NNAEU: A SIMPLE AND EFFICIENT CNN-AUTOENCODER FOR HYPERSPECTRAL UNMIXING

Edgard Dabier, Christophe Kervazo, Pietro Gori and Florence Tupin

LTCI, Télécom Paris, Institut Polytechnique de Paris, Palaiseau, France

ABSTRACT

Deep learning-based hyperspectral unmixing (HSU) has followed a trajectory of increasing architectural complexity, from simple autoencoders to transformers and, most recently, Mamba-based models, each claiming to advance the state of the art. Contrary to this trend, we propose a lightweight CNN-based autoencoder, coined "Not New Autoencoder Unmixing" (nnAEU). It is derived from the backbone of UNMamba (2025) through a pruning study that removes the Mamba component entirely. nnAEU uses less than 10 times fewer parameters than UNMamba, and far fewer than other transformer- and GAN-based methods, while outperforming them in accuracy. We provide a complete, minimal, reproducible code, available at *https://gitlab.telecom-paris.fr/ring/nnAEU.git*.

***Index Terms*—** Blind Hyperspectral Unmixing, Remote Sensing, Linear Mixture Model, Self-Supervised Learning.

1. INTRODUCTION

1.1. Context

Hyperspectral images (HSIs) capture multiple narrow, adjacent spectral bands, enabling material analysis in remote sensing with applications in agriculture, mineral mapping, and space exploration [1]. However, due to physical limitations, HSIs often have low spatial resolution: each pixel covers a large area (*e.g.*, $\sim$30 meters for satellite images) and may contain mixed material spectra. This mixing limits direct interpretation, motivating hyperspectral unmixing (HSU), which decomposes pixels into pure material spectra (endmembers) and their relative abundances. Specifically, this article focuses on blind HSU, aiming to estimate both endmembers and abundances. Despite its limitations [2], the Linear Mixture Model (LMM) is widely used for its simplicity. It assumes pixels to be linear combinations of endmembers [1]. Mathematically, for an HSI $Y \in R^{B,N}$ (with $N = H \times W$ pixels and B bands), endmember matrix $E \in R^{B,C}$, and abundance maps $A \in R^{C,N}$, the model is:

$$Y = EA + \varepsilon, \tag{1}$$

where $\varepsilon \in R^{B,N}$ is additive noise. The LMM is typically constrained by non-negativity (ANC) and sum-to-one (ASC) conditions for endmembers and abundances.

1.2. Related work

Due to their simplicity, pure pixel retrieval algorithms such as VCA [3] or N-FINDR [4] gained popularity in HSU. However, their reliance on the pure-pixel assumption —rarely satisfied in practice— led, during the last decade, to the dominance of optimisation-based methods with priors such as sparsity [5] or minimum volume [6], or even learned priors [7].

More recently, the application of deep learning to HSU has followed a trajectory of increasing architectural complexity. Early approaches used simple autoencoders for unsupervised endmember and abundance estimation [8], soon evolving into convolutional variants like CNNAEU [9] and UnDIP [10] to better exploit spatial context. Attention mechanisms and transformers, such as DeepTrans [11] and A2SAN [12], were later introduced to model long-range spectral-spatial dependencies, while adversarial schemes like SSST-GAN [13] enabled blind HSU by learning data distributions. There have also been attempts at using Hyperspectral Foundation Models for HSU [14]. Recently, state-space models (SSMs) such as Mamba have emerged as efficient alternatives, employed in ProMU [15], MBU-Net [16], and UNMamba [17]. At each stage, new architectures grow progressively larger and more intricate. Currently, UNMamba seems to lead the field by outperforming all state-of-the-art models.

1.3. Motivation and Contributions

This trajectory of increasing complexity raises a natural question: is each added layer of complexity actually responsible for the reported gains? In principle, this question should be straightforward to answer by looking at the results presented in the methods' articles. In practice, it rarely is, because HSU research is affected by a broader reproducibility issue in machine learning [18]: code accompanying published methods is frequently unavailable, incomplete, or dependent on undocumented hyperparameters, preprocessing steps, or initialisation schemes, making independent verification of reported gains difficult or, in many cases, impossible.

