

---

# Spectral Unmixing With Multinomial Mixture Kernel and Wasserstein Generative Adversarial Loss

---

Savas Ozkan, Gozde Bozdagi Akar

Department of Electrical and Electronics Engineering  
Middle East Technical University  
Ankara, Turkey

## Abstract

This study proposes a novel framework for spectral unmixing by using 1D convolution kernels and spectral uncertainty. High-level representations are computed from data, and they are further modeled with the Multinomial Mixture Model to estimate fractions under severe spectral uncertainty. Furthermore, a new trainable uncertainty term based on a nonlinear neural network model is introduced in the reconstruction step. All uncertainty models are optimized by Wasserstein Generative Adversarial Network (WGAN) to improve stability and capture uncertainty. Experiments are performed on both real and synthetic datasets. The results validate that the proposed method obtains state-of-the-art performance, especially for the real datasets compared to the baselines. Project page at: <https://github.com/savasozkan/dscn>.

## 1 Introduction

Spectral data provides comprehensive knowledge about the Earth’s surface in the form of narrow-band spectra. The limitations of spectral sensors induce pure materials in a scene to be mixed in different fractions. Indeed, the foremost concern is that pure materials (i.e., endmembers)  $\mathbf{E} = [\mathbf{e}_1, \mathbf{e}_2, \dots, \mathbf{e}_K] \in \mathbb{R}^{D \times K}$  and their fractions  $\mathbf{y} = [y_1, y_2, \dots, y_K] \in \mathbb{R}^K$  have to be estimated blindly from spectral data  $\mathbf{x} \in \mathbb{R}^D$ . Furthermore, nonlinear relations and spectral uncertainty are ubiquitous with severe scattering conditions (see Appendix A).

Even if the mixture of data can be represented with linear/nonlinear formulations, these assumptions underestimate the solution by considering pure materials as a finite set towards spectral uncertainty. To overcome the limitations, the mathematical formulation is redefined as follows in (1):

$$\mathbf{x} = \sum_{k=1}^K (\mathbf{e}_k + \gamma_k) y_k + \eta = \mathbf{E} \mathbf{y} + \eta, s.t. y_k \geq 0, \sum_{k=1}^K y_k = 1 \quad (1)$$

Here  $K$  is the number of endmembers, and  $D$  is the spectral bands of a pixel. Also,  $\eta \sim \mathcal{N}(0, 1)$  is the additive Gaussian noise to account for possible noise sources. Note that  $\gamma_k$  varies for each pixel and material. Hence, this term virtually defines the uncertainty of the data.

In the scope of this paper, we will present a novel framework that broadly uses this equation by computing fractions from pre-computed material signatures (i.e., only fractions are estimated from data). Specifically, we introduce a multinomial mixture kernel based on a neural network (NN) structure that obtains the fractions per-pixel. The learnable parameters of this kernel are optimized with WGAN for the first time. This step achieves theoretically more accurate and stable solutions.

## 2 Related Work

As expected, the initial methods are based on linear models due to their simplicity. In particular, sum-to-one and non-negative constraints are conditioned for least-square solutions (2). However, these models are insufficient to resolve complex cases such as multiple scattering effects and microscopic-level material mixtures. Hence, bilinear models are used to handle nonlinear scenarios in which bilinear interactions of linear relations are considered in the models (3; 4).

Even if the nonlinearity is retained, spectral uncertainty is still discarded in these models due to the finite set of materials. Distribution-based models represent endmember signatures as they are sampled from some distributions. In the literature, various distributions are exploited (5; 6). Notice that these models approximate each fraction to a unimodal distribution. Ultimately, this undermines the representation capacity as expected. Furthermore, the Gaussian Mixture Model (GMM) is adopted (1). However, the complexity and assumptions limit the applicability of this method for large-scale data.

Recently, AutoEncoder (AE) network is redefined with additional layers and a spectral angle distance metric (7) to unmix data that improves robustness for spectral uncertainty. Moreover, (8) uses a stacked deeper AE network and a robust initialization is achieved by matrix factorization.

### 3 Proposed Model

We first describe the basics of the proposed method (see Appendix B). The architecture is equivalent to the standard AE formulation where it encodes the input spectra  $\mathbf{x}$  to a hidden latent representation  $\hat{\mathbf{y}} = Enc(\mathbf{x}; \theta^e)$  and then decodes to recompute the reconstructed version of the original spectra  $\hat{\mathbf{x}} = Dec(\hat{\mathbf{y}}; \theta^d)$  by optimizing the learnable parameters  $\theta^e$  and  $\theta^d$  with some loss function.

