Entropic Descent Archetypal Analysis for Blind Hyperspectral Unmixing

Alexandre Zouaoui, *Graduate Student Member, IEEE*, Gedeon Muhawenayo, Behnood Rasti, *Senior Member, IEEE*, Jocelyn Chanussot, *Fellow, IEEE*, and Julien Mairal, *Senior Member, IEEE*

***Abstract*— In this paper, we introduce a new algorithm based on archetypal analysis for blind hyperspectral unmixing, assuming linear mixing of endmembers. Archetypal analysis is a natural formulation for this task. This method does not require the presence of pure pixels (i.e., pixels containing a single material) but instead represents endmembers as convex combinations of a few pixels present in the original hyperspectral image. Our approach leverages an entropic gradient descent strategy, which (i) provides better solutions for hyperspectral unmixing than traditional archetypal analysis algorithms, and (ii) leads to efficient GPU implementations. Since running a single instance of our algorithm is fast, we also propose an ensembling mechanism along with an appropriate model selection procedure that make our method robust to hyper-parameter choices while keeping the computational complexity reasonable. By using six standard real datasets, we show that our approach outperforms state-of-the-art matrix factorization and recent deep learning methods. We also provide an open-source PyTorch implementation: https://github.com/inria-thoth/EDAA.**

***Index Terms*— Hyperspectral image, remote sensing, blind spectral unmixing, non-negative matrix factorization, archetypal analysis, entropic gradient descent.**

I. Introduction

HYPERSPECTRAL imaging (HSI) [1], [3], [4], [5] consists of measuring the electromagnetic spectrum in a scene by using multiple narrow spectral bands. Thanks to its richer spectral information compared to traditional RGB images, HSI enables more accurate materials identification, leading to a broad range of applications including crop monitoring in agriculture [6], waste sorting [7], food safety inspection [8], or mineralogy [9].

Manuscript received 12 September 2022; revised 5 May 2023 and 13 July 2023; accepted 18 July 2023. Date of publication 8 August 2023; date of current version 17 August 2023. This work was supported in part by the European Research Council (ERC) through the SOLARIS Project under Grant 714381 and in part by the Agence Nationale de la Recherche (ANR) 3IA Multidisciplinary Institute in Artificial Intelligence (MIAI)@Grenoble Alpes under Grant ANR-19-P3IA-0003. The associate editor coordinating the review of this manuscript and approving it for publication was Dr. Emanuele Salerno. *(Corresponding author: Alexandre Zouaoui.)*

Alexandre Zouaoui, Jocelyn Chanussot, and Julien Mairal are with Inria, CNRS, Grenoble INP, LJK, Université Grenoble Alpes, 38000 Grenoble, France (e-mail: alexandre.zouaoui@inria.fr).

Gedeon Muhawenayo was with Inria, CNRS, Grenoble INP, LJK, Université Grenoble Alpes, 38000 Grenoble, France. He is now with the Ira A. Fulton Schools of Engineering, Arizona State University, Tempe, AZ 85283 USA.

Behnood Rasti is with Helmholtz-Zentrum Dresden-Rossendorf (HZDR), 01328 Dresden, Germany.

This article has supplementary downloadable material available at https://doi.org/10.1109/TIP.2023.3301769, provided by the authors.

Digital Object Identifier 10.1109/TIP.2023.3301769

Remote sensing [10], [11], such as airborne or satellite imagery, yields hyperspectral images whose pixels capture several objects or materials. As such, each pixel can include several pure spectral components (called *endmembers*), mixed in different proportions [12]. Any further analysis hence requires identifying and disentangling endmembers present in a scene before estimating their respective proportions, or fractional *abundances*, within each pixel of the HSI [13]. Since the endmembers spectrum signatures are not known beforehand and must be estimated from data, this operation is named *blind* hyperspectral unmixing (HU) [14], [15] owing to its link with blind source separation [16].

In this paper, we adopt a linear mixing model since it is often relevant in remote sensing scenes where mixtures occur between macroscopic materials. Therefore, we assume that each observed pixel can be represented as a linear combination of endmembers and some additive noise. In other words, we are interested in tackling unsupervised linear HU [13].

Further assumptions on the nature of endmembers are generally needed to estimate meaningful spectra. For instance, it can be assumed that there exists at least one pure pixel for each material present in the scene. The problem then requires finding these pure pixels within the original image. The pure pixel assumption is at the core of several geometrical endmember extraction methods including pixel purity index (PPI) [17], N-FINDR [18] and vertex component analysis (VCA) [19]. Once endmembers have been extracted, abundances can be estimated by minimizing the least squares errors between the original input spectra and the linearly reconstructed spectra as long as the abundances fractions satisfy the two physical constraints stating that they should be non-negative and sum to one for each pixel [20]. That being said, pure pixels are often missing in real scenarios. In the absence of pure pixels and in the case of linear models, endmembers and abundances can be simultaneously estimated by solving a constrained or penalized non-negative matrix factorization problem (NMF) [21]. For example, the authors of [22] have proposed a formulation that involves a data fidelity term and a minimum volume regularization term on endmembers, whose minimization consists in alternating between solving for endmembers and abundances.

In this work, we do not assume the existence of pure pixels as they are often missing in real data, since, for instance, the spectral signatures of endmembers in HSI can be significantly affected by various changes in atmospheric, illumination, and environmental conditions within the scene [23]. There are multiple strategies that can be employed to tackle

spectral variability such as (i) augmenting the linear mixing model [24], (ii) using a tensor-based approach, like the Sparsity-Enhanced Convolutional Decomposition (SeCoDe) method [25], (iii) performing unmixing in orthogonal subspaces as in [26]. Instead, we mitigate the effect of spectral variability by (i) normalizing each pixel by the ℓ_2-norm of its spectrum as a pre-processing step and (ii) modeling endmembers as convex combinations of pixels present in the scene. Not only HSI pixels are linear combinations of the estimated endmembers under the linear mixing model, but the estimated endmembers are also convex combinations of pixels. This corresponds to the archetypal analysis (AA) formulation introduced by Cutler and Breiman in [27]. AA has the advantage to be more interpretable than NMF because the basis elements (*i.e. endmembers*) are directly constructed from the data points (*i.e. pixels*). In addition, since the estimated endmembers spectral signatures generally correspond to averaging the contributions of several pixels, the resulting estimation appears to be more robust to noise and spectral variability than pure pixel methods that only select one pixel per endmember. However, AA usually suffers from a high data fitting error because the basis elements are constrained to be contained in the convex cone of the data points [28].

The contributions and innovations of this article are as follows:

1) We propose a new hyperspectral unmixing algorithm relying on entropic gradient descent for archetypal analysis. Our approach (i) provides solutions for hyperspectral unmixing as good as traditional alternating optimization schemes based on projected gradient methods or active set algorithms, and (ii) allows more efficient GPU implementations.
2) The efficiency of our method enables us to make a key practical contribution, consisting of an ensembling mechanism along with an appropriate model selection procedure, which makes our method almost parameter-free and thus easy to use (the only sensitive parameter is the number of endmembers we want to estimate).
3) Our approach, available in an open-source package,[1] outperforms state-of-the-art matrix factorization and deep learning methods on six standard real datasets.

The remainder of this paper is organized as follows. Section II presents related works for unsupervised linear hyperspectral unmixing. Section III introduces our method. Section IV presents experimental results highlighting the performance of our proposed approach. Finally, we conclude the article and underline future research directions in Section V.

II. Related Work on Unsupervised Linear Hyperspectral Unmixing

In this section, we present the framework of non-negative matrix factorization (NMF) before introducing archetypal analysis (AA). Finally, we mention widely used deep learning architectures that tackle blind hyperspectral unmixing.

[1]Code is available at https://github.com/inria-thoth/EDAA

A. Non-Negative Matrix Factorization (NMF)

NMF [21], [29] is popular for hyperspectral unmixing. It consists of factorizing a matrix representing the HSI signal into the product of two matrices with non-negative entries, where one factor carries the endmembers whereas the other one represents the abundances for each pixel (typically with the constraint that abundances sum to one). Several variants of NMF have been proposed for this task. For instance, the method of minimum volume constrained non-negative matrix factorization (MVC-NMF) of [30] uses a minimum volume term for endmembers that effectively gets rid of the pure pixel assumption. In minimum dispersion constrained NMF (MiniDisCo) [31], the regularization function is called *dispersion*, which encourages endmembers with minimum variance to avoid degenerate solutions and improve the unmixing robustness to flat spectra. Other approaches design regularization functions for abundances such as [32], [33], and [34]. NMF typically uses a least-squares objective function, which may not be suitable for handling noise and outliers in the data. As a result, researchers have proposed alternative approaches, such as general loss-based NMF [35] and self-paced NMF [36]. These methods utilize different optimization criteria that can be more robust to noisy data and outliers.

B. Archetypal Analysis (AA)

AA was first introduced in [27] and consists in modeling individual pixels present in the HSI as a linear mixture of archetypes (here the endmembers). Interestingly, archetypes are defined as convex combinations of the individual pixels present in the HSI. Motivated by the good interpretability of AA, Zhao et al. [37] have proposed a kernelized variant, which enables greater modeling flexibility at the cost of an extra hyper-parameter (the bandwidth of the RBF kernel functions). Notably, they use the relaxation form introduced in [38] to handle the cases where endmembers are located outside of the convex hull of the data.

Inspired by the robust AA formulation introduced in [39] that considers the Huber loss instead of the squared loss to reduce the impact of noise and outliers, Sun et al. [40] have introduced a robust kernel archetypal analysis (RKADA) method for blind hyperspectral unmixing. Their approach refines the standard AA formulation by adding a binary sparse constraint on the pixels contributions. Consequently, their method ensures that each endmember actually corresponds to real pixels rather than a sparse linear mixture of all pixels. The resulting optimization algorithm is a block-coordinate descent scheme that uses an active-set method [41] to solve the smooth least squares optimization subproblems with a simplicial constraint. As noted by [39], the active-set algorithm can be seen as an aggressive strategy that can leverage the underlying sparsity of the subproblems solutions.

Recently, Xu et al. [42] have proposed an ℓ_1 sparsity-constrained AA algorithm to increase the sparsity of the abundances. Rather than using an active-set algorithm, they adopt a projected gradient method [43] to solve the resulting optimization subproblems inside an alternating scheme. Their optimization approach is similar to the Near-Convex

Archetypal Analysis (NCAA) [28] method which aims at combining the advantages of both AA and NMF, by requiring the endmembers to be *linear*, instead of convex, combinations of the pixels.

C. Autoencoders (AE)

Interest in deep learning has been rapidly growing in many fields including remote sensing [44] thanks to the increasing amount of available data, rising computational power, and the development of suitable algorithms. Autoencoders for hyperspectral unmixing have been widely used since the pioneering work of Licciardi et al. [45]. In a nutshell, the encoder transforms an input into abundances maps that are then linearly decoded by the decoder. The encoder activations correspond to the fractional abundances while the parameters of the linear decoder layer correspond to the endmembers spectra. While the AE described above corresponds to a basic fully connected architecture, refinements have been introduced such as using a loss functions that involves the spectral angle distance and a dedicated sparsity penalty term as in [46], or convolutions to take advantage of the spatial structure in HSI [47]. See [48] for a comprehensive study on the various existing autoencoder architectures for blind hyperspectral unmixing. In addition to simple autoencoders, researchers have proposed more complicated architectures such as (i) a siamese network architecture, known as EGU-Net [49], which aims to enhance endmember extraction using nearly pure pixels, (ii) cascaded autoencoders as in [50] whose cycle-consistency property leads to improved unmixing performances. Finally, it is worth mentioning the approach by Rasti et al. [51] to incorporate a geometrically motivated penalty into deep learning-based approaches. Their method, called minimum simplex convolutional network (MiSiCNet), uses both geometrical and spatial information, by means of a convolutional encoder-decoder architecture, to tackle unsupervised hyperspectral unmixing.