C. Kervazo would like to thank the ANR agency for the grant ANR-25-CE48-1042.

In this context, the contributions of this work are the following:

- We introduce a light CNN architecture derived from an ablation study we conducted on UNMamba, a neural network matching SOTA. This new architecture, coined nnAEU, sets a new SOTA, while being an order of magnitude lighter than UNMamba.
- We provide a complete, easy-to-implement and reproducible code, tested on several datasets.

2. UNMAMBA: CONTEXT AND PRESENTATION OF THE ABLATION STUDY

2.1. UNMamba Description

UNMamba employs a cascaded spatial–spectral Mamba model tailored for blind HSU. By leveraging state-space models (SSMs), it captures long-range spatial and spectral dependencies with linear computational complexity, ensuring scalability. As illustrated in the top part of Figure 1, the model operates in two stages: abundance estimation and endmember estimation. The abundance module first uses 3 UNMamba blocks that extract long-range spatial dependencies with a spatial block, then global spectral features using a spectral block. This forms a unified spatial–spectral representation that is mapped into abundance estimates. Meanwhile, the endmember module incorporates weighted averages of trainable random sequences. Finally, both abundances and endmembers are combined in the LMM module to reconstruct the input HSI. The authors experimentally show that their method obtains highly accurate results.

2.2. Pruning Procedure

To evaluate the role of these complex modules in UNMamba's performance, we perform a simple pruning of the UNMamba blocks used in the abundance module, following the path of [19]. This leaves the embedding layer, followed by the abundance head. In detail, the embedding module is composed of a 2D convolutional layer reducing the spectral dimension of the input HSI from B to a hidden dimension of 64, a group norm and a SiLU non-linearity. As for the abundance head, it consists of a 2D convolutional layer augmenting the spectral dimension to 128, a group norm, a SiLU, another 2D convolution reducing the spectral dimension to C to reach the abundance size, a batch norm and finally a softmax with scale 3.5. In addition, it uses a dropout layer with probability $5e^{-2}$ after the softmax during training. We keep the endmember and LMM modules untouched, so the pruning procedure only affects the abundance module. As shown in subsection 4.3, the pruned architecture outperforms UNMamba, leading to our new proposed method described in the next section.

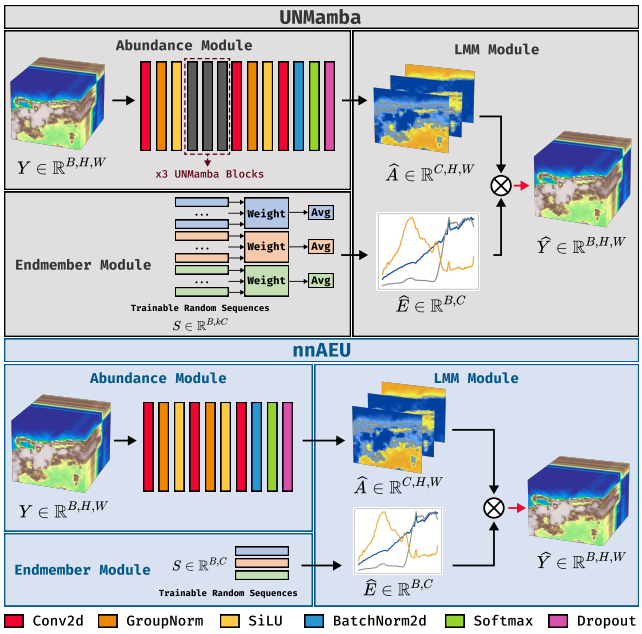


Fig. 1. Comparison of UNMamba and proposed nnAEU unmixing networks. nnAEU is derived from UNMamba by pruning the **UNMamba blocks** and simplifying the endmember module. Y is the input HSI, $\widehat{A}, \widehat{E}$ are the abundance and endmember estimations, $\widehat{Y}$ is the reconstructed HSI and S is the trainable endmember sequence randomly initialised. **Weight** are trainable multiplicative factors and **Avg.** represents the averaging operation.

3. PROPOSED ARCHITECTURE

The proposed architecture consists of 1) a convolutional network estimating the abundance maps, corresponding to the architecture used in UNMamba but removing the Mamba modules; 2) a linear layer producing the endmembers, simplifying the UNMamba endmember module and further improving its performance. The input HSI is then reconstructed in the LMM module by multiplying both estimated endmembers and abundances. We name this model nnAEU, which stands for Not New Autoencoder Unmixing, as the proposed architecture can be viewed as an autoencoder with the abundance module being the encoder and the endmember estimation as the weights of a decoder. The bottom part of Figure 1 illustrates the proposed architecture.