#### 3.1 Encoder Part

The dimensionality of data is one of the drawbacks, especially for unsupervised learning. The main reason is that a high-dimensional space causes irregular distributions of data in the learning process. Inevitably, overfitting occurs for the parameters. For additional background, we recommend (9; 10).

For this purpose, high-dimensional input data  $\mathbf{x}$  is mapped to a low-dimensional space  $\mathbf{z} \in \mathbb{R}^M$  where  $M$  denotes the dimension of latent representations. Hence, a latent representation is computed with a set of learnable parameters  $\theta^c$  as  $\mathbf{z} = H(\mathbf{x}; \theta^c)$ . Observe that  $M$  value should be selected as  $M \gg K$  to extract sparse representations. In particular, 1D convolution layers are utilized with batch normalization and sparse activation layers in this function. Since fewer parameters are optimized, the stack of convolutions boosts the representation capacity with deep hierarchies. Details can be found in Appendix C for the NN structure with input/output relations.

Later, fractions  $\hat{\mathbf{y}}$  are estimated by using these latent representations  $\mathbf{z}$  with a multinomial function  $\hat{\mathbf{y}} = G(\mathbf{z}; \theta^p)$  where  $\theta^p$  is the learnable parameters. A NN multinomial mixture model is utilized for this case. As explained, distribution-based methods reproduce materials as they are sampled from probability distributions. Initial methods employ a unimodal distribution as:

$$p(\mathbf{y}|\mathbf{z}) \sim \mathcal{N}(\mathbf{z}|\mu, \Sigma) \quad (2)$$

Fractions  $\mathbf{y}$  are estimated from  $\mathbf{z}$  by using a Gaussian distribution where  $\mu$  and  $\Sigma$  indicate the mean and covariance matrix, respectively. In fact, this formulation underestimates the solution by fitting a unimodal distribution for each material. Therefore, Eq. 2 is restored with the Gaussian Mixture Model (GMM) by computing fractions with the mixture of Gaussian distributions:

$$p(\mathbf{y}|\mathbf{z}) \sim G(\mathbf{z}; \theta^p) = \sum_{n=1}^N \alpha_n \mathcal{N}(\mathbf{z}|\mu_n, \Sigma_n) \quad (3)$$

Here,  $N$  is the total number of mixture components by constraining  $\alpha_n \geq 0$  and  $\sum_{n=1}^N \alpha_n = 1$ . Multinomial distributions are adopted in our method because fractions can be overly represented so that spectral uncertainty is adequately handled. Furthermore,  $N$  should be selected as  $N \gg K$  for effective representations. To this end, we mimic this approach with a set of NN models. We prefer to use NN models since the overall framework can be trained in an end-to-end manner. Note that similar attempts are also made in the literature (11).

If we extend Eq. 3 to separate the term into two learnable parts, this equation is expressed as  $G(\mathbf{z}; \theta^p) = \sum_{n=1}^N \beta_n g_n(\mathbf{z})$  where  $g_n(\mathbf{z}) = \exp(\frac{-1}{2}(\mathbf{z} - \mu_n)^T \Sigma_n^{-1}(\mathbf{z} - \mu_n))$  and  $\beta_n = \frac{\alpha_n}{(2\pi)^{\frac{N}{2}} |\Sigma_n|^{\frac{1}{2}}}$ . As noticed, at least two sets of parameters must be optimized, thus two separate single-layer subnetworks are employed as  $v_n(\mathbf{z}; \theta^{p1}) = g_n(\mathbf{z})$  and  $c_n(\mathbf{z}; \theta^{p2}) = \beta_n$ .

To model these terms, the quadratic form of  $g_n(\mathbf{z})$  is renounced in our formation. Precisely, multivariate Mahalanobis distance is normalized with a sigmoid function to calculate the probabilistic relations. To this end, the normal distribution resembles a cumulative distribution function. Similarly,  $\beta_n$  term

is calculated with a NN model, and the sum-to-one constraint is retained with softmax activation. In the end, all subparts of the encoder can be trained with differentiable layers.