Although the surge of deep learning methods has led to improving overall unmixing performances, training a single network per image remains costly due to the required extensive hyperparameters search. Moreover, training such networks requires GPU, and yet it can be considerably slower than traditional methods that run on CPU.

III. Methodology

In this section, we present our model formulation before describing its optimization. Next, we mention implementation details required to run our approach. Finally, we explain how to leverage our efficient GPU implementation and propose a procedure to make our model robust to hyper-parameter choices and thus easy to use in practice.

A. Model Formulation

Let $\mathbf{X}$ in $\mathbb{R}^{L\times N}$ be a hyperspectral image (HSI) where L is the number of channels and N is the number of pixels. We assume a linear mixing model such that

$$\mathbf{X} = \mathbf{EA} + \mathbf{N}, \tag{1}$$

where $\mathbf{E}$ in $\mathbb{R}^{L\times p}$ is the *mixing matrix* composed of the discretized spectra of p endmembers over L channels, and $\mathbf{A}$ in $\mathbb{R}^{p\times N}$ is the *abundance matrix* that describes, for each pixel, the fraction relative to each endmember. Finally, $\mathbf{N}$ represents some classical noise occuring in hyperspectral imaging.

We are interested in tackling the *blind* unmixing setting where both the mixing and abundance matrices are unknown. However, we assume that the number p of endmembers is known. Since $\mathbf{E}$ represents the endmembers reflectance over L channels, it follows that its elements should be non-negative, *i.e.* , $\mathbf{E} \geq 0$. This is also the case for $\mathbf{A}$, whose columns represent the abundances for each pixel, which in addition, should sum to one. In other words, each column of $\mathbf{A} = [\mathbf{a}_1, \ldots, \mathbf{a}_N]$ belongs to the simplex Δ_p defined as:

$$\Delta_p \triangleq \left\{ \mathbf{a} \in \mathbb{R}^p \text{ s.t. } \mathbf{a} \geq 0 \text{ and } \sum_{j=1}^{p} \mathbf{a}_j = 1 \right\}. \tag{2}$$

The previous model and constraint yields the classical optimization problem

$$\begin{aligned} &\arg\min_{\mathbf{E},\ \mathbf{A}} \ \frac{1}{2}\|\mathbf{X} - \mathbf{EA}\|_F^2. \\ &\text{s.t } \mathbf{E} \geq 0, \\ &\quad \mathbf{a}_i \in \Delta_p \quad \text{for } 1 \leq i \leq N, \end{aligned} \tag{3}$$

which is a variant of non-negative matrix factorization.

The archetypal analysis formulation we consider adds a constraint and forces the endmembers to be convex combinations of the pixels present in $\mathbf{X}$. Formally, it simply means that there exists a matrix $\mathbf{B}$ in $\mathbb{R}^{N\times p}$ such that $\mathbf{E} = \mathbf{XB}$ and the columns of $\mathbf{B}$ are in the simplex Δ_N (their entries are non-negative and sum to one). This formulation yields the archetypal analysis formulation [27]:

$$\begin{aligned} &\arg\min_{\mathbf{A},\ \mathbf{B}} \ \frac{1}{2}\|\mathbf{X} - \mathbf{XBA}\|_F^2, \\ &\text{s.t } \mathbf{a}_i \in \Delta_p \quad \text{for } 1 \leq i \leq N, \\ &\quad \mathbf{b}_j \in \Delta_N \quad \text{for } 1 \leq j \leq p \end{aligned} \tag{4}$$

where $\mathbf{A} = [\mathbf{a}_1, \ldots, \mathbf{a}_N]$ and $\mathbf{B} = [\mathbf{b}_1, \ldots, \mathbf{b}_p]$.

B. Optimization

Minimizing (4) is difficult since the objective function is not jointly convex in $(\mathbf{A}, \mathbf{B})$. However, it is convex with respect to (w.r.t.) one of the variables when the other one is fixed [38]. Consequently, it is natural to consider an alternating minimization scheme between $\mathbf{A}$ and $\mathbf{B}$, which is guaranteed to asymptotically provide a stationary point of the problem [52]. Yet, because of the non-convexity of the objective, the choice of optimization algorithm is important as different stationary points may not have the same quality in terms of statistical estimation. In other words, different optimization procedures may have a different "implicit bias", a phenomenon that is currently the focus of a lot of attention in machine learning [53], especially for deep learning models, suggesting that it may also be key for blind HU.

In this paper, we adopt an optimization scheme called entropic gradient descent which has shown better theoretical

properties in terms of convergence rates than gradient descent when optimizing over the simplex [54]. Our second motivation was the simplicity of the algorithm, which does not require performing orthogonal projections on the simplex, nor dealing with complicated active-set rules as in [39]. This enables us to take advantage of modern GPUs. We now present mathematical details before discussing implementation.

1) Entropic Descent Algorithm (EDA): As noted in [54], the entropic descent algorithm is simply a gradient descent method with a particular choice of a Bregman-like distance [55] generated by a specific function, here the negative entropy. As explained in [56], the choice of an appropriate distance-like function tailored to the geometry of the constraints, here the simplex, provide theoretical benefits in terms of convergence rates. The minimization over the simplex is precisely the reason why we adopt the negative entropy to derive the updates of the alternating minimization scheme. Formally, the negative entropy function h is defined as follows: for x in $\mathbb{R}^d$,

$$h(x) = \sum_{i=1}^{d} x_i \ln(x_i) \quad \text{if } x \in \Delta_d, \quad +\infty \text{ otherwise}, \tag{5}$$

with the convention that $0 \ln 0 \equiv 0$.

h exhibits several desirable properties, including convexity on Δ_d. This enables us to consider D_h, the Bregman divergence [55] w.r.t. h, defined, for x and y in $\mathbb{R}^d$

$$D_h(x, y) = h(x) - h(y) - \nabla h(y)^\top (x - y), \tag{6}$$

which is also called the Kullback-Leibler divergence. By convexity of h, we naturally have $D_h(x, y) \geq 0$.

We start by considering the following generic optimization problem:

$$\min_{\mathbf{z} \in \Delta_d} f(\mathbf{z}), \tag{7}$$

where f is a convex Lipschitz continuous function with a gradient at $\mathbf{z} \in \Delta_d$ denoted by $\nabla f(\mathbf{z})$ and Δ_d corresponds to the d-dimensional simplex (2).

The algorithm to solve (7) we consider performs the following iterates, given $\mathbf{z}^k \in \Delta_d$,

$$\mathbf{z}^{k+1} \leftarrow \arg\min_{\mathbf{z} \in \Delta_d} \left\{ \nabla f(\mathbf{z}^k)^\top (\mathbf{z} - \mathbf{z}^k) + \frac{1}{\eta_k} D_h(\mathbf{z}, \mathbf{z}^k) \right\}. \tag{8}$$

If D_h was simply a squared Euclidean norm, we would recover a projected gradient descent algorithm. By using instead the Bregman distance function D_h derived from the negative entropy, we obtain the entropic descent method. Here D_h measures the distance between two vectors in Δ_d. As such, the next iterate $\mathbf{z}^{k+1}$ should aim for the optimal balance between taking a gradient step and moving the least from the current iterate $\mathbf{z}^k$ according to the geometry induced by h, with η_k controlling this trade-off.

We will now see how the negative entropy h yields explicit steps that effectively enforce the simplicial constraints. Indeed, after a short classical calculation (see, for instance [54]), it is possible to show that the update (8) is equivalent to the following one: for all j in $\{1, \ldots, d\}$,

$$\mathbf{z}_j^{k+1} = \frac{\mathbf{z}_j^k e^{-\eta_k \nabla f(\mathbf{z}^k)_j}}{\sum_{l=1}^{d} \mathbf{z}_l^k e^{-\eta_k \nabla f(\mathbf{z}^k)_l}}, \tag{9}$$

where $\mathbf{z}_j^k$ is the j-th entry of the vector $\mathbf{z}^k$ and similarly, $\nabla f(\mathbf{z}^k)_j$ is the j-th entry of $\nabla f(\mathbf{z}^k)$.

We start by writing the Lagrangian corresponding to the constraint $\sum_{j=1}^{d} \mathbf{z}_j = 1$ using (6):

$$\mathcal{L}(\mathbf{z}, \nu) = \nabla f(\mathbf{z}^k)^\top \mathbf{z} + \frac{1}{\eta} \sum_{j=1}^{d} \mathbf{z}_j \ln(\mathbf{z}_j) - \frac{1}{\eta} \sum_{j=1}^{d} \mathbf{z}_j \ln(\mathbf{z}_j^k) + \nu \left(\sum_{j=1}^{d} \mathbf{z}_j - 1 \right). \tag{10}$$

Next, for $j \in \{1, \ldots, d\}$, we consider the derivative of $\mathcal{L}$ w.r.t. the j-th component of $\mathbf{z} \in \Delta_d$:

$$\nabla_{\mathbf{z}_j} \mathcal{L}(\mathbf{z}, \nu) = \nabla f(\mathbf{z}^k)_j + \frac{1}{\eta} \ln(\mathbf{z}_j) - \frac{1}{\eta} \ln(\mathbf{z}_j^k) + \nu + \frac{1}{\eta}, \tag{11}$$

We now consider the equation $\nabla_{\mathbf{z}_j} \mathcal{L}(\mathbf{z}, \nu) = 0$. Then, we have:

$$\ln(\mathbf{z}_j) = \ln(\mathbf{z}_j^k) - \eta \nabla f(\mathbf{z}^k)_j - \eta \nu - 1, \tag{12}$$

and

$$\mathbf{z}_j^{k+1} = \mathbf{z}_j^k e^{-\eta \nabla f(\mathbf{z}^k)_j} e^{-\eta \nu - 1}. \tag{13}$$

As a result, $\mathbf{z}_j^{k+1} \geq 0$ and it remains to choose ν such that $\sum_{j=1}^{d} \mathbf{z}_j^{k+1} = 1$ as in (9).

It is thus easy to see that the iterates $(\mathbf{z}^k)_{k \in \mathbb{N}}$ stay in the simplex Δ_d, and it is possible to show (see [54]) that the sequence $(\mathbf{z}^k)_{k \in \mathbb{N}}$ converges to the set of solutions of (7) with the appropriate step-sizes η_k. Interestingly, the update (9) can be implemented efficiently by using the softmax function, assuming the entries of $\mathbf{z}^k$ are positive:

$$\mathbf{z}^{k+1} = \text{softmax}\left(\log(\mathbf{z}^k) - \eta_k \nabla f(\mathbf{z}^k) \right). \tag{14}$$

where $\log(\mathbf{z}^k)$ is the vector carrying the logarithm of each entry of $\mathbf{z}^k$. This update immediately suggests a high compatibility with GPUs.