In detail, the encoder uses an alternation of 2D convolutional layers, group norms and SiLU nonlinearities, with the spectral dimension going from B to 48, 128 and finally C. It then uses a 2D batch norm, a softmax to enforce both the ANC and ASC and dropout with probability $5e^{-2}$.

In practice, the input HSI is normalised to have values between 0 and 1. Additionally, we use a typical strategy in HSU by initialising the decoder's weights using a first guess of the endmember, using a Simplex Volume Maximisation algorithm [6].

3.1. Training loss

nnAEU is trained end-to-end in an unsupervised manner by minimising a mixture of losses, taken from the UNMamba article :

$$\text{SAD}(\widehat{Y}, Y) + \lambda_1 \text{MSE}(\widehat{Y}, Y) + \lambda_2 \mathcal{L}_{\frac{1}{2}}(\widehat{A}) + \lambda_3 \sum_{i=0}^{C} \text{MSE}(\widehat{E}_i, \bar{Y}), \quad (2)$$

where SAD is the Spectral Angular Distance (3), and MSE is the Mean Squared Error (4), assessing the quality of the reconstruction. In addition, a $\frac{1}{2}$-norm regularisation term encourages sparsity in the abundance estimation (5), and an MSE between the predicted endmembers $\widehat{E}$ and the average spectrum $\bar{Y}$ assists the convergence of the endmembers towards the data distribution, with $\bar{Y} = \frac{1}{N}\sum_{i=0}^{N} Y_{:,i}$. Mathematically, for any dumb variable matrices $X, \widehat{X} \in R^{D_1, D_2}$, we denote:

$$\text{SAD} = \frac{1}{D_1}\sum_{i=1}^{D_1} \arccos\left(\sum_{j=1}^{D_2} \frac{X_{j,i}}{\|\mathbf{X}_{:,i}\|_2} \frac{\widehat{X}_{j,i}}{\|\widehat{\mathbf{X}}_{:,i}\|_2}\right), \quad (3)$$

$$\text{MSE} = \frac{1}{D_1}\sum_{i=1}^{D_1} (\widehat{X}_{:,i} - X_{:,i})^2, \quad (4)$$

$$\mathcal{L}_{\frac{1}{2}}(\widehat{A}) = \frac{1}{CN}\sum_{i=0}^{C}\sum_{j=0}^{N} \sqrt{\widehat{A}_{i,j}}. \quad (5)$$

4. EXPERIMENTS

4.1. Data Description

Experiments were conducted on several available labelled datasets, providing an in-depth comparison of the methods :

- **Samson**: the smallest dataset, relatively simple, with a spatial size of 95×95, 156 spectral bands, and 3 endmembers.
- **Jasper Ridge**: it has 100×100 pixels, 198 spectral bands and 4 endmembers.
- **Apex**: has 110×110 pixels, 285 spectral bands and contains 4 endmembers.
- **Urban**: has a significantly larger spatial extent of 307 pixels, 162 spectral bands and 6 endmembers

4.2. Experimental Setup

The estimated endmember quality is assessed through the SAD (3) and the abundance through the NMSE (6). Both are averaged over all endmembers.

$$\text{NMSE} = \frac{\sum_{i=1}^{C}\sum_{j=1}^{N}(A_{\text{GT}_{i,j}} - \widehat{A}_{i,j})^2}{\sum_{i=1}^{C}\sum_{j=1}^{N}(A_{\text{GT}_{i,j}})^2}. \quad (6)$$

We also compare the execution time in seconds (*Time*), measured on an NVIDIA V100 GPU, as well as the number of parameters (*Size*) for all methods. Additionally, we compute a score s_i for each method i, aggregating over all the datasets $j \in \{1..J\}$ (here we consider $J = 4$ datasets) the SAD and NMSE after a per-dataset normalisation. We average this score to obtain a global ranking of the methods (*Rank*), for easier readability. The score is obtained with:

$$s_i = \frac{1}{2J}\sum_{j=1}^{J}\left(\frac{\text{SAD}_{i,j}}{\max_j(\text{SAD}_{i,j})} + \frac{\text{NMSE}_{i,j}}{\max_j(\text{NMSE}_{i,j})}\right), \quad (7)$$