### 3.1.1 Parameter Updates for Multinomial Mixture Kernel

Expectation-Maximization (EM) is the most popular technique to estimate GMM parameters. This approach guarantees a stable solution by optimizing the negative log-likelihood function. Note that our loss function exploits the negative log-likelihood term for parameter updates. However, the studies show that EM approach suffers to achieve a true solution for several reasons, such as the random initialization of parameters and discontinuity. For further details, we recommend to read (12; 13).

To overcome these limitations, WGAN is employed, which provides several theoretical benefits over the classical approaches (12). Specifically, Wasserstein distance is continuous and differentiable almost everywhere. Also, since it has a smooth loss function, the initialization is not critical. Formally, Wasserstein distance  $W(p, q)$  measures the distance between two distributions  $p$  and  $q$  where they generate variables as  $\mathbf{x}^p$  and  $\mathbf{x}^q$ . Kantorovich-Rubinstein duality minimizes the distance as:

$$\max \|f\|_L \leq 1(\mathbb{E}_{\mathbf{x}^p}[f(\mathbf{x}^p)] - \mathbb{E}_{\mathbf{x}^q}[f(\mathbf{x}^q)]) \quad (4)$$

where  $f(\cdot)$  is a 1-Lipschitz function. Inspired by the structure of Variational AutoEncoder (VAE) (14), this duality can be adopted to our formulation in Eqs. 1 and 3 as:

$$\max \|f\|_L \leq 1\left(\mathbb{E}_{\mathbf{x}}[f(\mathbf{x})] - \mathbb{E}_{\hat{\mathbf{x}}}\left[f\left(\sum_{k=1}^K (\mathbf{e}_k + \gamma_k) \sum_{n=1}^N \beta_n g_n(\mathbf{z})\right)\right]\right) \quad (5)$$

Here  $\gamma_k$  is omitted for the simplicity and  $\beta_n$  value is accepted as not random. Note that this simplification is resolved in the decoder part. In addition, the generated sample from multinomial components is denoted as  $\hat{\mathbf{x}}$ . With these observations, the formulation is simplified as:

$$\max \|f\|_L \leq 1\left(\mathbb{E}_{\mathbf{x}}[f(\mathbf{x})] - \sum_{k=1}^K \mathbf{e}_k \sum_{n=1}^N \beta_n \mathbb{E}_{\hat{\mathbf{x}}}[f(g_n(\mathbf{z}))]\right) \quad (6)$$

To this end, this notion mathematically proves that WGAN loss can be used for the parameter updates of  $g_n(z)$  terms. Furthermore, an additional gradient penalty term is used for stable solutions (15):

$$\mathcal{L}_{disc} = \mathbb{E}[d(\mathbf{x}; \theta^d)] - \mathbb{E}[d(\hat{\mathbf{x}}; \theta^d)] + \lambda_{pq} \mathbb{E}[(\|d(\tilde{\mathbf{x}}; \theta^d)\|_2 - 1)^2]. \quad (7)$$

Here,  $d(\cdot, \cdot)$  is the discriminative module and it is a 1-Lipschitz function. Furthermore,  $\tilde{\mathbf{x}}$  is uniformly sampled data from real  $\mathbf{x}$  and reconstructed  $\hat{\mathbf{x}}$  data.  $\lambda_{pq}$  scales the contribution of penalty term that is set to 10. The details for the NN structure of discriminative module can be found in Appendix C.

## 3.2 Decoder Part

As stated, in Eq. 1, the formulation defines  $\gamma_k$  value for spectral uncertainty. However, we omit this term in Eq. 6 for the simplicity. Here, we remodel this term with extra NN layers.

For this purpose, we add an uncertainty NN model  $u(\hat{\mathbf{y}}, \eta; \theta^u)$  that takes the output of multinomial mixture kernel and random noise as input. This model computes an additive full-size spectral signature. Observe that there is random noise  $\eta$  to mimic the spectral uncertainty for different pixels in the formulation. Lastly, WGAN loss is also used for these parameters. Moreover, we utilize a refinement NN model  $u(\hat{\mathbf{y}}; \theta^r)$  that allows updates and nonlinearity for the precomputed endmembers.

The final output of the decoder part is reformulated by rescaling the contributions of refinement and uncertainty NN models with  $\lambda_r = 0.05$  and  $\lambda_u = 0.1$ , respectively:

$$\hat{\mathbf{x}} = Dec(\hat{\mathbf{y}}; \theta_d) = \mathbf{E}\hat{\mathbf{y}} + \lambda_u u(\hat{\mathbf{y}}, \eta; \theta^u) + \lambda_r u(\hat{\mathbf{y}}; \theta^r) \quad (8)$$

## 3.3 Implementation Details

For the parameter updates, spectral angle distance based loss function is utilized (7). WGAN loss is also used for the parameters at encoder and decoder parts, as explained. Adam stochastic optimizer is used by setting  $\beta_1$  as 0.7 (16).

