM signatures as inputs to the nonlinear part of the decoder. Therefore, by employing an NN architecture with high representation capacity, we can model arbitrary mixtures. This leads to a decoder $\boldsymbol{h} : \mathbb{R}^P \to \mathbb{R}^L$ of the form

$$\boldsymbol{h}(\boldsymbol{a}_n) = r\big(\boldsymbol{M}\boldsymbol{a}_n + \boldsymbol{\omega}_D(\boldsymbol{a}_n, \boldsymbol{M};\, \boldsymbol{W}_D)\big) \tag{4}$$

where $\boldsymbol{\omega}_D$ is a nonlinear function in the decoder (e.g., an MLP) with parameters $\boldsymbol{W}_D$ representing the nonlinear part of the mixing process, and $r : \mathbb{R}^L \to \mathbb{R}_+^L$ is a function that projects the decoder results to the nonnegative orthant. Note that, unlike in multikernel based methods, such as [6], which consider nonlinearities in the mixing process to be only a function of $\boldsymbol{M}$, $\boldsymbol{\omega}_D$ is a function of both the abundances and the EM signatures, which makes it more general.

The second part of the proposed design concerns the encoder, which we denote by $\boldsymbol{g} : \mathbb{R}^L \to \mathbb{R}^P$. This has received far less attention from previous works, as discussed in Section II-B. Although most works adopt general linear or nonlinear NNs to design the encoder function, a better design can be obtained if we consider that the encoder should in principle be close to $\boldsymbol{h}^{-1}$. For the case of linear HU (i.e., in which $\boldsymbol{\omega}_D \equiv \boldsymbol{0}$), this translates directly into a special kind of tied architecture, in which the encoder becomes $\boldsymbol{g} \approx \boldsymbol{M}^\dagger$ [11], where † denotes the pseudoinverse operator. Note that, unlike previous works that consider tied architectures with $\boldsymbol{g} \approx \boldsymbol{M}^\top$ [9], the architecture that we mentioned is tied by the pseudoinverse, which is physically more reasonable. This has been used to motivate untied architectures, in which the encoder is left unconstrained [11]. However, this important knowledge has not been further employed to design or constrain neither the linear nor the nonlinear AEC encoder architecture due to the difficulty it introduces. Nonetheless, we can leverage this idea for nonlinear HU to make the method more principled and robust while proposing a tractable training procedure.

We propose to consider an encoder of the following form:

$$\boldsymbol{g}(\boldsymbol{y}_n) = s\big(\mathrm{diag}(\boldsymbol{\alpha})\boldsymbol{M}^\dagger \boldsymbol{y}_n + \boldsymbol{\omega}_E(\boldsymbol{y}_n; \boldsymbol{W}_E)\big) \tag{5}$$

where $\boldsymbol{\omega}_E$ is a function parameterized on $\boldsymbol{W}_E$ representing the nonlinear part of the encoder, and $s : \mathbb{R}^P \to \mathcal{S}^1$ is a function that maps the combinations from the linear and nonlinear abundance branch estimates to the unity simplex $\mathcal{S}^1 = \{\boldsymbol{x} \in \mathbb{R}^P : \boldsymbol{x} \geq \boldsymbol{0}, \boldsymbol{1}^\top \boldsymbol{x} = 1\}$ to ensure that the estimated abundances are physically meaningful. Parameter $\boldsymbol{\alpha} \in \mathbb{R}_+^P$ balances the contributions of the linear and nonlinear parts of the encoder and is also a learnable parameter. Note that the nonlinear parts of the encoder and decoder are closely related, in which, to achieve small reconstruction errors, the contributions of $\boldsymbol{\omega}_E$ and $\boldsymbol{\omega}_D$ must be similar.

To illustrate this, suppose that a pixel $\boldsymbol{y}_n$ is generated from the decoder model with abundances $\boldsymbol{a}_n$, r and s are the identity functions, $\boldsymbol{\alpha} = \boldsymbol{1}$, and $\boldsymbol{\omega}_D$ and $\boldsymbol{\omega}_E$ belong to normed function spaces. Then, to accurately reconstruct the abundances, we need $\|\boldsymbol{a}_n - \boldsymbol{g}(\boldsymbol{h}(\boldsymbol{a}_n))\|$ to be small. Using the reverse triangle inequality, this can generally be written as