where $\text{SAD}_{i,j}$ and $\text{NMSE}_{i,j}$ are the metrics of method i on dataset j. We also include a reproducibility item (*Repr.*) to highlight methods for which announced performance could or couldn't be reproduced. The training and loss hyperparameters are chosen to be the same across all the datasets. They were obtained using the Optuna Bayesian optimisation framework [20]: 300 training epochs, with the AdamW optimiser, a weight decay of $3e^{-3}$, a learning rate of $7e^{-3}$ and $\lambda_1 = 0.8, \lambda_2 = 3e^{-2}$ and $\lambda_3 = 1$.

4.3. UNMamba Pruning

Table 1 shows results obtained with the ablation described in subsection 2.2, where UNMamba-p denotes the pruning of Mamba blocks. This shows that this ablation doesn't affect the performance of the model; in fact, it can even improve unmixing capacity while significantly reducing model size (by a factor of 3 to 5). This size reduction speeds up training and shrinks the memory footprint, making it better suited for embedded systems.

4.4. Results

To evaluate our proposed framework, we compare averaged scores from 10 independent training runs against 9 state-of-the-art methods, all tested using the authors' recommended hyperparameters. From classic optimization method using SiVM and FCLS [6, 21], to CNNs with CNNAEU [9] and UnDIP [10], ViTs with DeepTrans [10], A2SAN [12], GAN with SSST-GAN [13], and Mamba with ProMU [15], MBU-Net [16] [1], and UNMamba [17]. Qualitative results about the estimation of abundances and endmembers on the Apex dataset are shown in Figures 2 and 3. In addition, Table 2 displays the results obtained using the proposed unmixing autoencoder. Moreover, as mentioned in subsection 1.3, it is important to note that some methods' results may differ from their original paper due to reproducibility issues with the available code.

We observe that, as mentioned in section 3, our proposed simplification of the endmember module further improves the performance compared to using UNMamba's (Table 1). This choice reduces the size of the already pruned UNMamba model, reaching up to 20 times fewer parameters than UNMamba, placing it among the smallest and fastest learning-based models of the state-of-the-art. In addition, this method

[1] we could not get results on the Urban data set for ProMU using the available code due to missmatch in parameters, and for MBU-Net due to exceeding memory issues

	Samson				Jasper				Apex				Urban				Score ↓
	SAD ↓	NMSE ↓	Time	Size	SAD ↓	NMSE ↓	Time	Size	SAD ↓	NMSE ↓	Time	Size	SAD ↓	NMSE ↓	Time	Size	
UNMamba [17]	$0.025_{\pm 0.01}$	**$0.014_{\pm 0}$**	32	194k	$0.095_{\pm 0.03}$	$0.033_{\pm 0.04}$	40	242k	$0.139_{\pm 0.06}$	$0.116_{\pm 0.05}$	38	260k	$0.165_{\pm 0.05}$	**$0.215_{\pm 0.13}$**	202	3.8M	1.23
UNMamba-p	**$0.024_{\pm 0}$**	$0.016_{\pm 0}$	9	43k	**$0.065_{\pm 0.01}$**	**$0.019_{\pm 0.01}$**	12	62k	$0.122_{\pm 0.02}$	**$0.096_{\pm 0.02}$**	9	85k	**$0.148_{\pm 0.04}$**	$0.217_{\pm 0.09}$	83	69k	**0.81**

Table 1. Comparison of the performance of UNMamba and UNMamba-p, without the UNMamba blocks. *Time* is in seconds, *Score* is detailed in subsection 4.2, and *Size* is the number of trainable parameters. Best results in **bold**, second best underlined