Table 1: RMSE performance for Urban and Jasper datasets. Best results are bold.

	Urban Dataset			Jasper Dataset		
$\times 10^{-2}$	DMaxD	SCM	EndNet	DMaxD	SCM	EndNet
N= 4	15.96 $\pm$ 2.6	13.26 $\pm$ 0.6	8.27 $\pm$ 0.4	<b>12.03</b> $\pm$ 0.7	18.10 $\pm$ 0.9	6.38 $\pm$ 0.7
N= 8	<b>15.47</b> $\pm$ 1.9	13.29 $\pm$ 0.8	8.05 $\pm$ 0.5	12.29 $\pm$ 0.9	18.33 $\pm$ 1.1	6.61 $\pm$ 0.7
N= 16	16.32 $\pm$ 3.2	13.24 $\pm$ 0.5	8.16 $\pm$ 0.4	12.12 $\pm$ 0.7	<b>17.92</b> $\pm$ 0.8	<b>6.31</b> $\pm$ 0.7
N= 24	16.56 $\pm$ 2.1	<b>13.23</b> $\pm$ 0.8	<b>8.04</b> $\pm$ 0.3	12.27 $\pm$ 0.6	18.03 $\pm$ 1.1	6.62 $\pm$ 1.1
w/o EU	34.14 $\pm$ 0.3	15.91 $\pm$ 0.3	8.38 $\pm$ 0.4	14.78 $\pm$ 0.4	20.26 $\pm$ 0.4	8.36 $\pm$ 0.5
w/o WGAN	31.43 $\pm$ 1.6	15.76 $\pm$ 1.0	8.52 $\pm$ 0.5	15.63 $\pm$ 0.3	19.03 $\pm$ 0.4	10.15 $\pm$ 0.5

Table 2: RMSE performance on Urban and Jasper datasets. Best results are bold.

	Urban Dataset			Jasper Dataset		
$\times 10^{-2}$	DMaxD	SCM	EndNet	DMaxD	SCM	EndNet
LMM	31.04	28.00	27.59	15.31	19.78	22.60
GBM	31.04	27.00	27.08	15.98	19.95	21.91
PPNM	36.54	28.17	15.06	12.99	19.78	14.72
MLM	31.43	16.34	9.48	16.59	20.73	8.13
NCM	31.04	28.00	27.51	15.29	19.56	22.42
DSCN++	<b>15.47</b>	<b>13.23</b>	<b>8.04</b>	<b>12.03</b>	<b>17.92</b>	<b>6.31</b>

## 4 Experiments

Experiments to compare the performance of the proposed method (DSCN++) with the baseline techniques are conducted on two real datasets (Urban (17) and Jasper (18) datasets) (see Appendix D for additional experiments). For this purpose, Linear Mixture Model (LMM) (19), Generalized Bilinear Model (GBM) (3), Post-Nonlinear Mixing Model (PPNM) (20), Multilinear Mixing Model (MLM) (19) and Normal Compositional Model (NCM) (21) are selected that are extensively used nonlinear and distribution-based methods in the literature. Scores of the baselines are reproduced.

Furthermore, Spatial Compositional Model (SCM) (22), Distance-MaxD (DMaxD) (23) and Sparse AutoEncoder Network (EndNet) (7) are exploited to estimate endmember spectra from the scenes. Scores are reported in terms of Root Mean Square Error (RMSE) and tests are repeated 20 times.

First, we test all individual steps and parameter configurations for different datasets. For clarity, EU stands for NN models that correspond to uncertainty and refinement models in the decoder part. Furthermore, WGAN indicates the use of Wasserstein GAN in parameter optimization.  $N$  denotes the total number of mixture components. Experimental results are reported in Tab. 1. An important observation is that all these contributions ultimately improve performance. In particular, both WGAN and EU steps improve the performance of DmaxD methods for both datasets. A similar observation can be made for SCM on the Urban dataset and Endnet on the Jasper dataset. Lastly, the selection of  $N$  value larger than the material number  $K$  ultimately advances the unmixing performance.