$$\begin{aligned}
&\big\|\boldsymbol{a}_n - \boldsymbol{g}(\boldsymbol{h}(\boldsymbol{a}_n))\big\| \\
&\quad= \big\|\boldsymbol{a}_n - \boldsymbol{M}^\dagger(\boldsymbol{M}\boldsymbol{a}_n + \boldsymbol{\omega}_D(\boldsymbol{a}_n, \boldsymbol{M}; \boldsymbol{W}_D)) - \boldsymbol{\omega}_E(\boldsymbol{h}(\boldsymbol{a}_n); \boldsymbol{W}_E)\big\| \\
&\quad= \big\| - \boldsymbol{M}^\dagger \boldsymbol{\omega}_D(\boldsymbol{a}_n, \boldsymbol{M}; \boldsymbol{W}_D) - \boldsymbol{\omega}_E(\boldsymbol{h}(\boldsymbol{a}_n); \boldsymbol{W}_E)\big\| \\
&\quad\geq \Big|\big\|\boldsymbol{\omega}_E(\boldsymbol{h}(\boldsymbol{a}_n); \boldsymbol{W}_E)\big\| - \big\|\boldsymbol{M}^\dagger \boldsymbol{\omega}_D(\boldsymbol{a}_n, \boldsymbol{M}; \boldsymbol{W}_D)\big\|\Big|
\end{aligned} \tag{6}$$

in which

$$\big\|\boldsymbol{M}^\dagger \boldsymbol{\omega}_D(\boldsymbol{a}_n, \boldsymbol{M}; \boldsymbol{W}_D)\big\| \leq \big\|\boldsymbol{M}^\dagger\big\| \big\|\boldsymbol{\omega}_D(\boldsymbol{a}_n, \boldsymbol{M}; \boldsymbol{W}_D)\big\|. \tag{7}$$

This implies that, if $\|\boldsymbol{\omega}_D\|$ is small (small amounts of nonlinearity), we must necessarily have $\|\boldsymbol{\omega}_E\|$ small to obtain a small abundance reconstruction error. More generally, good abundance reconstruction requires the contributions of $\boldsymbol{\omega}_E$ and $\boldsymbol{M}^\dagger \boldsymbol{\omega}_D$ to be similar for any amount of nonlinearity. Therefore, if the norms of $\boldsymbol{\omega}_D$ and $\boldsymbol{\omega}_E$ are bounded above by the norm of the NN weights, we can account for different degrees of nonlinearity in a principled way by appropriately regularizing $\boldsymbol{W}_E$ and $\boldsymbol{W}_D$. We call the proposed method Model-based AutoenCoder for hyperspectral Unmixing (MAC-U).

A. Cost Function

The training process now consists of determining the proposed model-based encoder and decoder networks, $\boldsymbol{g}$ and $\boldsymbol{h}$, based on the set of available image pixels $\boldsymbol{y}_n$, $n = 1, \ldots, N$. Note that, HU being an inference problem, we only have a single data set (i.e., one HI) with N pixels to both learn the model parameters and perform inference of the abundances using the AEC (this is in contrast to other traditional AECs that are trained and tested on separate data sets). Thus, the NN parameters must be learned using a single HI. A particular difficulty in learning the model-based AEC described in (4) and (5) is that it involves both $\boldsymbol{M}$ and its pseudoinverse $\boldsymbol{M}^\dagger$. In order to obtain a more tractable architecture, let us rewrite (5) equivalently as

$$\boldsymbol{g}(\boldsymbol{y}_n) = s\big(\mathrm{diag}(\boldsymbol{\alpha})\boldsymbol{Q}\boldsymbol{y}_n + \boldsymbol{\omega}_E(\boldsymbol{y}_n; \boldsymbol{W}_E)\big) \ \text{ s.t. } \ \boldsymbol{Q} = \boldsymbol{M}^\dagger. \tag{8}$$

The constraint in (8) is, however, difficult to enforce even for moderate numbers of EMs P. Thus, we rewrite it as

$$\boldsymbol{Q} = \boldsymbol{M}^\dagger = \big(\boldsymbol{M}^\top \boldsymbol{M}\big)^{-1} \boldsymbol{M}^\top \Rightarrow \boldsymbol{M}^\top \boldsymbol{M}\boldsymbol{Q} = \boldsymbol{M}^\top. \tag{9}$$