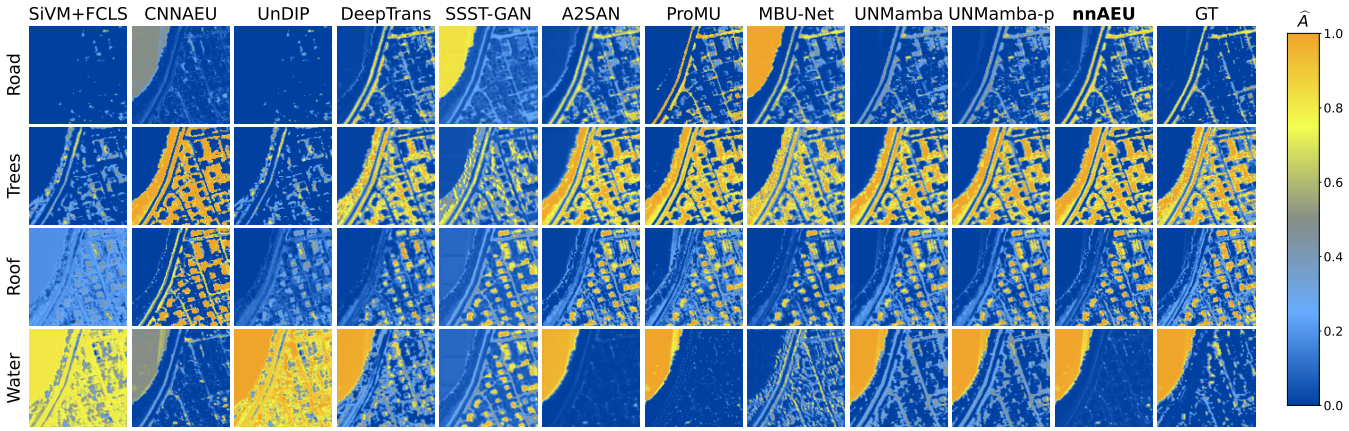


Fig. 2. Comparison of all method's estimated $\widehat{A}$ on the Apex dataset, our proposed method is in **bold**.

sets a new state-of-the-art, beating all other approaches to unmixing, surpassing the pruned version of UNMamba.

Our findings should be read with two limitations in mind: our evaluation is restricted to standard benchmark scenes, itself a consequence of HSU's scarce, costly-to-obtain ground truth; and Mamba- or transformer-based components may still prove useful in other regimes — larger datasets, more endmembers, different sensors — where more data could justify their added capacity.

5. CONCLUSION

Why doesn't added complexity improve results? A likely explanation lies in the single-image training framework, dominant in HSU: heavy, expressive models are trained on a single scene with little data, leaving limited room for their extra capacity to be exploited — the convolutional backbone alone may already capture what signal is available.

Within current benchmark conditions, our results support a clear message: complexity should not be treated as a proxy for progress. We demonstrated this through a controlled ablation of UNMamba, and proposed a light, fully reproducible CNN architecture matching SOTA at an order of magnitude fewer parameters, offered as a simple baseline for the field.

6. REFERENCES

[1] Longfei Ren, Zhu Han, et al., "Advances in hyperspectral image unmixing: From algorithmic frameworks to practical applications," *Information Geography*, vol. 2, no. 1, pp. 100035, 2026.

[2] Christophe Kervazo, Nicolas Gillis, et al., "Provably robust blind source separation of linear-quadratic near-separable mixtures," *SIAM Journal on Imaging Sciences*, vol. 14, no. 4, pp. 1848–1889, 2021.

[3] J.M.P. Nascimento and J.M.B. Dias, "Vertex component analysis: a fast algorithm to unmix hyperspectral data," *IEEE TGRS*, vol. 43, no. 4, pp. 898–910, 2005.

[4] Michael E. Winter, "N-findr: an algorithm for fast autonomous spectral end-member determination in hyperspectral data," in *Imaging Spectrometry V*. SPIE, 1999, vol. 3753, pp. 266 – 275.

[5] José M. Bioucas-Dias and Mario A. T. Figueiredo, "Alternating direction algorithms for constrained sparse regression: Application to hsu," in *WHISPERS*, 2010, pp. 1–4.

[6] Rob Heylen, Dževdet Burazerovic, et al., "Non-linear spectral unmixing by geodesic simplex volume maximization," *IEEE JSTSP*, vol. 5, no. 3, pp. 534–542, 2011.

[7] Christophe Kervazo and Jérémy E Cohen, "Unrolled multiplicative updates for nonnegative matrix factorization applied to hyperspectral unmixing," 2026.

[8] Burkni Palsson, Jakob Sigurdsson, et al., "Hyperspectral unmixing using a neural network autoencoder," *IEEE Access*, vol. 6, pp. 25646–25656, 2018.