Second, we also compare our method with the baselines. Results are demonstrated in Tab. 2. Observe that no single baseline method can continuously yield accurate results for different scenes, even if these baselines cover both nonlinear and distribution-based methods. This observation shows the superiority of our method that can capture both nonlinearity and uncertainty conditions perfectly. Of course, our method obtains state-of-the-art performance compared to baselines by a large margin.

## 5 Conclusion

In this paper, we introduce a novel neural network (NN) framework for spectral unmixing. It is composed of 1D convolutions by addressing spectral uncertainty, which is one of the severe physical conditions frequently observed in real data. For this purpose, a novel NN layer is proposed to estimate the fractions per-pixel by leveraging the multinomial mixture kernel. Wasserstein GAN is exploited for the parameter optimization, which leads to more accurate and stable performance than other loss functions. The experimental results validate that the proposed method outperforms baselines by a large margin.

## References

- [1] Y. Zhou, A. Rangarajan, and P. D. Gader, "A gaussian mixture model representation of endmember variability in hyperspectral unmixing," *IEEE Transactions on Image Processing*, vol. 27, no. 5, pp. 2242–2256, 2018.
- [2] R. Heylen *et al.*, "Fully constrained least squares spectral unmixing by simplex projection," *IEEE TGRS*, pp. 4112–4122, 2011.
- [3] A. Halimi, Y. Altmann, N. Dobigeon, and J.-Y. Tourneret, "Nonlinear unmixing of hyperspectral images using a generalized bilinear model," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 49, no. 11, pp. 4153–4162, 2011.
- [4] Y. Altmann, N. Dobigeon, and J.-Y. Tourneret, "Bilinear models for nonlinear unmixing of hyperspectral images," in *Hyperspectral Image and Signal Processing: Evolution in Remote Sensing (WHISPERS), 2011 3rd Workshop on*, pp. 1–4, IEEE, 2011.
- [5] B. Zhang, L. Zhuang, L. Gao, W. Luo, Q. Ran, and Q. Du, "Pso-em: A hyperspectral unmixing algorithm based on normal compositional model," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 52, no. 12, pp. 7782–7792, 2014.
- [6] X. Du, A. Zare, P. Gader, and D. Dranishnikov, "Spatial and spectral unmixing using the beta compositional model," *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, vol. 7, no. 6, pp. 1994–2003, 2014.
- [7] S. Ozkan, B. Kaya, and G. B. Akar, "Endnet: Sparse autoencoder network for endmember extraction and hyperspectral unmixing," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 57, no. 1, pp. 482–496, 2018.
- [8] Y. Su, J. Li, A. Plaza, A. Marinoni, P. Gamba, and S. Chakravorty, "Daen: Deep autoencoder networks for hyperspectral unmixing," *IEEE Transactions on Geoscience and Remote Sensing*, 2019.
- [9] M. Norouzi and D. M. Blei, "Minimal loss hashing for compact binary codes," in *Proceedings of the 28th international conference on machine learning (ICML-11)*, pp. 353–360, Citeseer, 2011.
- [10] H. Jegou, M. Douze, and C. Schmid, "Product quantization for nearest neighbor search," *IEEE transactions on pattern analysis and machine intelligence*, vol. 33, no. 1, pp. 117–128, 2011.
- [11] S. Jalali, C. Nuzman, and I. Saniee, "Efficient deep approximation of gmms," in *Advances in Neural Information Processing Systems*, pp. 4552–4560, 2019.
- [12] S. Kolouri, G. K. Rohde, and H. Hoffman, "Sliced wasserstein distance for learning gaussian mixture models," *arXiv preprint arXiv:1711.05376*, 2017.
- [13] C. Jin, Y. Zhang, S. Balakrishnan, M. J. Wainwright, and M. I. Jordan, "Local maxima in the likelihood of gaussian mixture models: Structural results and algorithmic consequences," in *Advances in Neural Information Processing Systems*, pp. 4116–4124, 2016.
- [14] D. P. Kingma and M. Welling, "Auto-encoding variational bayes," *arXiv preprint arXiv:1312.6114*, 2013.
- [15] I. Gulrajani, F. Ahmed, M. Arjovsky, V. Dumoulin, and A. C. Courville, "Improved training of wasserstein gans," in *Advances in Neural Information Processing Systems*, pp. 5769–5779, 2017.
- [16] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.
- [17] F. Zhu, Y. Wang, B. Fan, S. Xiang, G. Meng, and C. Pan, "Spectral unmixing via data-guided sparsity," *IEEE Transactions on Image Processing*, vol. 23, no. 12, pp. 5412–5427, 2014.
- [18] F. Zhu, Y. Wang, S. Xiang, B. Fan, and C. Pan, "Structured sparse method for hyperspectral unmixing," *ISPRS Journal of Photogrammetry and Remote Sensing*, vol. 88, pp. 101–118, 2014.
- [19] R. Heylen and P. Scheunders, "A multilinear mixing model for nonlinear spectral unmixing," *IEEE transactions on geoscience and remote sensing*, vol. 54, no. 1, pp. 240–251, 2016.
- [20] Y. Altmann, A. Halimi, N. Dobigeon, and J.-Y. Tourneret, "Supervised nonlinear spectral unmixing using a postnonlinear mixing model for hyperspectral imagery," *IEEE Transactions on Image Processing*, vol. 21, no. 6, pp. 3017–3025, 2012.