Denoting the network parameters by $\boldsymbol{W} = \{\boldsymbol{W}_D, \boldsymbol{W}_E\}$ and the training variables by $\boldsymbol{\Theta} = \{\boldsymbol{M}, \boldsymbol{W}, \boldsymbol{Q}, \boldsymbol{\alpha}\}$, the cost function can be written as

$$\begin{aligned}
\mathcal{L}(\boldsymbol{\Theta}) = \mathbb{E}_{\boldsymbol{y}\sim\mathcal{D}}\big\{\|\boldsymbol{y} - \boldsymbol{h}(\boldsymbol{g}(\boldsymbol{y}))\|^2\big\} + \mathcal{R}_{\mathcal{W}}(\boldsymbol{W}) + \mathcal{R}_{\mathcal{M}}(\boldsymbol{M}) \\
+ \lambda_Q \big\|\boldsymbol{M}^\top \boldsymbol{M}\boldsymbol{Q} - \boldsymbol{M}^\top\big\|_F^2
\end{aligned} \tag{10}$$

where the expectation in the first term is taken with respect to the empirical distributions of the image pixels, supported at $\{\boldsymbol{y}_1, \ldots, \boldsymbol{y}_N\}$. The constraint (9) was introduced into the cost function in the form of an additive term. Function $\mathcal{R}_{\mathcal{W}}(\boldsymbol{W}) \triangleq \lambda_{\mathcal{W}}\big(\|\boldsymbol{W}_D\|_F^2 + \|\boldsymbol{W}_E\|_F^2\big)$ is a regularization, which governs the nonlinear contributions to the encoder and the decoder, while $\mathcal{R}_{\mathcal{M}}(\boldsymbol{M}) \triangleq \lambda_{\mathcal{M}}(\boldsymbol{m}_k^\top \boldsymbol{m}_k^{(0)})/(\|\boldsymbol{m}_k\|\|\boldsymbol{m}_k^{(0)}\|)$ constrains the spectral angle between the updated EMs and the initialization $\boldsymbol{M}_0$. Parameters $\lambda_Q, \lambda_{\mathcal{W}}, \lambda_{\mathcal{M}} \in \mathbb{R}_+$ balance the contributions of the regularizing terms in the cost function.

B. NN Architecture and Cost Function Optimization

For the proposed MAC-U method, we considered the following NN architectures for the nonlinear part of the encoder and decoder blocks, adapted from the ones used in [21]. For $\boldsymbol{\omega}_E$, seven fully connected layers were used, with the leaky ReLU activation function and no bias term in the last layer. The layers contained L, $2L$, $L/2$, $L/4$, $4P$, P, and P neurons, with noninteger values rounded up. For $\boldsymbol{\omega}_D$, five fully connected layers were used, with the leaky ReLU activation function and no bias term in the last layer. The layers contained $P(L+1)$, PL, L, L, and L neurons. Function $r(\cdot)$ was implemented by a ReLU activation, and function $s(\cdot)$ was implemented by the normalized absolute value rectification mapping. $\mathcal{L}$ was minimized using the stochastic optimization method, Adam [23], with hyperparameters set as:

TABLE I

QUANTITATIVE RESULTS FOR DATA CUBES DC1 AND DC2

DC1 data cube					
	BLMM		PNMM		
Method	RMSE$_{\boldsymbol{A}}$	RMSE$_{\boldsymbol{Y}}$	RMSE$_{\boldsymbol{A}}$	RMSE$_{\boldsymbol{Y}}$	Time
FCLS	0.2427	0.0904	0.1608	0.0748	0.85
K-Hype	0.1859	**0.0795**	0.1523	**0.0740**	7.54
CDA-NL	0.1965	0.0808	0.1628	0.0745	10.75
MAC-U (Prop.)	**0.1050**	0.0810	**0.0795**	0.0747	121.65
MF-AEC	0.3460	0.0891	0.3693	0.0793	163.94
NF-AEC	0.2230	0.0832	0.1047	0.0787	75.34
DC2 data cube					
	BLMM		PNMM		
Method	RMSE$_{\boldsymbol{A}}$	RMSE$_{\boldsymbol{Y}}$	RMSE$_{\boldsymbol{A}}$	RMSE$_{\boldsymbol{Y}}$	Time
FCLS	0.1674	0.1061	0.0636	0.0769	0.85
K-Hype	0.0992	**0.0770**	0.0976	**0.0759**	4.13
CDA-NL	0.0910	0.0780	0.0616	0.0765	6.18
MAC-U (Prop.)	**0.0864**	0.0895	**0.0512**	0.0769	103.61
MF-AEC	0.3846	0.3943	0.1913	0.3287	86.15
NF-AEC	0.2337	0.1297	0.1841	0.0876	80.10

gradientDecayFactor = 0.9, squaredGradientDecayFactor = 0.95, and miniBatchSize = 128. Other hyperparameters were left as the default values in MATLAB. Training was performed for at least one full epoch and stopped when the relative change of $\mathcal{L}$ between two iterations was smaller than 0.01.