2) Alternating Optimization Scheme: We are now in shape to describe an alternating optimization scheme, consisting of performing, alternatively, K_1 updates of EDA for minimizing $\mathbf{A}$ when $\mathbf{B}$ is fixed, and vice versa using K_2 updates. This strategy is presented in Algorithm 1. Formally, by replacing the generic function f with the functions corresponding to the two optimization subproblems, we obtain the following updates:

$$\mathbf{A}^{k+1} = \text{softmax}\left(\log(\mathbf{A}^k) + \eta_{1_k} \mathbf{B}^\top \mathbf{X}^\top (\mathbf{X} - \mathbf{X}\mathbf{B}\mathbf{A}^k) \right), \tag{15}$$

$$\mathbf{B}^{k+1} = \text{softmax}\left(\log(\mathbf{B}^k) + \eta_{2_k} \mathbf{X}^\top (\mathbf{X} - \mathbf{X}\mathbf{B}^k\mathbf{A}) \mathbf{A}^\top \right), \tag{16}$$

where $\log(\mathbf{A}^k)$ is the matrix carrying the logarithm of each entry of $\mathbf{A}^k$ while softmax is applied in parallel on the columns of $\log(\mathbf{A}^k) + \eta_{1_k} \mathbf{B}^\top \mathbf{X}^\top (\mathbf{X} - \mathbf{X}\mathbf{B}\mathbf{A}^k)$. Note that when $\mathbf{A}$ and $\mathbf{B}$ are initialized with positive values, these iterates keep them positive. In addition, our optimization strategy does not require inverting any matrix contrarily to ADMM-based approaches [57].

Algorithm 1 Entropic Descent Archetypal Analysis (EDAA)

Input: ℓ_2-normalized data $\mathbf{X}$ in $\mathbb{R}^{L\times N}$; p (number of endmembers); T (number of outer iterations); K_1 (number of inner iterations for $\hat{\mathbf{A}}$); K_2 (number of inner iterations for $\hat{\mathbf{B}}$).
Initialize $\hat{\mathbf{A}} \in \mathbb{R}^{p\times N}$ using (18).
Initialize $\hat{\mathbf{B}} \in \mathbb{R}^{N\times p}$ using (19).
Set η_1 according to (20).
Set η_2 according to (21).
for $t = 1, \ldots, T$ **do**
  for $k = 1, \ldots, K_1$ **do**
    $\hat{\mathbf{A}} \leftarrow \text{softmax}\left(\log(\hat{\mathbf{A}}) + \eta_1 \hat{\mathbf{B}}^\top \mathbf{X}^\top (\mathbf{X} - \mathbf{X}\hat{\mathbf{B}}\hat{\mathbf{A}})\right)$
      ▷ log is applied element-wise;
      ▷ softmax is applied along the first dimension.
  end for
  for $k = 1, \ldots, K_2$ **do**
    $\hat{\mathbf{B}} \leftarrow \text{softmax}\left(\log(\hat{\mathbf{B}}) + \eta_2 \mathbf{X}^\top (\mathbf{X} - \mathbf{X}\hat{\mathbf{B}}\hat{\mathbf{A}})\hat{\mathbf{A}}^\top\right)$
  end for
end for
$\hat{\mathbf{E}} \leftarrow \mathbf{X}\hat{\mathbf{B}}$
Return $\hat{\mathbf{E}}$, $\hat{\mathbf{A}}$ ▷ Estimated endmembers, abundances.

C. Implementation Details

1) Normalization: The input image $\mathbf{X} = [\mathbf{x}_1, \ldots, \mathbf{x}_N]$ is ℓ_2-normalized for each pixel: for all i in $\{1, \ldots, N\}$,

$$\mathbf{x}_i \leftarrow \frac{\mathbf{x}_i}{||\mathbf{x}_i||_2}, \tag{17}$$

where $\mathbf{x}_i$ denotes the i-th pixel. This step is important to gain invariance to illumination changes.

2) Initialization: We initialize the abundance matrix $\mathbf{A}$ uniformly,

$$\mathbf{A}^0 = \frac{1}{p}\mathbb{1}_p \mathbb{1}_N^T, \tag{18}$$

where $\mathbb{1}_d$ denotes a d-dimensional vector of ones. This corresponds to the maximal entropy configuration for each pixel. The entropy for each pixel will naturally decrease as a result of the optimization, but the high entropy of the initialization will have a regularization effect.

The initialization of the pixel contribution matrix $\mathbf{B}$ is then also close to uniform. Nevertheless, we introduce a small random perturbation which is necessary to break the symmetry between the columns of $\mathbf{B}$ (otherwise, the updates of $\mathbf{A}$ and $\mathbf{B}$ will keep them invariant). Concretely, the entries of $\mathbf{B}$ are randomly sampled according to the uniform distribution on $[0, 1]$, $\mathcal{U}_{[0,1]}$. Next, they are rescaled by a factor 0.1. Finally, we apply the softmax function on each column so that the columns of $\mathbf{B}$ belong to the simplex Δ_N, for j in $\{1, \ldots, p\}$,

$$\mathbf{b}_j^0 = \text{softmax}(0.1\ \mathbf{u}), \tag{19}$$

where $\mathbf{u} \sim \mathcal{U}_{[0,1]^N}$. In practice, we observe that such an initialization leads to a matrix $\mathbf{B}^0$ that is very close to a uniform initialization $\frac{1}{N}\mathbb{1}_N \mathbb{1}_p^T$.

3) Step Sizes: We use constant step sizes η_1 and η_2, for $\mathbf{A}$ and $\mathbf{B}$ respectively.

$$\eta_1 = \frac{\gamma}{\sigma_{\max}^2}, \tag{20}$$

where γ is a value in $S = \{0.125, 0.25, 0.5, 1, 2, 4, 8\}$ and $\sigma_{\max}$ is the largest singular value of the matrix $\mathbf{X}\mathbf{B}^0$. We recover the classical convergence of gradient descent with fixed step size [58] up to the factor γ, since $\sigma_{\max}^2$ corresponds to the Lipschitz constant of the sub-problem related to (4) when minimizing w.r.t $\mathbf{A}$, $\mathbf{B}$ being fixed. Having γ in S allows us to use slightly different step sizes and yields better performance in practice. Note that our model selection procedure, presented later, will automatically choose the right parameter γ, thus removing the burden for the user of having to deal with an extra hyper-parameter. Finally, η_2 is simply a rescaled version of η_1 to account for the matrices being of transposed dimensions:

$$\eta_2 = \sqrt{\frac{p}{N}}\ \eta_1. \tag{21}$$

4) Hyperparameters: For all experiments, if not stated otherwise, we set $T = 100$ and $K_1 = K_2 = 5$ as it provides a good trade-off between convergence speed and unmixing accuracy. Note also that these hyper-parameters are robust to different real datasets as detailed in section IV.

D. Model Selection Procedure

As stated above, the initialization of the matrix $\mathbf{B}$ is random, leading to different solutions for each run of the algorithm since the overall optimization problem is non convex. Besides, we allow for different step-sizes γ, which we draw randomly from the set S in practice, Since the convergence of the algorithm is very fast (see experimental section for concrete benchmarks), we are able to provide a large diversity of solutions given a dataset by running M times our method with different random seeds, while keeping the global computational complexity reasonable. A major question we address next is then *how to select optimally the best solution in terms of unmixing accuracy*.

For this, we take inspiration from classical model selection and sparse estimation theory [59]. First, we measure the fit of each solution in terms of residual error $\|\mathbf{X}-\mathbf{X}\mathbf{A}\mathbf{B}\|_1$, choosing the ℓ_1-norm which is known to be more robust to outliers than the mean squared error. Second, we *select* the set of solutions that are in the same ball park as the best solution we have obtained in terms of ℓ_1 fit. This selection process is illustrated by the red dotted line in Figure 1, while the precise criterion is described in Algorithm 2.

From the subset of solutions with good fit, we then choose the one whose endmembers have the best incoherence, a desired property, which is classical in the theory of sparse estimation [60], [61], [62]. Indeed, dictionaries (here endmembers) with more incoherence will benefit from better theoretical guarantees in terms of estimation of abundances, making it a natural criterion for model selection in the context of unmixing. Formally, the coherence is simply defined as the maximal pairwise spectral correlation between the estimated

Algorithm 2 Model Selection Procedure

Input: M (number of runs); ℓ_2-normalized data $\mathbf{X}$ in $\mathbb{R}^{L\times N}$; p (number of endmembers); T (number of outer iterations); K_1 (number of inner iterations for $\hat{\mathbf{A}}$); K_2 (number of inner iterations for $\hat{\mathbf{B}}$).
for $m = 1, \ldots, M$ **do**
  Set random seed s_m.
  $\hat{\mathbf{E}}_m, \hat{\mathbf{A}}_m \leftarrow \text{EDAA}(\mathbf{X}, p, T, K_1, K_2, s_m)$ ▷ See (1)
  $\text{fit}_m \leftarrow ||\mathbf{X} - \hat{\mathbf{E}}_m\hat{\mathbf{A}}_m||_1$
  Compute coherence μ_m on $\hat{\mathbf{E}}_m$. ▷ See (22)
end for
$\text{fit}_{\min} \leftarrow \min\{\text{fit}_1, \ldots, \text{fit}_M\}$
$\mathcal{I} \leftarrow \{m \mid \text{fit}_m \leq 1.05 \times \text{fit}_{\min}\}$ ▷ Subset of models.
$\text{best} \leftarrow \arg\min\{\mu_i, i \in \mathcal{I}\}$
Return: $\hat{\mathbf{E}}_{\text{best}}$, $\hat{\mathbf{A}}_{\text{best}}$.

endmembers. More precisely, for $\hat{\mathbf{E}} = [\hat{\mathbf{e}}_1, \ldots, \hat{\mathbf{e}}_p]$ the endmembers matrix, the coherence μ is defined as:

$$\mu = \max_{k\neq k'} \langle \hat{\mathbf{e}}_k, \hat{\mathbf{e}}_{k'} \rangle, \tag{22}$$

where $\langle . \rangle$ denotes the inner product.

To the best of our knowledge, this is the first time the coherence μ is used as a model selection criterion for archetypal analysis. Our experiments, see next section, show that it is highly effective.

In summary, we automatically select the model whose endmembers have the lowest maximal pairwise spectral correlation among the ones that have a good ℓ_1 fit. This strategy is illustrated in figure 1 and described in Algorithm 2. In the experiments, the number of runs M was set to 50.

IV. Experiments

We have performed experiments on one simulated dataset with different noise and purity levels as well as six standard real datasets whose descriptions are given below.