- [21] D. Stein, "Application of the normal compositional model to the analysis of hyperspectral imagery," in *Advances in techniques for analysis of remotely sensed data, 2003 IEEE Workshop on*, pp. 44–51, IEEE, 2003.
- [22] Y. Zhou, A. Rangarajan, and P. D. Gader, "A spatial compositional model for linear unmixing and endmember uncertainty estimation," *IEEE Transactions on Image Processing*, vol. 25, no. 12, pp. 5987–6002, 2016.
- [23] R. Heylen, D. Burazerovic, and P. Scheunders, "Non-linear spectral unmixing by geodesic simplex volume maximization," *IEEE Journal of Selected Topics in Signal Processing*, vol. 5, no. 3, pp. 534–542, 2011.

## A Appendix For Spectral Uncertainty

As explained, spectral images can be significantly affected by variations in atmospheric, illumination, or environmental conditions. Hence, spectral uncertainty ultimately influences the mixture of materials, especially for real data. Fig. 1 illustrates the pure material samples from real datasets. Observe that this uncertainty can be tedious for spectral data.

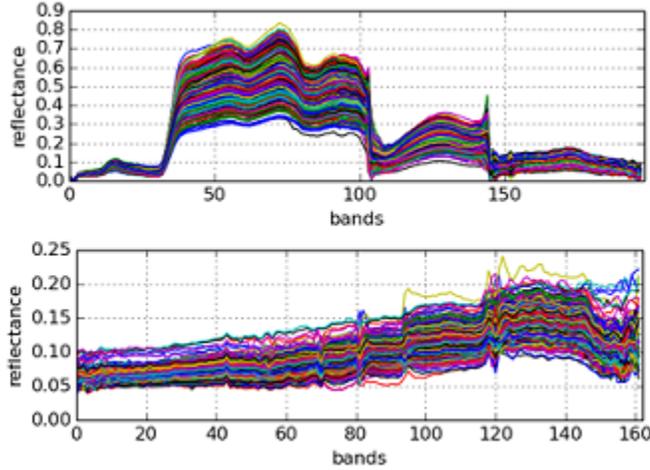


Figure 1: Spectral signatures from Jasper (first row) and Urban (second row) data correspond to two different pure materials.

## B Appendix For Model Flow

The overall flow of the proposed method is illustrated in Fig. 2. Our model takes spectral data  $x$  as input and computes a latent representation  $z$ . Later, this representation is modeled with a multinomial mixture model to estimate fractions  $y$ . These fractions are fed to two NN models with the matrix multiplication of precomputed endmembers in the decoder part. In the end, individual outputs are summed, and the input is reconstructed.