IV. EXPERIMENTS

This section illustrates the performance of the proposed method using simulations with both synthetic and real data. The proposed MAC-U method is compared to the fully constrained least squares (FCLS), K-Hype [6], and CDA-NL [24]. We also compare MAC-U with two other AEC-based methods. The first is a completely model-free AEC architecture (i.e., where neither the encoder nor the decoder has linear parts), which we call MF-AEC. The second is based on our implementation of the architecture proposed in [21], which uses a linear model and nonlinear fluctuation only in the decoder (i.e., with a model-free encoder), which we call NF-AEC. Both MF-AEC and NF-AEC were implemented using the same framework and code as MAC-U, with the NN architectures described in Section III-B. In all experiments, EMs extracted from the observed HI using the VCA algorithm [25] were used for FCLS, K-Hype, and CDA-NL, and as initialization for the different AEC strategies. The performances of the methods were evaluated using the root mean squared error (RMSE) between the estimated abundance maps (RMSE$_{\boldsymbol{A}}$) and between the reconstructed images (RMSE$_{\boldsymbol{Y}}$). The RMSE is defined as RMSE$_{\boldsymbol{X}} = \sqrt{\|\boldsymbol{X} - \boldsymbol{X}^*\|_F^2 / N_{\boldsymbol{X}}}$, where $N_{\boldsymbol{X}}$ denotes the number of elements in $\boldsymbol{X}$.

A. Synthetic Data

Two synthetic data sets were considered, namely, Data Cube 1 (DC1), with 10^4 pixels, and Data Cube 2 (DC2), with 2500 pixels. Both DC1 and DC2 contained $P = 3$ EMs with 224 bands extracted from the USGS Spectral Library. The abundance maps were sampled from a Dirichlet distribution for DC1 and from a spatially correlated Gaussian random field for DC2. The pixel reflectance values were generated using two nonlinear mixture models, namely, the BLMM: $\boldsymbol{y}_n = \boldsymbol{M}\boldsymbol{a}_n + \sum_{i=1}^{P-1}\sum_{j=i+1}^{P} a_{n,i}a_{n,j}\boldsymbol{m}_i \odot \boldsymbol{m}_j + \boldsymbol{e}_n$, where $\odot$ is the Hadamard product, and the PNMM: $\boldsymbol{y}_n = (\boldsymbol{M}\boldsymbol{a}_n)^{0.7} + \boldsymbol{e}_n$, where the exponent is applied elementwise. The Gaussian noise was added through $\boldsymbol{e}_n$ to result in a signal-to-noise ratio of 20 dB. The parameters of the methods for each data set were selected using a grid search within the ranges discussed in the original publications. For K-Hype, the parameter was selected among the values $\mu \in \{0.001, 0.002, 0.005, 0.01, 0.02, 0.1, 1\}$. For MAC-U and its variants, the parameters were selected by performing a grid search using the following values: $\lambda_{\mathcal{Q}}, \lambda_{\mathcal{W}}, \lambda_{\mathcal{M}} \in \{10^{-6}, 10^{-2}, 1\}$ and the learning rate $\gamma \in \{10^{-6}, 10^{-4}\}$.

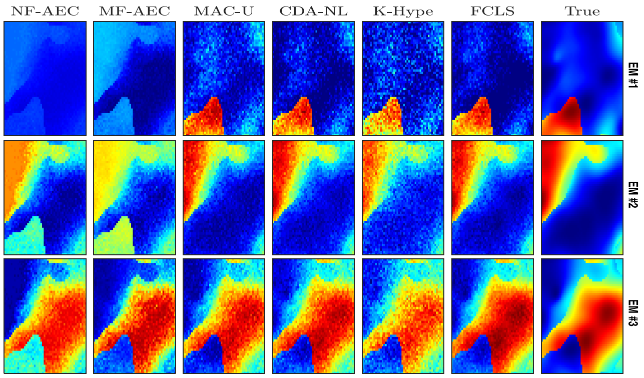

Fig. 2. Abundance maps for DC2 for the PNMM model.