A. Hyperspectral Data Description

1) Simulated dataset: For our study, we chose six endmembers from the USGS library and generated a 1000-pixel data cube using the methodology outlined in [63]. This approach allowed us to vary the purity level of the pixels by adjusting the parameter ρ, which is used in Table I. Specifically, lower values of ρ correspond to pixels that are less pure, while higher values indicate greater purity. By manipulating this parameter, we were able to simulate a range of real-world scenarios and evaluate the robustness of our algorithm under different conditions.
2) Samson: The Samson[2] hyperspectral image is a 95 × 95 pixels sub-region of a larger image captured using 156 bands spanning from 401 to 889 nm. Three main materials have been identified: Tree, Soil and Water. Note that we use a different ground truth from [51] that we selected for its sharper details on the abundances.
3) Jasper Ridge: The Jasper Ridge[2] hyperspectral image is a 100 × 100 pixels sub-region of a larger image initially captured using 224 bands spanning from 380 to 2500 nm. In total, 198 bands remain as 26 were removed as a pre-processing step due to dense water vapor and atmospheric effects. Four main materials have been identified: Tree, Dirt, Water and Road.
4) Urban4 and Urban6: The Urban[2] hyperspectral image is a 307×307 pixels image collected by the Hyperspectral Digital Image Collection Experiment (HYDICE) [64] sensor using 210 bands spanning from 400 to 2500 nm. In total, 162 bands remain as 48 were removed as a pre-processing step due to dense water vapor and atmospheric effects. There exists three versions of this dataset w.r.t. the number of endmembers. In this study, we focus on the two extremes: Urban4 contains 4 endmembers (Asphalt Road, Grass, Tree and Roof) and Urban6 contains two additional materials: Dirt and Metal, making it more challenging.
5) APEX: The APEX [65] hyperspectral image that we consider in this paper is a 111 × 122 pixels cropped region[3] of a larger image captured over 285 bands spanning from 413 to 2420nm. Four main materials were identified: Road, Tree, Roof and Water. Note that the ground truth abundance map for water appears to contain shadows due to the sunlight direction (see figure 3). This is a common issue when dealing with real remote sensing data and we cannot expect the semi-automated ground truth abundances map generation to be perfect.
6) Washington DC Mall: The Washington DC Mall (WDC) hyperspectral dataset[3] consists in a 319 × 292 pixels image captured by the HYDICE [64] sensor over 191 bands spanning from 400 to 2400 nm. Six main materials were identified: Grass, Tree, Roof, Road, Water and Trail.

According to [66] (Samson, Jasper Ridge and Urban) and [51] (APEX and WDC), the endmembers spectra were manually selected from the images and the ground truth abundances were set by FCLS. Illustrations of the datasets and their ground truth endmembers are available in the supplementary material.

B. Experimental Setup

We compare our approach to nine competitive methods from different unmixing categories:

- Geometrical unmixing baseline: FCLS [20] using VCA [19] to extract endmembers. Our implementation of the FCLS algorithm uses the *DecompSimplex* routine implemented in SPAMS.[4] This method relies on the active-set algorithm [41] that enables significantly faster convergence than generic quadratic programming solvers by leveraging the underlying sparsity of the abundances as noted by [39].

[2]Downloaded at https://rslab.ut.ac.ir/data

[3]Downloaded at https://github.com/BehnoodRasti/MiSiCNet

[4]http://thoth.inrialpes.fr/people/mairal/spams/

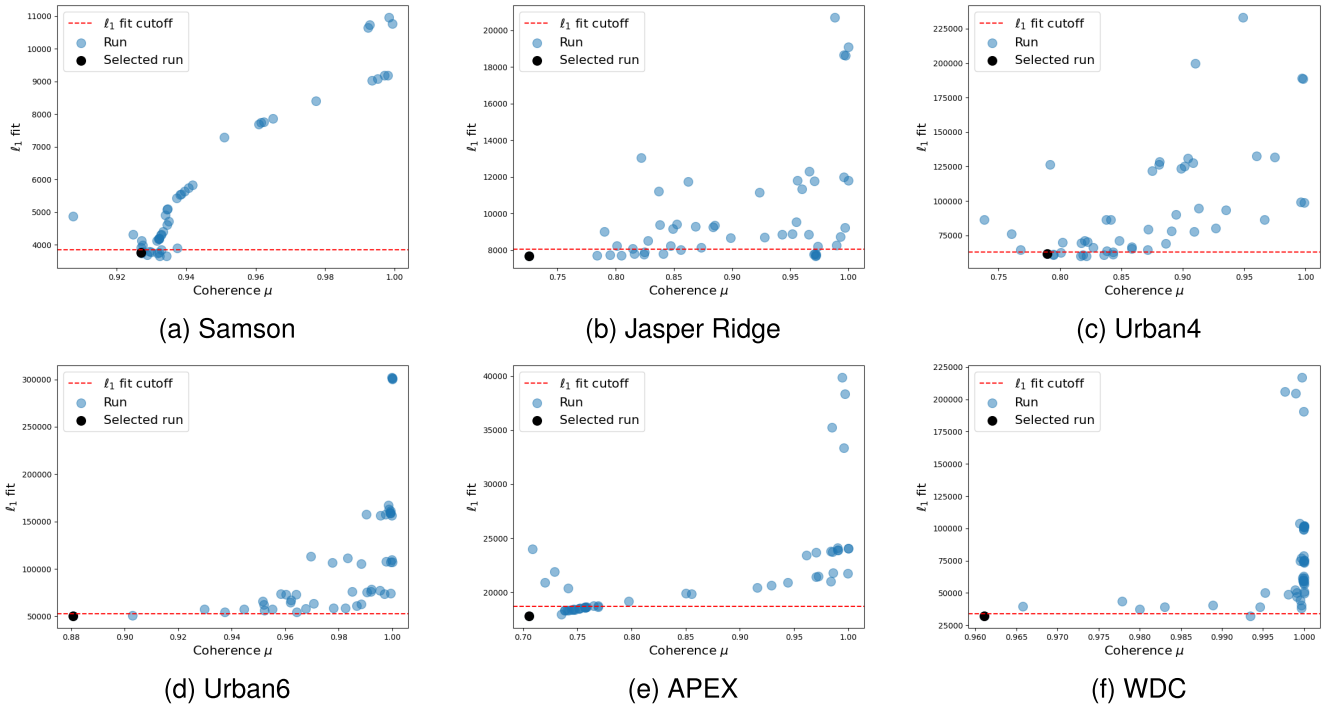

Fig. 1. Illustration of the model selection procedure on six datasets using $M = 50$ runs. Runs are illustrated by blue dots and the selected one is in black. The selected run is the one with lowest coherence μ under the dashed red line representing the ℓ_1 fit threshold, see Alg. 2.

TABLE I

ABUNDANCES RMSE AND ENDMEMBERS SAD FOR THE SIMULATED DATASET WITH DIFFERENT NOISE AND PURITY LEVELS. THE BEST RESULTS ARE SHOWN IN BOLD. THE SECOND BEST RESULTS ARE UNDERLINED

Methods	ρ	RMSE SNR (dB)			SAD SNR (dB)		
		20	30	40	20	30	40
FCLS [19], [20]	0.7	16.53	17.15	17.97	7.20	7.18	7.38
	0.85	12.96	11.44	10.15	3.86	3.05	2.45
	1.0	8.03	4.43	3.31	1.84	0.58	**0.17**
MiSiCNet [51] ($\lambda = 0.1$)	0.7	10.60	8.64	8.31	**3.94**	**1.88**	**1.58**
	0.85	13.76	8.62	7.68	4.70	1.20	**0.75**
	1.0	15.08	6.42	3.73	5.11	1.12	0.39
ADMMNet [67]	0.7	15.49	16.92	16.68	11.44	13.76	13.48
	0.85	17.18	19.54	19.58	8.52	10.23	9.93
	1.0	21.13	16.57	16.60	9.03	6.26	6.55
MVCNMF [30]	0.7	15.80	9.75	15.53	6.85	2.52	6.85
	0.85	9.88	9.80	9.54	**1.85**	**1.16**	0.76
	1.0	6.84	7.03	3.55	1.12	0.78	**0.17**
NMFQMV [22]	0.7	**10.11**	**8.32**	**7.36**	4.96	4.76	3.32
	0.85	10.58	**7.11**	**6.79**	2.91	1.18	0.98
	1.0	11.24	6.62	3.80	3.25	1.54	0.41
SeCoDe [25]	0.7	13.49	13.36	13.62	16.06	16.38	16.54
	0.85	13.56	14.76	14.69	14.62	15.79	15.73
	1.0	7.88	4.74	4.35	7.76	5.07	4.76
NCAA [28]	0.7	14.62	13.43	14.64	8.32	8.00	7.44
	0.85	12.15	7.66	7.19	5.72	2.56	1.10
	1.0	11.61	3.20	3.25	5.37	1.70	0.71
AA ($T = 100$)	0.7	17.67	17.23	17.18	7.61	6.71	6.43
	0.85	9.44	8.23	8.04	2.55	2.09	1.86
	1.0	**5.13**	**2.98**	**2.75**	**0.88**	**0.45**	0.34
EDAA ($K_1 = K_2 = 30$)	0.7	14.35	13.75	14.91	5.50	5.01	6.52
	0.85	**8.84**	8.37	8.37	2.63	2.18	1.92
	1.0	5.82	4.26	2.89	0.96	0.51	0.33

- Deep learning unmixing: Endnet[5] [46] using VCA [19] to initialize the endmembers, MiSiCNet[6] [51] and the deep unrolling network ADMMNet[7] [67].
- NMF-based blind unmixing: minimum-volume-constraint non-negative matrix factorization (MVCNMF)[8] [30], non-negative matrix factorization quadratic minimum

[5]Implementation at https://github.com/burknipalsson/hu_autoencoders
[6]Implementation at https://github.com/BehnoodRasti/MiSiCNet
[7]No implementation available online.
[8]Implementation at http://bit.ly/NCAAv1

TABLE II

ABUNDANCES RMSE ON SIX REAL DATASETS. THE BEST RESULTS ARE SHOWN IN BOLD. THE SECOND BEST RESULTS ARE UNDERLINED

		FCLS	Endnet	MiSiCNet	ADMMNet	MVCNMF	NMF-QMV	SeCoDe	NCAA	AA ($T = 100$)	EDAA
Samson	Soil	11.28	11.61	6.47	18.41	6.62	13.67	7.67	12.37	6.16	**5.74**
	Tree	9.13	7.72	5.38	14.32	**3.47**	8.40	6.68	4.87	4.00	3.77
	Water	5.05	6.86	3.47	6.67	4.66	11.61	2.57	7.80	**2.30**	2.59
	Overall	8.88	8.97	5.25	14.00	5.08	11.44	6.05	8.89	4.44	**4.24**
Jasper Ridge	Dirt	21.23	18.26	21.68	21.58	20.19	19.97	20.62	18.85	9.29	**7.32**
	Road	24.72	29.40	24.94	21.37	24.28	26.13	21.42	23.95	8.04	**7.61**
	Tree	11.20	4.00	3.41	13.19	9.80	14.55	9.57	**3.34**	7.43	6.63
	Water	13.61	22.38	7.07	13.92	17.83	19.81	9.38	10.36	6.10	**5.69**
	Overall	18.52	20.70	16.98	17.96	18.79	20.53	16.31	16.18	7.80	**6.85**
Urban4	Road	30.54	12.04	10.30	19.45	12.89	20.25	12.84	11.58	11.13	**8.62**
	Grass	32.99	17.94	12.35	22.17	14.00	20.22	24.34	**7.57**	11.47	9.28
	Roof	15.40	11.28	8.01	13.31	10.98	13.29	13.20	10.46	7.30	**6.37**
	Tree	20.02	11.69	8.78	16.45	13.63	21.56	26.36	7.63	8.51	**6.27**
	Overall	25.78	13.52	10.00	18.15	12.93	19.11	20.17	9.48	9.76	**7.75**
Urban6	Road	31.61	17.44	19.18	24.43	21.86	24.81	18.56	17.51	13.84	**11.39**
	Grass	23.62	31.47	18.84	32.86	28.19	27.97	20.97	**11.25**	11.88	18.61
	Roof	13.00	9.35	7.41	14.07	11.40	16.39	20.82	13.31	11.20	**6.01**
	Tree	16.14	15.50	11.72	22.71	16.12	19.95	26.67	10.52	**9.71**	9.85
	Dirt	25.06	30.67	23.95	22.33	20.42	24.31	20.18	26.71	**10.99**	15.94
	Metal	12.99	25.21	33.67	11.20	17.35	13.40	12.05	**8.93**	17.73	15.86
	Overall	21.54	23.09	16.27	22.41	19.92	21.74	20.34	15.88	**12.83**	13.63
APEX	Road	33.31	29.32	32.66	35.81	32.14	31.43	29.11	34.83	20.11	**16.54**
	Tree	20.97	18.18	19.97	27.25	23.96	24.62	22.25	21.98	14.58	**14.48**
	Roof	14.15	15.88	18.42	15.88	15.24	14.73	14.11	14.58	13.06	**11.27**
	Water	18.03	17.47	16.88	17.16	17.86	17.99	**16.77**	17.62	16.80	16.83
	Overall	22.77	20.90	22.86	25.35	23.23	23.10	21.35	23.55	16.36	**14.94**
WDC	Grass	30.90	**27.35**	31.61	39.87	37.79	34.69	30.55	34.78	35.88	32.53
	Tree	22.42	35.98	23.64	26.66	28.55	19.90	34.44	27.19	17.17	**11.46**
	Road	27.90	38.49	34.91	21.00	20.77	22.49	22.15	20.06	39.86	**13.97**
	Roof	8.71	27.04	11.98	9.95	**6.96**	19.81	10.17	18.21	20.52	29.31
	Water	17.76	12.94	14.72	20.44	12.93	20.34	21.23	24.83	24.03	**9.63**
	Trail	12.80	12.63	12.07	14.09	**11.26**	12.36	15.35	11.83	16.83	13.19
	Overall	21.57	27.64	23.39	24.00	22.42	22.60	23.81	24.89	27.24	**20.46**