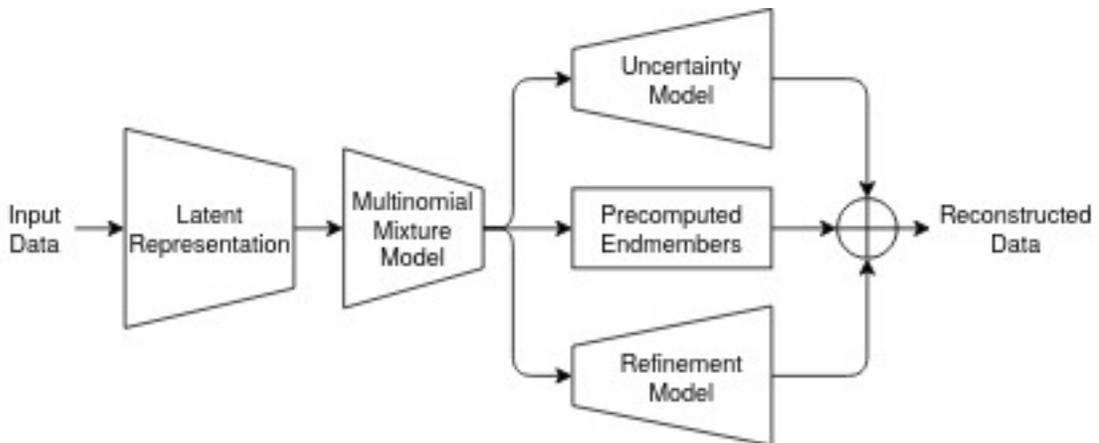


Figure 2: Flow of the proposed model.

## C Appendix For NN Structures

In our method, two deep NN models are primarily utilized. For the encoder part, a stack of 1D convolution layers is used to transform high-dimensional spectral data to a low-dimensional space, as illustrated in Tab. 3. Observe that the order of convolution and normalization layers is modified. This modification enhances the selectivity of representations by feeding more statistical knowledge to the convolution layers. Furthermore, Inception module that has multiple kernels of different sizes is also integrated. Hence, a rich representation from multiple kernel sizes can be computed.

Table 3: The NN Structure of the encoder part. The notation of Conv1D(W,K,S) indicates 1D convolution whose output dimension, kernel size and stride are W, K and S, respectively. Furthermore, Nrm and PReLU denote normalization and activation layers. AvgP(K) corresponds to the average pooling layer with kernel size and stride K.

Index	Index of Inputs	Operation(s)	Output Shape
(1)	-	Input Spectra	D
(2)	(1)	Conv1D(10, 21, 1)	$D \times 10$
(3)	(2)	PReLU + AvgP(5) + Nrm	$D/5 \times 10$
(4)	(3)	Conv1D(10, 3, 1)	$D/5 \times 10$
(5)	(3)	Conv1D(10, 5, 1)	$D/5 \times 10$
(6)	(3)	Conv1D(10, 7, 1)	$D/5 \times 10$
(7)	(4),(5),(6)	Concat	$D/5 \times 30$
(8)	(7)	PReLU + AvgP(2) + Nrm	$D/10 \times 30$
(9)	(8)	Conv1D(5, 3, 1)	$D/10 \times 10$
(10)	(9)	PReLU + AvgP(2) + Nrm	$D/20 \times 10$
(11)	(10)	Linear(M) + LReLU	M

To demonstrate the superiority of this model, Fig. 3 visualizes the distributions of input data and latent representation transformed by our method. Note that colors are discriminative based on their real fractions. Our method can blindly achieve more separable space for spectral data.

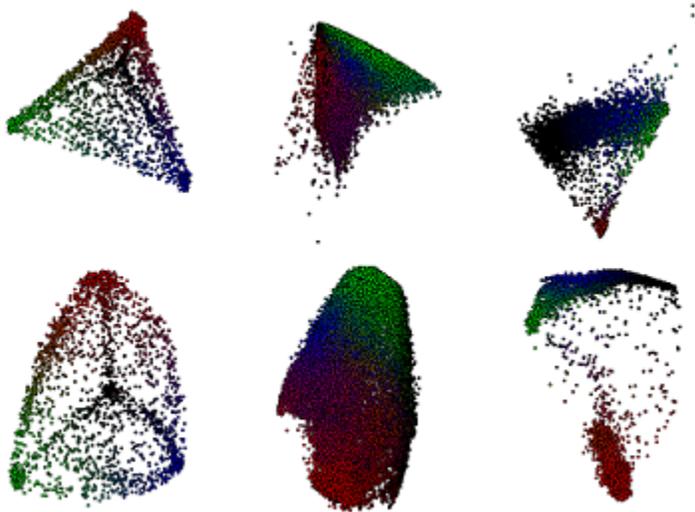


Figure 3: Data distributions before (first row) and after (second row) applying the encoder part.

Moreover, we utilize a deep NN model for the discriminative part of WGAN. This structure is also given in Tab. 4. In particular, PatchGAN is adopted in our method. Hence, local patches are fed to the discriminative model than full-dimensional data. Local characteristics (i.e., local spectral uncertainty) are practically captured.

Table 4: NN Structure of discriminative network.