The objective results are summarized in Table I. Fig. 2 presents the abundance maps only for DC2 with the PNMM due to space limitations. The proposed MAC-U outperformed the competing algorithms for all datacubes and nonlinearity models. CDA-NL also provided generally good results for DC1, while the performance of K-Hype varied according to the nonlinearity model. Moreover, the model-free architecture MF-AEC did not perform well, while NF-AEC (where the model is only enforced in the decoder) presented intermediate results. The proposed model-based architecture significantly improved the unmixing results among the AEC-based solutions. The RMSE$_{\boldsymbol{Y}}$ values for MAC-U were slightly larger than those of K-Hype but smaller compared to the other AEC solutions. However, we note that RMSE$_{\boldsymbol{Y}}$ is not directly related to the abundance reconstruction and, thus, not a good metric to evaluate the unmixing performance, especially when flexible models are considered. A visual inspection of the abundance maps in Fig. 2 corroborates the objective results. The average execution times of MAC-U, as shown in Table I, were significantly larger than the ones of KHype and CDA-NL but compatible with those of the other AEC-based strategies.

B. Real Data

To evaluate the algorithms with real data, we considered a subscene of the Urban HI [22], with $L = 162$ bands. This subscene is known to contain three EMs: asphalt, vegetation, and ground, where multiple scattering is expected to occur. Fig. 3 presents the abundance maps obtained with all the competing methods. By visual inspection, one can notice that almost all methods present well-defined and coherent abundance maps with exception of MF-AEC, which apparently merged the asphalt and ground EMs but presented a coherent map for the Tree abundance. This behavior is expected due to the nonsupervised and nonstructured nature of this model. Regarding the remaining methods, we can see an advantage of the proposed MAC-U algorithm for both the Asphalt and Ground EMs, while K-Hype seems

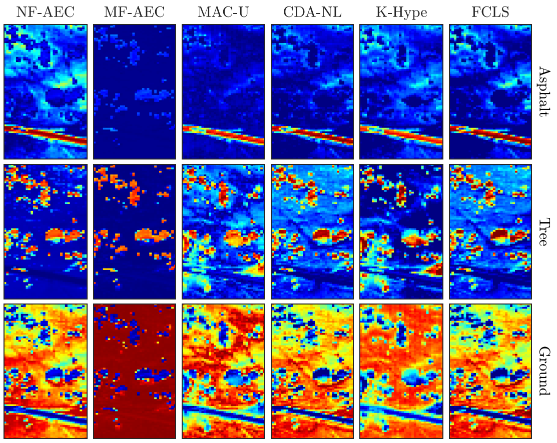

Fig. 3. Abundance maps of the urban scene.

TABLE II

QUANTITATIVE RESULTS FOR THE URBAN SCENE

Method	FCLS	K-Hype	CDA-NL	MAC-U	MF-AEC	NF-AEC
RMSE$_{\boldsymbol{Y}}$	0.02279	**0.00610**	0.00626	0.03756	0.12523	0.04403
Time	0.41	2.41	38.51	168.65	236.30	82.88

to have provided a marginally better Tree abundance map, with energy concentrated in areas known to have vegetation. Nevertheless, the remaining methods, including the proposed MAC-U, also provide relatively accurate/coherent maps for the Tree EM. The estimated linear scaling coefficients $\boldsymbol{\alpha} = [1.0567, 0.9523, 1.0402]^\top$ indicate strong contributions of the linear model. We highlight, however, that $\boldsymbol{\alpha}$ values cannot directly measure the contribution of the linear parcel of the model since the parameters of the nonlinear branch can grow to compensate posterior scaling parameters. The reconstruction errors of MAC-U, as shown in Table II, were comparable to those of FCLS and smaller than those of the other AEC-based architectures. However, the connection between small RMSE$_{\boldsymbol{Y}}$ and abundance reconstruction is not direct.

V. Conclusion

In this letter, a model-based AEC network was proposed for nonlinear HU. Considering the mixing model composed of a nonlinear fluctuation over a linear mixture, the proposed AEC can represent arbitrary nonlinear mixtures. Moreover, different from previous approaches, the fact that the encoder should invert the mixing process was explicitly explored in this work. We showed that this restriction naturally imposes a particular structure to both the encoder and the decoder networks, which explicitly makes use of the pseudoinverse of the EM matrix. This introduced prior information into the AEC without reducing the flexibility of the mixing model. Simulations with synthetic and real data showed that the proposed strategy can improve the quality of nonlinear HU.

References

[1] 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, Jan. 2014.

[2] M.-D. Iordache, J. Bioucas-Dias, and A. Plaza, "Sparse unmixing of hyperspectral data," *IEEE Trans. Geosci. Remote Sens.*, vol. 49, no. 6, pp. 2014–2039, Jun. 2011.