volume (NMF-QMV)[9] [22] using the *boundary* term as the quadratic minimum volume penalty, near-convex archetypal analysis (NCAA)[10] [28] and AA [27] using the implementation from [39] developed in SPAMS. The approach denoted as AA involves solving (4) using an active-set method to optimize the convex sub-problems. This method is a conventional alternating approach that uses a fixed initialization ($\mathbf{A}$ and $\mathbf{B}$ entries are set to 0) and a fixed number of iterations ($T = 100$). While AA is a well-known method, it has not been thoroughly evaluated on various real-world unmixing datasets. Therefore, we include it as a competing method in our study to compare its performance with other state-of-the-art algorithms. By using AA, we can also assess the improvements in optimization achieved by EDAA since the underlying model (4) is the same for both methods. This comparison enables us to gain insights into the benefits of EDAA for hyperspectral unmixing.

- Finally, we include a recent technique addressing spectral variability: SeCoDe[11] [25].

To quantitatively evaluate the performance of the selected methods, we consider two metrics that are computed both globally and individually for each endmember. On one hand, we measure the quality of the generated abundances by means of the abundances root mean square error (RMSE) in percent between the ground truth and the estimated abundances:

$$\mathrm{RMSE}(\mathbf{A}, \hat{\mathbf{A}}) = 100 \times \sqrt{\frac{1}{pN} \sum_{i=1}^{p} \sum_{j=1}^{N} \left(\mathbf{A}_{i,j} - \hat{\mathbf{A}}_{i,j}\right)^2}. \quad (23)$$

On the other hand, we assess the quality of the estimated endmembers spectra by using the spectral angle distance (SAD) in degrees between the ground truth and the generated endmembers:

$$\mathrm{SAD}(\mathbf{E}, \hat{\mathbf{E}}) = \frac{180}{\pi} \times \frac{1}{p} \sum_{i=1}^{p} \arccos\left(\frac{\langle \mathbf{e}_i, \hat{\mathbf{e}}_i \rangle}{||\mathbf{e}_i||_2 ||\hat{\mathbf{e}}_i||_2}\right), \quad (24)$$

where $\mathbf{e}_i$ denotes the i-th column of $\mathbf{E}$, *i.e.* the spectrum of the i-th endmember.

C. Unmixing Experiments

Table I presents the results of our unmixing accuracy evaluation on the simulated dataset. Our analysis indicates that the performance of the AA variants (active-set based and EDAA) is lower in scenarios where the pixels are highly mixed ($\rho = 0.7$) when compared to geometrically motivated methods, such as MiSiCNet, MVCNMF and NMF-QMV, which do not rely directly on the pixel values to estimate the endmembers. However, it is worth noting that the EDAA method performs better

[9] Implementation at https://github.com/LinaZhuang/NMF-QMV_demo
[10] Implementation at http://bit.ly/NCAAv1
[11] Implementation at https://github.com/danfenghong/IEEE_TGRS_SeCoDe

TABLE III
ENDMEMBERS SAD ON SIX REAL DATASETS. THE BEST RESULTS ARE SHOWN IN BOLD. THE SECOND BEST RESULTS ARE UNDERLINED

		FCLS	Endnet	MiSiCNet	ADMMNet	MVCNMF	NMF-QMV	SeCoDe	NCAA	AA ($T = 100$)	EDAA
Samson	Soil	2.76	**0.61**	1.21	7.48	1.08	4.90	0.79	4.57	0.78	1.64
	Tree	3.05	1.93	3.38	2.48	2.54	5.34	1.96	2.42	**1.80**	1.98
	Water	7.15	1.48	5.36	6.68	4.82	11.14	2.56	5.15	1.38	**1.31**
	Overall	4.32	1.34	3.32	5.55	2.81	7.13	1.77	4.05	**1.32**	1.64
Jasper Ridge	Dirt	13.03	**1.63**	4.26	21.90	9.41	12.40	9.94	4.08	2.54	2.74
	Road	40.39	32.85	20.04	46.06	36.38	45.66	45.87	41.84	5.26	**3.10**
	Tree	11.16	1.39	**1.27**	10.78	8.28	14.46	12.42	2.25	4.68	4.23
	Water	13.24	3.21	4.18	13.14	10.40	14.53	6.44	5.00	3.03	**2.80**
	Overall	19.46	9.77	7.44	22.97	16.12	21.77	18.67	13.29	3.88	**3.22**
Urban4	Road	15.40	6.40	5.73	27.68	9.00	14.51	17.28	9.44	**3.70**	6.01
	Grass	24.18	3.09	5.84	18.58	7.32	16.39	7.25	4.81	**1.80**	2.14
	Roof	47.56	**3.76**	16.10	33.08	27.80	36.31	30.08	23.98	15.64	10.49
	Tree	19.82	2.32	4.60	11.34	6.39	22.48	16.17	**1.44**	3.47	2.81
	Overall	26.74	**3.89**	8.07	22.67	12.63	22.42	17.69	9.92	6.15	5.36
Urban6	Road	13.43	**3.26**	7.47	35.54	23.96	26.60	32.28	18.95	4.26	4.85
	Grass	22.30	4.13	10.97	29.20	12.60	21.63	13.34	7.36	3.44	**2.17**
	Roof	15.65	17.76	13.97	41.57	28.09	15.55	42.86	27.14	21.18	**13.70**
	Tree	20.70	7.72	9.99	16.67	12.93	23.31	25.85	**5.62**	8.76	8.92
	Dirt	69.81	17.42	19.57	36.46	19.59	23.60	27.66	37.21	**9.87**	13.33
	Metal	39.35	7.04	9.75	39.91	5.78	68.73	57.78	11.13	7.59	**4.52**
	Overall	30.21	9.56	11.95	29.20	17.16	29.90	33.30	17.90	9.18	**8.64**
APEX	Road	40.23	14.46	33.02	49.02	44.02	54.53	48.91	37.26	**4.80**	6.83
	Tree	14.13	7.53	**3.16**	17.57	7.49	16.06	16.37	6.50	7.56	7.68
	Roof	8.25	**4.36**	11.31	11.41	5.34	7.98	8.13	4.41	7.44	7.50
	Water	7.15	2.83	6.02	4.01	3.16	9.71	4.06	9.06	2.32	**2.21**
	Overall	17.44	7.30	13.38	20.50	15.00	22.07	19.37	14.31	**5.53**	6.06
WDC	Grass	17.40	**3.54**	17.46	43.51	11.01	34.36	25.43	15.33	17.10	8.11
	Tree	23.73	12.85	12.36	23.58	20.73	17.70	50.57	25.36	8.17	**1.81**
	Road	32.56	26.76	33.20	39.15	16.95	17.28	21.96	16.06	35.29	**7.67**
	Roof	34.84	13.70	28.87	31.47	**4.77**	44.87	46.46	33.94	16.19	50.97
	Water	4.78	1.75	1.57	11.60	5.43	19.74	30.22	4.64	3.62	**1.08**
	Trail	9.94	**1.49**	3.24	24.85	13.57	9.60	28.89	11.41	4.28	4.75
	Overall	20.54	**10.02**	16.11	29.03	12.08	23.93	33.92	17.79	14.11	12.40

TABLE IV
PROCESSING TIMES IN SECONDS ON SIX REAL DATASETS. THE BEST RESULTS ARE IN BOLD AND THE SECOND BEST UNDERLINED. EDAA INCLUDES THE MODEL SELECTION PROCEDURE OVER $M = 50$ RUNS

	FCLS	Endnet	MiSiCNet	ADMMNet	MVCNMF	NMF-QMV	SeCoDe	NCAA	AA ($T = 100$)	EDAA
Samson	**0.3**	≈ 560	≈ 80	55.2	3.8	20.1	24.2	113.1	33.0	11.2
JasperRidge	**0.4**	≈ 680	≈ 90	54.1	8.0	22.3	28.6	501.4	43.6	9.6
Urban4	**3.6**	≈ 720	≈ 411	504.6	25.4	112.5	475.1	3529.9	415.7	66.3
Urban6	**4.4**	≈ 1000	≈ 417	507.5	45.0	158.4	707.7	4880.8	593.7	75.2
APEX	**0.6**	≈ 720	≈ 92	80.0	11.3	27.2	56.1	441.5	75.4	15.4
WDC	**4.0**	≈ 1000	≈ 409	535.8	55.1	174.4	524.7	4179.0	493.1	81.0

than the plain AA method in this setting, thanks to its advanced model selection. In the case where pure pixels are present ($\rho = 1.0$), the AA model formulation (4) yields the best results, as it estimates the endmembers as convex combinations of the pixels. This approach is more robust to noise than the FCLS method. For the medium case ($\rho = 0.85$), both AA and EDAA exhibit better performance than MiSiCNet and NMF-QMV for low Signal-to-Noise Ratio (SNR) due to their greater robustness to noise. However, they are outperformed by MiSiCNet and NMF-QMV for high SNR, as there are no pure pixels available for these methods to leverage.

Tables II and III report the unmixing accuracy in terms of abundances RMSE (23) and endmembers SAD (24) on six standard real datasets. The datasets are arbitrarily ranked based on their difficulty. For a fair comparison, all methods were evaluated on the ℓ_2-normalized data (17) which induces slight changes compared to the results reported in [51].