Index	Index of Inputs	Operation(s)	Output Shape
(1)	-	Input Spectra	D
(2)	(1)	Conv1D(5, 21, 5)	$D/5 \times 5$
(3)	(2)	Nrm + PReLU	$D/5 \times 5$
(4)	(3)	Conv1D(10, 5, 2)	$D/10 \times 10$
(5)	(4)	Nrm + PReLU	$D/10 \times 10$
(6)	(5)	Conv1D(20, 5, 2)	$D/20 \times 20$
(7)	(6)	Nrm + PReLU	$D/20 \times 20$
(8)	(7)	Linear(5)	$D/20 \times 5$

## D Appendix For Additional Experiments on Synthetic Data

Our method is also tested on a synthetic dataset released for the experiments in (1). The dataset consists of four constituent materials from the ASTER spectral library: limestone, basalt, concrete and asphalt. For the scene, limestone is set as the background material; the rest is randomly placed to the image by maintaining the Gaussian-shaped distributions.

Even if synthetic data is conventional to evaluate the methods, it is not suitable to have all adverse cases exhibited on real data. Hence, it is hard to simulate nonlinearity and spectral uncertainty on the data. In particular, additive noise is inserted to simulate spectral uncertainty, which cannot reflect the actual physical conditions. Fig. 4 illustrates samples from two pure materials on this synthetic dataset. Unlike real data, these samples lead to corrupted signatures.

Experimental results are reported in Tab. 5. If we do not use EU and WGAN steps, the model achieves the best performance. This result makes sense since spectral uncertainty is simulated as random additive noise. Hence, WGAN loss eventually learns this noise characteristics instead of the real spectral uncertainty. This characteristic drastically reduces the performance and undermines the model. However, as explained, the actual capacity of these steps on real data is obvious.

We also compare our method with baselines. In Tab. 6, these results show that the proposed method performs better even if the full capacity of model is not used.

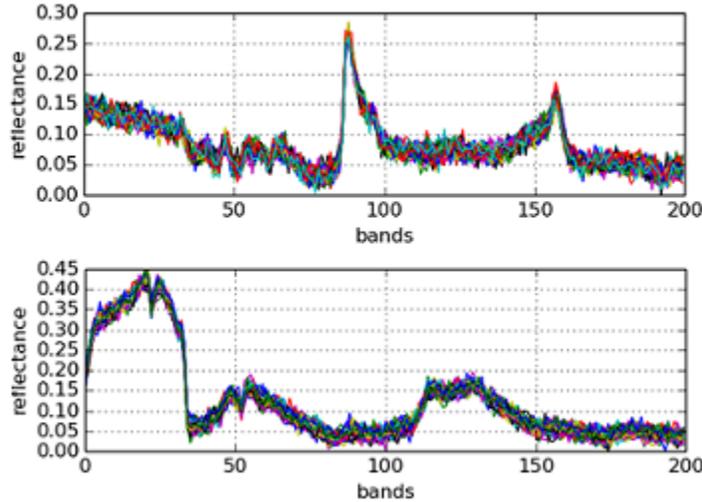


Figure 4: Spectra of pixels from the Synthetic dataset.

Table 5: RMSE performance on synthetic dataset. Best results are bold.

$\times 10^{-2}$	DMaxD	SCM	EndNet
N= 4	8.32 $\pm$ 1.4	4.47 $\pm$ 0.7	9.20 $\pm$ 1.4
N= 8	8.15 $\pm$ 1.4	4.36 $\pm$ 0.9	9.14 $\pm$ 1.0
N= 16	8.18 $\pm$ 1.0	3.97 $\pm$ 0.5	9.06 $\pm$ 1.1
N= 24	7.55 $\pm$ 1.1	4.42 $\pm$ 0.6	9.51 $\pm$ 1.2
w\o EU	5.43 $\pm$ 0.3	3.81 $\pm$ 0.9	<b>6.43</b> $\pm$ 0.4
w\o WGAN	<b>5.23</b> $\pm$ 0.4	<b>3.22</b> $\pm$ 1.2	8.88 $\pm$ 0.3

Table 6: RMSE performance on synthetic dataset. Best results are bold.

$\times 10^{-2}$	DMaxD	SCM	EndNet
LMM	5.17	3.36	17.62
GBM	5.62	3.26	17.39
PPNM	7.66	4.01	9.34
MLM	6.52	3.85	6.71
NCM	<b>4.09</b>	3.85	11.17
DSCN++	5.23	<b>3.22</b>	<b>6.43</b>