[3] R. A. Borsoi, T. Imbiriba, J. C. M. Bermudez, and C. Richard, "A fast multiscale spatial regularization for sparse hyperspectral unmixing," *IEEE Geosci. Remote Sens. Lett.*, vol. 16, no. 4, pp. 598–602, Apr. 2019.

[4] R. A. Borsoi, T. Imbiriba, and J. C. M. Bermudez, "A data dependent multiscale model for hyperspectral unmixing with spectral variability," *IEEE Trans. Image Process.*, vol. 29, pp. 3638–3651, 2020.

[5] Y. Qian, S. Jia, J. Zhou, and A. Robles-Kelly, "Hyperspectral unmixing via $L_1/2$ sparsity-constrained nonnegative matrix factorization," *IEEE Trans. Geosci. Remote Sens.*, vol. 49, no. 11, pp. 4282–4297, Nov. 2011.

[6] J. Chen, C. Richard, and P. Honeine, "Nonlinear unmixing of hyperspectral data based on a linear-mixture/nonlinear-fluctuation model," *IEEE Trans. Signal Process.*, vol. 61, no. 2, pp. 480–492, Jan. 2013.

[7] R. Heylen, D. Burazerovic, and P. Scheunders, "Non-linear spectral unmixing by geodesic simplex volume maximization," *IEEE J. Sel. Topics Signal Process.*, vol. 5, no. 3, pp. 534–542, Jun. 2011.

[8] T. Imbiriba, J. C. M. Bermudez, C. Richard, and J.-Y. Tourneret, "Nonparametric detection of nonlinearly mixed pixels and endmember estimation in hyperspectral images," *IEEE Trans. Image Process.*, vol. 25, no. 3, pp. 1136–1151, Mar. 2016.

[9] R. Guo, W. Wang, and H. Qi, "Hyperspectral image unmixing using autoencoder cascade," in *Proc. 7th Workshop Hyperspectral Image Signal Processing, Evol. Remote Sens. (WHISPERS)*, Tokyo, Japan, Jun. 2015, pp. 1–4.

[10] B. Palsson, J. Sigurdsson, J. R. Sveinsson, and M. O. Ulfarsson, "Hyperspectral unmixing using a neural network autoencoder," *IEEE Access*, vol. 6, pp. 25646–25656, 2018.

[11] 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, Mar. 2019.

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

[13] 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, Jul. 2019.

[14] 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, Jan. 2019.

[15] Z. Dou, K. Gao, X. Zhang, H. Wang, and J. Wang, "Hyperspectral unmixing using orthogonal sparse prior-based autoencoder with hyper-Laplacian loss and data-driven outlier detection," *IEEE Trans. Geosci. Remote Sens.*, vol. 58, no. 9, pp. 6550–6564, Sep. 2020.

[16] Z. Dou, K. Gao, X. Zhang, H. Wang, and J. Wang, "Blind hyperspectral unmixing using dual branch deep autoencoder with orthogonal sparse prior," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, Barcelona, Spain, May 2020, pp. 2428–2432.

[17] 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, Jan. 2021.

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

[19] R. A. Borsoi, T. Imbiriba, J. C. M. Bermudez, and C. Richard, "Deep generative models for library augmentation in multiple endmember spectral mixture analysis," *IEEE Geosci. Remote Sens. Lett.*, early access, Jul. 22, 2020, doi: 10.1109/LGRS.2020.3007161.

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

[21] M. Zhao, M. Wang, J. Chen, and S. Rahardja, "Hyperspectral unmixing via deep autoencoder networks for a generalized linear-mixture/nonlinear-fluctuation model," 2019, *arXiv:1904.13017*. [Online]. Available: http://arxiv.org/abs/1904.13017

[22] R. A. Borsoi, T. Imbiriba, J. C. M. Bermudez, and C. Richard, "A blind multiscale spatial regularization framework for kernel-based spectral unmixing," *IEEE Trans. Image Process.*, vol. 29, pp. 4965–4979, 2020.

[23] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *Proc. Int. Conf. Learn. Represent. (ICLR)*, 2015, pp. 1–15.

[24] A. Halimi, P. Honeine, and J. M. Bioucas-Dias, "Hyperspectral unmixing in presence of endmember variability, nonlinearity, or mismodeling effects," *IEEE Trans. Image Process.*, vol. 25, no. 10, pp. 4565–4579, Oct. 2016.

[25] 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.