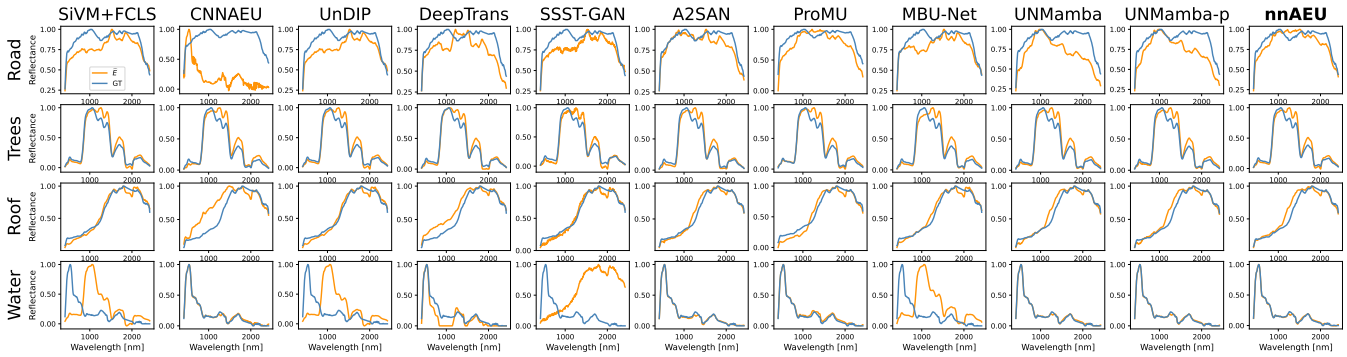


Fig. 3. Comparison of all method's estimated $\widehat{E}$ on the Apex dataset, our proposed method is in **bold**.

	Samson				Jasper				Apex				Urban				Repr.	Rank
	SAD ↓	NMSE ↓	Time	Size	SAD ↓	NMSE ↓	Time	Size	SAD ↓	NMSE ↓	Time	Size	SAD ↓	NMSE ↓	Time	Size		
SiVM + FCLS [21]	$0.056_{\pm 0}$	$0.438_{\pm 0}$	0	0	$0.261_{\pm 0}$	$0.187_{\pm 0}$	0	0	$0.244_{\pm 0}$	$0.943_{\pm 0}$	0.1	0	$0.436_{\pm 0}$	$1.293_{\pm 0}$	0.1	0	✔	7
CNNAEU [9]	$0.058_{\pm 0.08}$	$0.112_{\pm 0.11}$	1.4	124k	$0.253_{\pm 0.17}$	$0.328_{\pm 0.16}$	1.4	182k	$0.362_{\pm 0.24}$	$0.263_{\pm 0.12}$	1.5	261k	$0.238_{\pm 0.07}$	$0.712_{\pm 0.24}$	5.4	188k	✗	5
UnDIP [10]	$0.056_{\pm 0}$	$0.453_{\pm 0}$	57	1.6M	$0.261_{\pm 0}$	$0.191_{\pm 0}$	72	1.7M	$0.244_{\pm 0}$	$1.003_{\pm 0}$	108	1.9M	$0.436_{\pm 0}$	$1.242_{\pm 0.01}$	269	1.6M	✗	8
DeepTrans [11]	$0.198_{\pm 0.01}$	$0.346_{\pm 0}$	3.2	5M	$0.282_{\pm 0.01}$	$0.255_{\pm 0.01}$	3.2	7.5M	$0.289_{\pm 0}$	$0.445_{\pm 0.01}$	3	8M	$0.369_{\pm 0.03}$	$0.786_{\pm 0.11}$	9	34M	✗	10
SSST-GAN [13]	$0.136_{\pm 0.01}$	$0.078_{\pm 0.03}$	173	175M	$0.304_{\pm 0.06}$	$0.259_{\pm 0.04}$	172	175M	$0.380_{\pm 0.09}$	$0.808_{\pm 0.03}$	179	175M	$0.334_{\pm 0.09}$	$0.617_{\pm 0.17}$	175	175M	✔	9
A2SAN [12]	$0.060_{\pm 0.01}$	**$0.014_{\pm 0}$**	14	37k	$0.136_{\pm 0.03}$	$0.089_{\pm 0.05}$	14	43k	$0.080_{\pm 0.01}$	$0117_{\pm 0.01}$	14	56k	$0.221_{\pm 0.06}$	$0.409_{\pm 0.17}$	16	38k	✔	3
ProMU [15]	$0.106_{\pm 0}$	$0.018_{\pm 0}$	431	52k	$0.278_{\pm 0.01}$	$0.364_{\pm 0.03}$	252	17k	$0.146_{\pm 0.05}$	$0.188_{\pm 0.10}$	582	18k	//	//	//	//	✗	6
MBU-Net [16]	$0.028_{\pm 0}$	$0.052_{\pm 0}$	144	199k	$0.144_{\pm 0.01}$	$0.045_{\pm 0.01}$	101	144k	$0.293_{\pm 0.04}$	$0.941_{\pm 0.26}$	145	194k	//	//	//	//	✗	4
UNMamba [17]	**$0.025_{\pm 0.01}$**	**$0.014_{\pm 0}$**	32	194k	$0.095_{\pm 0.03}$	$0.033_{\pm 0.04}$	40	242k	$0.139_{\pm 0.06}$	**$0.116_{\pm 0.05}$**	38	260k	$0.165_{\pm 0.05}$	$0.215_{\pm 0.13}$	202	3.8M	✔	2
nnAEU (ours)	$0.027_{\pm 0}$	$0.015_{\pm 0}$	2	15k	**$0.066_{\pm 0.02}$**	**$0.032_{\pm 0.01}$**	2	17k	**$0.064_{\pm 0}$**	$0.127_{\pm 0.01}$	2	22k	**$0.111_{\pm 0}$**	**$0.149_{\pm 0.03}$**	4	16k	✔	**1**