Overall EDAA obtains the best abundances RMSE on five out of six real datasets. The results on Urban6 fall in favor of plain AA but the gap is moderate (12.83 vs 13.63) given that EDAA is more than seven times faster (594 versus 75 seconds) than plain AA on this dataset (see table IV). Note however that plain AA performs poorly on WDC where the ground truth endmembers are highly correlated. We argue that our model selection technique is instrumental in avoiding collapsing runs in which endmembers spectra are highly correlated. This is underlined by the overall competitive SAD results obtained by EDAA across datasets. It should be noted that the SAD metric alone is not sufficient to assess the unmixing performance as a good SAD score does not necessarily lead to better abundance

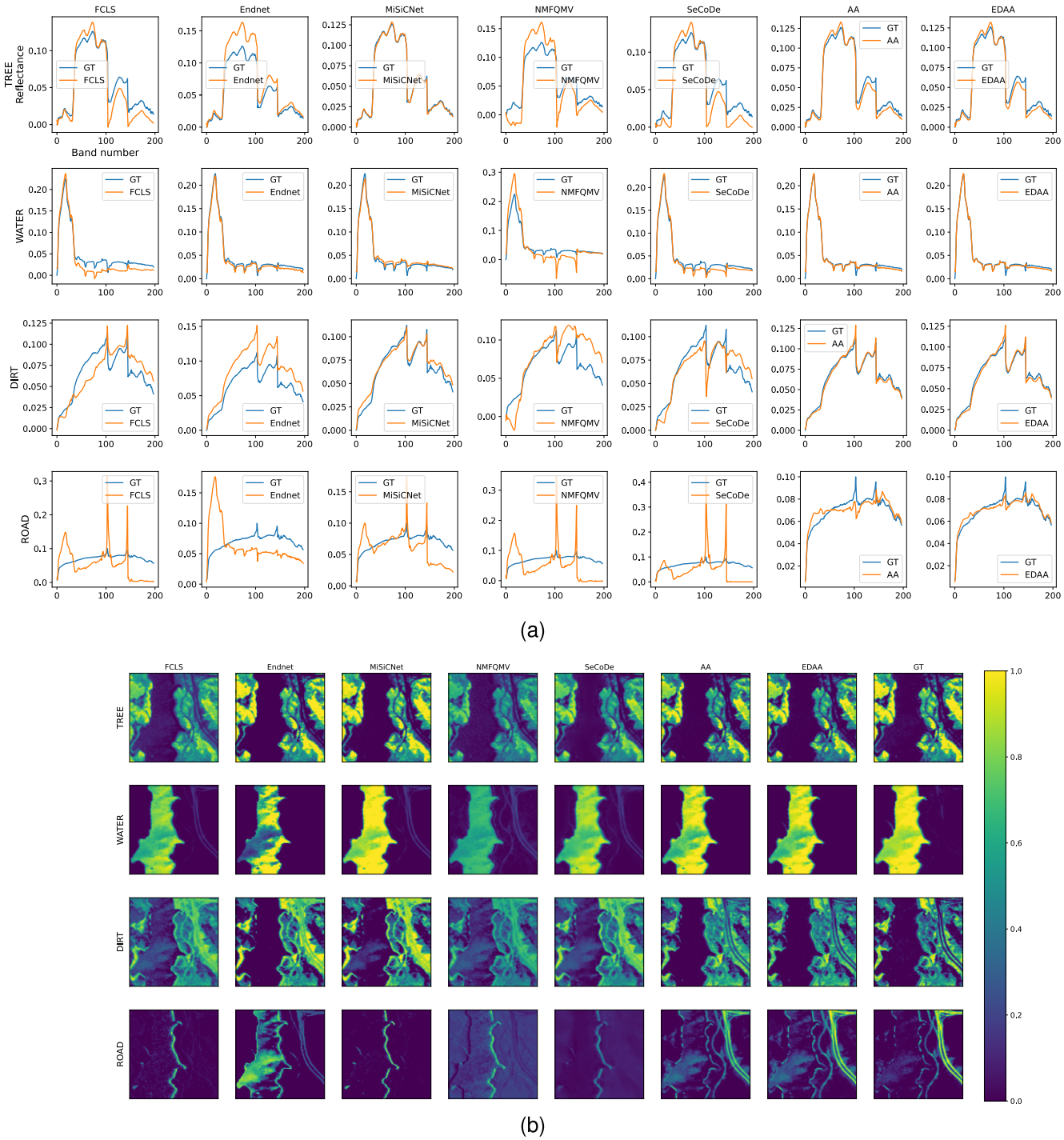

Fig. 2. Estimated endmembers (a) and abundances (b) on the Jasper Ridge dataset. Ground truth abundances are displayed on the right-most column.

maps. Thus it is not contradictory to have slightly worse SAD scores yet better looking abundance maps.

The FCLS baseline based on VCA obtains rather poor results except for WDC. This is likely due to the pure pixel assumption. Indeed, VCA selects a single pixel to represent the endmembers spectra, which is too stringent in real scenarios where spectral variability is ubiquitous.

Despite its quadratic minimum volume *boundary* term, NMF-QMV generally obtains worse results than the FCLS baseline. Since it operates the unmixing in a subspace, NMF-QMV cannot prevent the endmembers spectra from having negative values, which breaks the physical interpretability of the estimates and subsequently harms the unmixing performance. This phenomenon can be observed in figures 2, 3 and 4 for several endmembers. The associated abundances show that NMF-QMV produces maps that are too uniform and lack sparsity.

In contrast, MVCNMF is a strong baseline as it does not operate in a subspace.

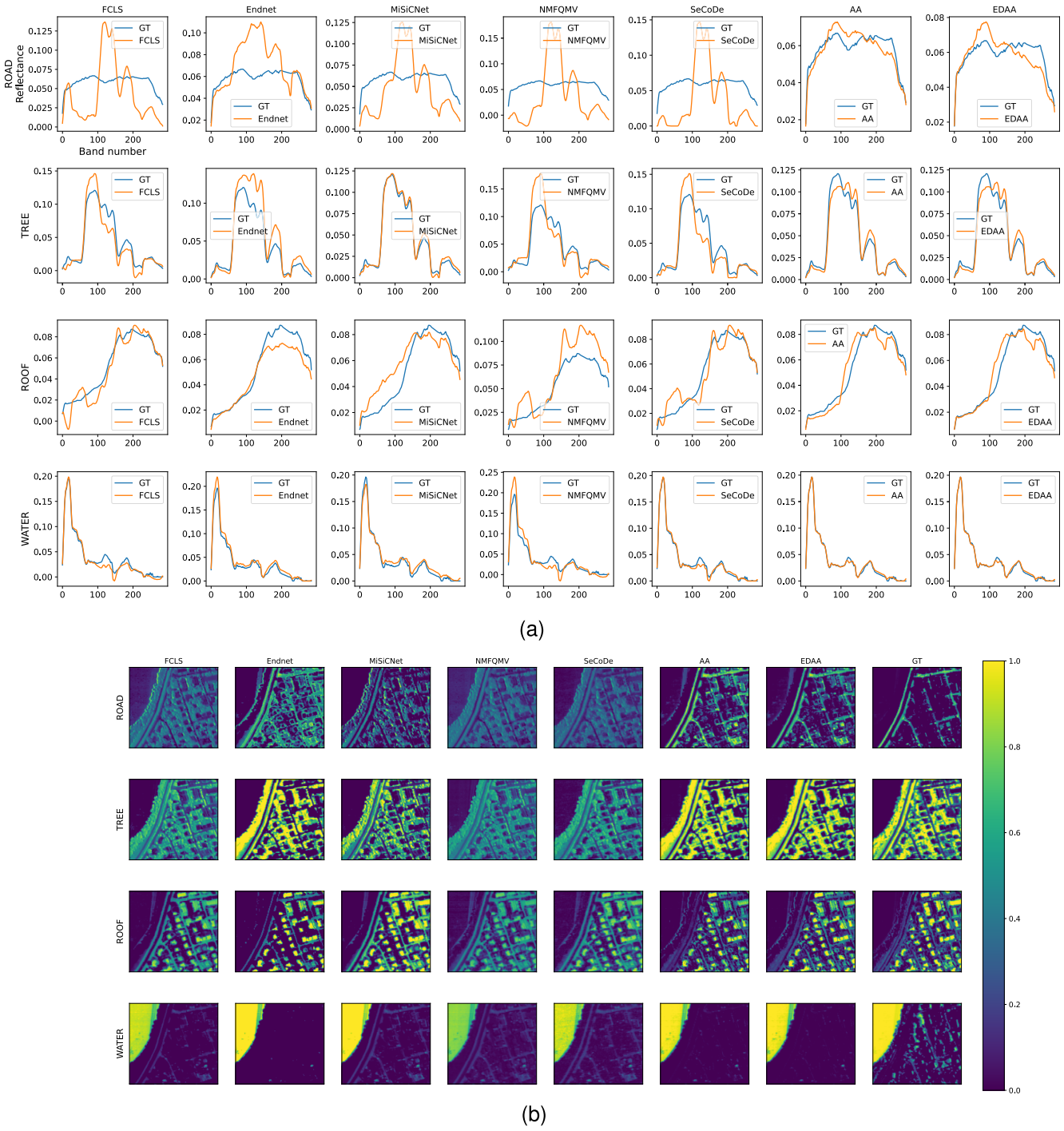

Fig. 3. Estimated endmembers (a) and abundances (b) on the APEX dataset. Ground truth abundances are displayed on the right-most column.

When it comes to deep learning based methods, Endnet achieves very good results in terms of SAD but tends to create overly sparse abundances which hinders its performance in terms of abundances RMSE. For instance, as can be seen in figure 2, the Road endmember is overlooked by Endnet even though EDAA recovers it neatly. Likewise, in figure 3, the Road endmember spreads too much compared to EDAA which appears closer to the ground truth.

MiSiCNet gives better unmixing results than Endnet in terms of abundances RMSE except for APEX although the SAD results falls in favor of Endnet except for Jasper Ridge. This is likely due to Endnet using the spectral angle distance on the input data in its loss which helps in achieving better SAD accuracy. However a good SAD is not sufficient to obtain good abundance maps, an area where MiSiCNet tends to shine as it incorporates spatial information by using convolutional filters and implicitly applying a regularizer on abundances.

ADMMNet obtains rather poor results on all datasets, which is likely due to using ADMM to blindly solve the linear mixing model rather than leveraging a known library.

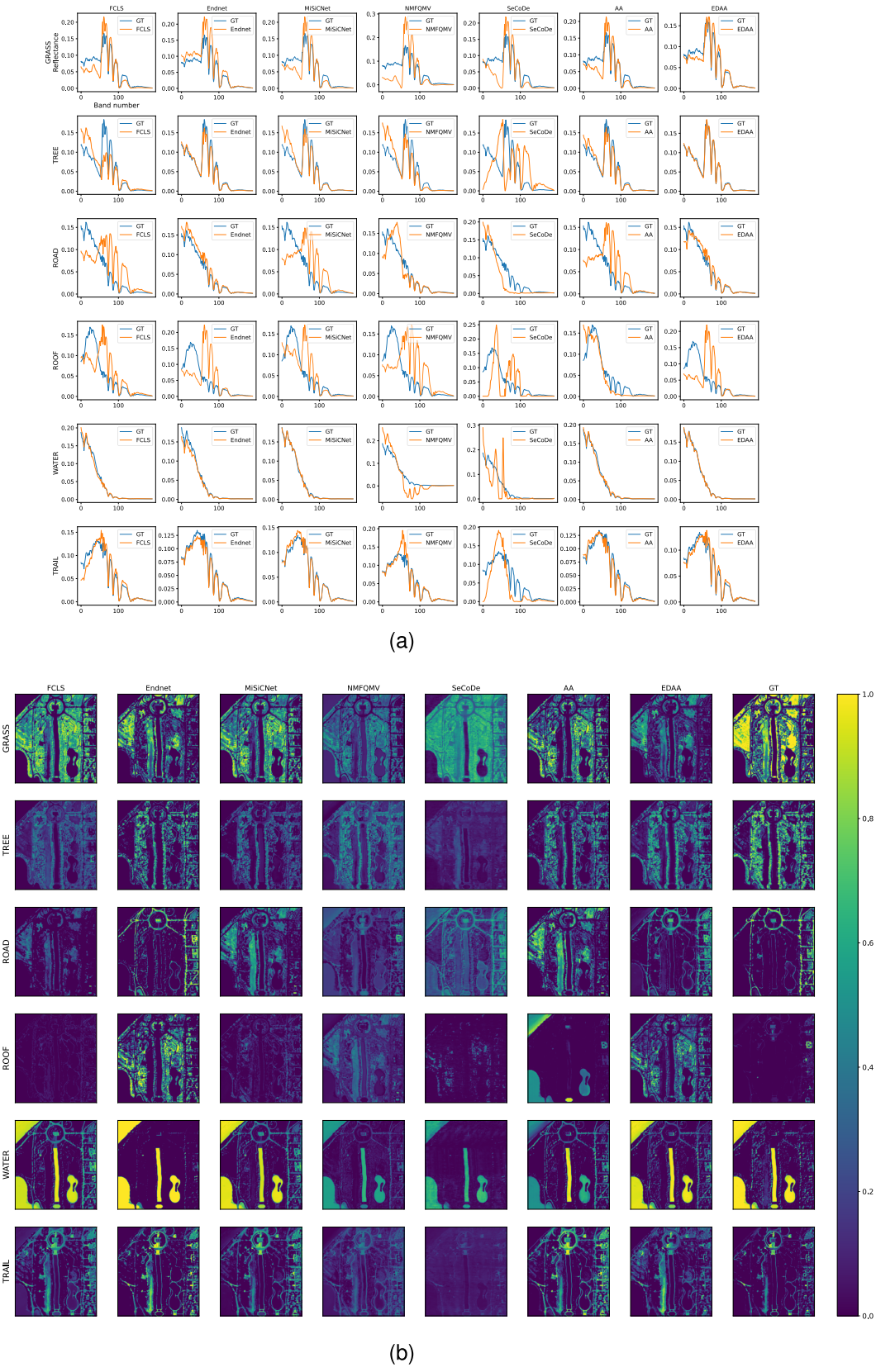

Fig. 4. Estimated endmembers (a) and abundances (b) on the WDC dataset. Ground truth abundances are displayed on the right-most column.

SeCoDe works reasonably well on the Samson dataset due to its implementation available online that has been tailored to this dataset. However it does not perform as well on other datasets, which is likely due to requiring precise hyperparameters tuning depending on the data at hand.

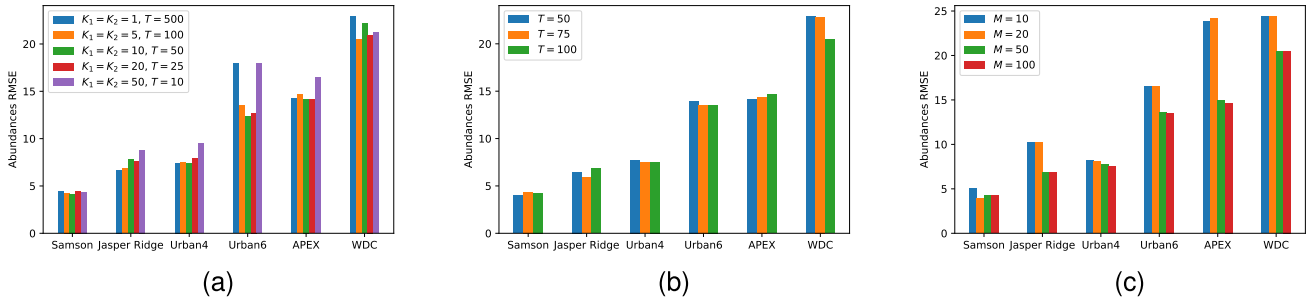

Fig. 5. Sensitivity analysis to the hyperparameters in Algorithms 1 and 2 measured in global abundances RMSE: (a) Varying inner and outer iterations K_1, K_2 and T for a constant number of updates (1000) and runs $M = 100$, (b) Varying outer iterations T using $K_1 = K_2 = 5$ and (c) Varying number of runs M using $K_1 = K_2 = 5$ and $T = 100$.

NCAA obtains reasonable results due to its fine-tuning procedure but its computational cost is prohibitive as we will be discussing next.

Finally, the plain AA method leveraging an active-set algorithm is a very competitive method for five out of six datasets, yet its performance drops significantly when dealing with the hardest mixing scenario that is WDC. For example, in figure 2 only AA and EDAA are able to uncover the Road endmember in Jasper Ridge whereas all the other techniques fail. Unlike EDAA, AA does not rely on a random initialization of the estimates as they are set to zeros in practice. Instead, EDAA requires starting from feasible points w.r.t. the optimization sub-problems. In addition, the initial estimates in EDAA should not contain any zeros due to the presence of the logarithm in (14). Thanks to the entropic gradient descent speed, it is possible to fit several models that have been randomly initialized as described in III-C and use the model selection procedure presented in Algorithm 2 which selects the model that exhibits the lowest coherence among the pool of candidates that are well fitted. Notably, this approach prevents the estimated endmembers from *collapsing* into degenerate solutions that would end up being perfectly correlated. This likely explains why the performance of plain AA on WDC drops as the endmembers are highly correlated.

Additional qualitative results for the Samson, Urban4 and Urban6 datasets can be found in the supplementary material. Moreover, we have also included the results on a 250×191 pixels subset of the Cuprite dataset, which is more challenging and does not come with ground truth abundance maps.

D. Computational Cost

Table IV reports the processing times for the different unmixing algorithms on the six real datasets. MVCNMF, NMF-QMV, SeCoDe and NCAA were implemented in Matlab (2020b) while FCLS, Endnet, MiSiCNet, ADMMNet and the AA variants were implemented in Python (3.8). NMF-QMV, MVCNMF, SeCoDe, NCAA, FCLS and AA run on CPU whereas Endnet, MiSiCNet, ADMMNet and EDAA run on GPU. The processing times were obtained using a computer with an Intel(R) Xeon(R) Silver 4110 processor (2.10 GHz), 32 cores, 64GB of memory, and a NVIDIA GeForce RTX (2080 Ti) graphical processing unit. The table shows that FCLS is clearly faster than the other unmixing techniques, however it is a supervised method that relies on an endmembers extraction algorithm. In this case, VCA is used which is also fast. The deep learning methods are the slowest techniques despite running on GPU. Interestingly, EDAA requires a lower computational cost than NMF-QMV and AA although our approach consists in aggregating 50 runs obtained iteratively. For example, it takes on average 1.5 seconds for EDAA to perform a single unmixing task on the Urban6 dataset, which is three times faster than FCLS. This demonstrates the efficiency of EDAA which allows us to use an adequate model selection procedure over several runs. Note that the processing time of NCAA is prohibitive due to its fine-tuning component despite using a fast projected gradient method.

E. Ablation Study

Finally, we study the sensitivity to hyper-parameters for Algorithm 1 and 2 in figure 5 where the Y-axis corresponds to the overall abundances RMSE. Given a fixed computational budget of 1000 updates, figure 5 (a) shows that the hyper-parameters of EDAA are robust provided that the number of runs M in the model selection is large enough (here 100). Only the two extremes configurations ($K_1 = K_2 = 1$, $T = 500$ and $K_1 = K_2 = 50$, $T = 10$) are slightly worse, especially on Urban6. For the remaining experiments, we use $K_1 = K_2 = 5$. In figure 5 (b), we see that the number of outer iterations is quite stable except for WDC which requires more updates (1000, *i.e.* $T = 100$). Finally, we study the importance of the number of runs M from which to select the best candidate in figure 5 (c). We observe that the model selection procedure requires at least 50 runs to obtain very good performances, hence we use $M = 50$ in our unmixing experiments. On unknown datasets where real-time unmixing is not required, it is advised to use a large number of runs (at least 100) to ensure that the model selection procedure yields a good candidate. Detailed results for both abundances RMSE and SAD metrics are available in the supplementary material.

V. Conclusion

We have proposed a new algorithm based on archetypal analysis for blind hyperspectral unmixing. We have shown how to take advantage of its efficient GPU implementation in order to develop an adequate model selection procedure to obtain state-of-the-art performances. Remarkably, our simple and easy-to-use approach considerably improves the unmixing results on a comprehensive collection of standard real

datasets. In addition, we have made our results reproducible by releasing an open source codebase which also includes the plain archetypal analysis variant presented in this study. While this paper was under review, we also investigated in [68] the problem of semi-supervised unmixing by using a variant of archetypal analysis, showing that such a framework may be useful beyond the problem of blind unmixing that we address here. Finally, it is worth noting that our approach does not consider the spatial structure of the data. This feature suggests that a natural extension to our approach would be to incorporate missing spatial information, which could potentially improve the accuracy of our results.

REFERENCES

[1] D. Landgrebe, "Hyperspectral image data analysis," *IEEE Signal Process. Mag.*, vol. 19, no. 1, pp. 17–28, Jan. 2002.

[2] A. Plaza et al., "Recent advances in techniques for hyperspectral image processing," *Remote Sens. Environ.*, vol. 113, no. 1, pp. 110–122, Sep. 2009.

[3] M. E. Schaepman, S. L. Ustin, A. J. Plaza, T. H. Painter, J. Verrelst, and S. Liang, "Earth system science related imaging spectroscopy—An assessment," *Remote Sens. Environ.*, vol. 113, pp. 123–137, Jan. 2009.

[4] A. F. H. Goetz, G. Vane, J. E. Solomon, and B. N. Rock, "Imaging spectrometry for Earth remote sensing," *Science*, vol. 228, no. 4704, pp. 1147–1153, Jun. 1985.

[5] R. O. Green et al., "Imaging spectroscopy and the airborne visible/infrared imaging spectrometer (AVIRIS)," *Remote Sens. Environ.*, vol. 65, no. 3, pp. 227–248, Sep. 1998.

[6] T. Adão et al., "Hyperspectral imaging: A review on UAV-based sensors, data processing and applications for agriculture and forestry," *Remote Sens.*, vol. 9, no. 11, p. 1110, Oct. 2017.

[7] A. C. Karaca, A. Ertürk, M. K. Güllü, M. Elmas, and S. Ertürk, "Automatic waste sorting using shortwave infrared hyperspectral imaging system," in *Proc. 5th Workshop Hyperspectral Image Signal Process., Evol. Remote Sens. (WHISPERS)*, Jun. 2013, pp. 1–4.

[8] A. Gowen, C. Odonnell, P. Cullen, G. Downey, and J. Frias, "Hyperspectral imaging—An emerging process analytical tool for food quality and safety control," *Trends Food Sci. Technol.*, vol. 18, no. 12, pp. 590–598, Dec. 2007.

[9] N. Fox, A. Parbhakar-Fox, J. Moltzen, S. Feig, K. Goemann, and J. Huntington, "Applications of hyperspectral mineralogy for geoenvironmental characterisation," *Minerals Eng.*, vol. 107, pp. 63–77, Jun. 2017.

[10] R. N. Clark et al., "Imaging spectroscopy: Earth and planetary remote sensing with the USGS tetracorder and expert systems," *J. Geophys. Res., Planets*, vol. 108, no. E12, pp. 1–15, Dec. 2003.

[11] 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, Jun. 2013.

[12] P. Ghamisi et al., "Advances in hyperspectral image and signal processing: A comprehensive overview of the state of the art," *IEEE Geosci. Remote Sens. Mag.*, vol. 5, no. 4, pp. 37–78, Dec. 2017.