Table 2. Comparison of all HSU methods on the 4 datasets, averaged over 10 runs. We display results as $\text{Mean}_{\pm \text{ std}}$, *Time* (in seconds), *Rank* (as described in 4.2), *Size* (number of trainable parameters), and *Repr.* indicates whether the method's announced performance could be replicated or not.

[9] Burkni Palsson, Magnus O. Ulfarsson, et al., "Convolutional autoencoder for spectral–spatial hsu," *IEEE TGRS*, vol. 59, no. 1, pp. 535–549, 2021.

[10] Behnood Rasti, Bikram Koirala, et al., "Undip: Hsu using deep image prior," *IEEE TGRS*, vol. 60, pp. 1–15, 2022.

[11] Preetam Ghosh, Swalpa Kumar Roy, et al., "Hsu using transformer network," *IEEE TGRS*, vol. 60, pp. 1–16, 2022.

[12] Xuanwen Tao, Mercedes E. Paoletti, et al., "An abundance-guided attention network for hsu," *IEEE TGRS*, vol. 62, pp. 1–14, 2024.

[13] Yu Zhang, Jiageng Huang, et al., "Ssst-gan: A sampling-based spatial-spectral transformer and generative adversarial network for hsu," *IEEE J-STARS*, vol. 19, pp. 5741–5757, 2026.

[14] Edgard Dabier, Christophe Kervazo, et al., "Benchmarking hyperspectral foundation models for hyperspectral unmixing," *arXiv*, 2026.

[15] Yang Liu, Shujun Liu, et al., "Efficient progressive mamba model for hyperspectral sequence unmixing," *IEEE J-STARS*, vol. 18, pp. 19511–19526, 2025.

[16] Mingle Zhang, Hongyu Xie, et al., "Mamba-enhanced spatial–spectral feature learning for hsu," *IEEE JSTARS*, vol. 18, pp. 22798–22815, 2025.

[17] Dong Chen, Junping Zhang, et al., "Unmamba: Cascaded spatial–spectral mamba for blind hsu," *IEEE GRSL*, vol. 22, pp. 1–5, 2025.

[18] J. Pineau, P. Vincent-Lamarre, et al., "Improving reproducibility in machine learning research(a report from the neurips 2019 reproducibility program)," *JMLR*, 2021.

[19] Weihao Yu and Xinchao Wang, "Mambaout: Do we really need mamba for vision?," in *CVPR*, 2025.

[20] Takuya Akiba, Shotaro Sano, et al., "Optuna: A next-generation hyperparameter optimization framework," in *ACM SIGKDD*, 2019.

[21] D. Heinz, C.-I. Chang, et al., "Fully constrained least-squares based linear unmixing [hyperspectral image classification]," in *IEEE IGARSS (Cat. No.99CH36293)*, 1999, vol. 2, pp. 1401–1403 vol.2.