[13] L. Parra, C. Spence, P. Sajda, A. Ziehe, and K.-R. Müller, "Unmixing hyperspectral data," in *Proc. Adv. Neural Inf. Process. Syst. (NIPS)*, vol. 12, 1999, pp. 1–12.

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

[15] J. M. Bioucas-Dias et al., "Hyperspectral unmixing overview: Geometrical, statistical, and sparse regression-based approaches," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 5, no. 2, pp. 354–379, Apr. 2012.

[16] P. Comon and C. Jutten, *Handbook of Blind Source Separation: Independent Component Analysis and Applications*. New York, NY, USA: Academic, 2010.

[17] J. W. Boardman, F. A. Kruse, and R. O. Green, "Mapping target signatures via partial unmixing of AVIRIS data," in *Proc. 5th Annu. JPL Airborne Earth Sci. Workshop*, 1995, pp. 1–11.

[18] M. E. Winter, "N-FINDR: An algorithm for fast autonomous spectral end-member determination in hyperspectral data," *Proc. SPIE*, vol. 3753, pp. 266–275, Oct. 1999.

[19] 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.

[20] D. C. Heinz, "Fully constrained least squares linear spectral mixture analysis method for material quantification in hyperspectral imagery," *IEEE Trans. Geosci. Remote Sens.*, vol. 39, no. 3, pp. 529–545, Mar. 2001.

[21] D. Lee and H. S. Seung, "Algorithms for non-negative matrix factorization," in *Adv. Neural Inf. Process. Syst. (NIPS)*, vol. 13, 2000.

[22] L. Zhuang, C.-H. Lin, M. A. T. Figueiredo, and J. M. Bioucas-Dias, "Regularization parameter selection in minimum volume hyperspectral unmixing," *IEEE Trans. Geosci. Remote Sens.*, vol. 57, no. 12, pp. 9858–9877, Dec. 2019.

[23] R. A. Borsoi et al., "Spectral variability in hyperspectral data unmixing: A comprehensive review," *IEEE Geosci. Remote Sens. Mag.*, vol. 9, no. 4, pp. 223–270, Dec. 2021.

[24] D. Hong, N. Yokoya, J. Chanussot, and X. X. Zhu, "An augmented linear mixing model to address spectral variability for hyperspectral unmixing," *IEEE Trans. Image Process.*, vol. 28, no. 4, pp. 1923–1938, Apr. 2019.

[25] J. Yao, D. Hong, L. Xu, D. Meng, J. Chanussot, and Z. Xu, "Sparsity-enhanced convolutional decomposition: A novel tensor-based paradigm for blind hyperspectral unmixing," *IEEE Trans. Geosci. Remote Sens.*, vol. 60, 2022, Art. no. 5505014.

[26] L. Ren, D. Hong, L. Gao, X. Sun, M. Huang, and J. Chanussot, "Orthogonal subspace unmixing to address spectral variability for hyperspectral image," *IEEE Trans. Geosci. Remote Sens.*, vol. 61, 2023, Art. no. 5501713.

[27] A. Cutler and L. Breiman, "Archetypal analysis," *Technometrics*, vol. 36, no. 4, pp. 338–347, 1994.

[28] P. De Handschutter, N. Gillis, A. Vandaele, and X. Siebert, "Near-convex archetypal analysis," *IEEE Signal Process. Lett.*, vol. 27, pp. 81–85, 2020.

[29] P. Paatero and U. Tapper, "Positive matrix factorization: A non-negative factor model with optimal utilization of error estimates of data values," *Environmetrics*, vol. 5, no. 2, pp. 111–126, Jun. 1994.

[30] 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, Mar. 2007.

[31] A. Huck, M. Guillaume, and J. Blanc-Talon, "Minimum dispersion constrained nonnegative matrix factorization to unmix hyperspectral data," *IEEE Trans. Geosci. Remote Sens.*, vol. 48, no. 6, pp. 2590–2602, Jun. 2010.

[32] A. Zymnis, S.-J. Kim, J. Skaf, M. Parente, and S. Boyd, "Hyperspectral image unmixing via alternating projected subgradients," in *Proc. Asilomar Conf. Signals, Syst. Comput.*, 2007, pp. 1164–1168.

[33] Z. Yang, G. Zhou, S. Xie, S. Ding, J.-M. Yang, and J. Zhang, "Blind spectral unmixing based on sparse nonnegative matrix factorization," *IEEE Trans. Image Process.*, vol. 20, no. 4, pp. 1112–1125, Apr. 2011.

[34] J. Yao, D. Meng, Q. Zhao, W. Cao, and Z. Xu, "Nonconvex-sparsity and nonlocal-smoothness-based blind hyperspectral unmixing," *IEEE Trans. Image Process.*, vol. 28, no. 6, pp. 2991–3006, Jun. 2019.

[35] J. Peng, W. Sun, F. Jiang, H. Chen, Y. Zhou, and Q. Du, "A general loss-based nonnegative matrix factorization for hyperspectral unmixing," *IEEE Geosci. Remote Sens. Lett.*, vol. 19, pp. 1–5, 2022.

[36] J. Peng, Y. Zhou, W. Sun, Q. Du, and L. Xia, "Self-paced nonnegative matrix factorization for hyperspectral unmixing," *IEEE Trans. Geosci. Remote Sens.*, vol. 59, no. 2, pp. 1501–1515, Feb. 2021.

[37] G. Zhao, X. Jia, and C. Zhao, "Multiple endmembers based unmixing using archetypal analysis," in *Proc. IEEE Int. Geosci. Remote Sens. Symp. (IGARSS)*, Jul. 2015, pp. 5039–5042.

[38] M. Mørup and L. K. Hansen, "Archetypal analysis for machine learning and data mining," *Neurocomputing*, vol. 80, pp. 54–63, Mar. 2012.

[39] Y. Chen, J. Mairal, and Z. Harchaoui, "Fast and robust archetypal analysis for representation learning," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2014, pp. 1478–1485.

[40] W. Sun, G. Yang, K. Wu, W. Li, and D. Zhang, "Pure endmember extraction using robust kernel archetypoid analysis for hyperspectral imagery," *ISPRS J. Photogramm. Remote Sens.*, vol. 131, pp. 147–159, Sep. 2017.

[41] J. Nocedal and S. J. Wright, *Numerical Optimization*. Springer, 1999.

[42] M. Xu et al., "L_1 sparsity-constrained archetypal analysis algorithm for hyperspectral unmixing," *IEEE Geosci. Remote Sens. Lett.*, vol. 19, pp. 1–5, 2022.

[43] C.-J. Lin, "Projected gradient methods for nonnegative matrix factorization," *Neural Comput.*, vol. 19, no. 10, pp. 2756–2779, Oct. 2007.

[44] L. Zhang, L. Zhang, and B. Du, "Deep learning for remote sensing data: A technical tutorial on the state of the art," *IEEE Geosci. Remote Sens. Mag.*, vol. 4, no. 2, pp. 22–40, Jun. 2016.

[45] G. A. Licciardi, X. Ceamanos, S. Douté, and J. Chanussot, "Unsupervised nonlinear spectral unmixing by means of NLPCA applied to hyperspectral imagery," in *Proc. IEEE Int. Geosci. Remote Sens. Symp.*, Jul. 2012, pp. 1369–1372.

[46] 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.

[47] 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.

[48] B. Palsson, J. R. Sveinsson, and M. O. Ulfarsson, "Blind hyperspectral unmixing using autoencoders: A critical comparison," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 15, pp. 1340–1372, 2022.

[49] D. Hong et al., "Endmember-guided unmixing network (EGU-Net): A general deep learning framework for self-supervised hyperspectral unmixing," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 33, no. 11, pp. 6518–6531, Nov. 2022.

[50] 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.*, vol. 60, 2022, Art. no. 5503914.

[51] B. Rasti, B. Koirala, P. Scheunders, and J. Chanussot, "MiSiCNet: Minimum simplex convolutional network for deep hyperspectral unmixing," *IEEE Trans. Geosci. Remote Sens.*, vol. 60, 2022, Art. no. 5522815.

[52] D. P. Bertsekas, "Nonlinear programming," *J. Oper. Res. Soc.*, vol. 48, no. 3, p. 334, 1997.

[53] S. Pesme, L. Pillaud-Vivien, and N. Flammarion, "Implicit bias of SGD for diagonal linear networks: A provable benefit of stochasticity," in *Proc. Adv. Neural Inf. Process. Syst. (NeurIPS)*, 2021, pp. 29218–29230.

[54] A. Beck and M. Teboulle, "Mirror descent and nonlinear projected subgradient methods for convex optimization," *Oper. Res. Lett.*, vol. 31, no. 3, pp. 167–175, May 2003.

[55] L. M. Bregman, "The relaxation method of finding the common point of convex sets and its application to the solution of problems in convex programming," *USSR Comput. Math. Math. Phys.*, vol. 7, no. 3, pp. 200–217, Jan. 1967.

[56] M. Teboulle, "Entropic proximal mappings with applications to nonlinear programming," *Math. Oper. Res.*, vol. 17, no. 3, pp. 670–690, Aug. 1992.

[57] J. M. Bioucas-Dias and M. A. T. Figueiredo, "Alternating direction algorithms for constrained sparse regression: Application to hyperspectral unmixing," in *Proc. 2nd Workshop Hyperspectral Image Signal Process., Evol. Remote Sens.*, Jun. 2010, pp. 1–4.

[58] Y. Nesterov, *Introductory Lectures on Convex Optimization: A Basic Course*, vol. 87. Springer, 2003.

[59] T. Hastie, R. Tibshirani, and J. H. Friedman, *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*, vol. 2. Springer, 2009.

[60] M. Elad and A. M. Bruckstein, "A generalized uncertainty principle and sparse representation in pairs of bases," *IEEE Trans. Inf. Theory*, vol. 48, no. 9, pp. 2558–2567, Sep. 2002.

[61] R. Gribonval and M. Nielsen, "Sparse representations in unions of bases," *IEEE Trans. Inf. Theory*, vol. 49, no. 12, pp. 3320–3325, Dec. 2003.

[62] J. Mairal, "Sparse modeling for image and vision processing," *Found. Trends Comput. Graph. Vis.*, vol. 8, nos. 2–3, pp. 283–285, 2014.

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

[64] L. J. Rickard, R. W. Basedow, E. F. Zalewski, P. R. Silverglate, and M. Landers, "HYDICE: An airborne system for hyperspectral imaging," *Proc. ISTE*, vol. 1937, pp. 173–179, Sep. 1993.

[65] M. E. Schaepman et al., "Advanced radiometry measurements and Earth science applications with the airborne prism experiment (APEX)," *Remote Sens. Environ.*, vol. 158, pp. 207–219, Mar. 2015.

[66] F. Zhu, "Hyperspectral unmixing: Ground truth labeling, datasets, benchmark performances and survey," 2017, *arXiv:1708.05125*.

[67] C. Zhou and M. R. D. Rodrigues, "ADMM-based hyperspectral unmixing networks for abundance and endmember estimation," *IEEE Trans. Geosci. Remote Sens.*, vol. 60, 2022, Art. no. 5520018.

[68] B. Rasti, A. Zouaoui, J. Mairal, and J. Chanussot, "SUnAA: Sparse unmixing using archetypal analysis," *IEEE Geosci. Remote Sens. Lett.*, vol. 20, pp. 1–5, 2023.