Model-Aware Deep Learning for the Clustering of Hyperspectral Images with Context Preservation

Xianlu Li*, Nicolas Nadisic*, Shaoguang Huang†, Nikos Deligiannis‡ and Aleksandra Pižurica*
*Department of Telecommunications and Information Processing, Ghent University, Belgium.
†School of Computer Science, China University of Geosciences, Wuhan, China.
‡Department Electronics and Informatics, Vrije Universiteit Brussel, Belgium.
imec, Kapeldreef 75, B-3001 Leuven, Belgium.

***Abstract*—Deep subspace clustering is an effective method for clustering high-dimensional data, and it provides state-of-the-art results in clustering hyperspectral images (HSI). However, these methods typically suffer from the size of the so-called self-expression matrix that increases quadratically in size with the image to be clustered. This can result in significant demands on computing power and storage capacity, making it challenging to apply these methods to large-scale data. Recently emerging Efficient Deep Embedded Subspace Clustering focuses on learning the basis of different subspaces which need much fewer parameters. Here, we extend and generalize this approach to account for local and non-local spatial context. We propose a structured model-aware deep subspace clustering network for hyperspectral images where the contextual information is captured in the appropriately defined loss functions. A self-supervised loss captures the local spatial structure, and the non-local structure is incorporated through a contrastive loss that encourages pixels with small feature distances to have the same prediction, while pixels with large feature distances have a distinct prediction. The experiments on real-world hyperspectral datasets demonstrate clear advantages over state-of-the-art methods for subspace clustering.**

***Index Terms*—subspace clustering, deep learning, hyperspectral images, structure information**

I. Introduction

Hyperspectral images (HSI) are typically captured using sensors that measure the reflectance of objects across hundreds of narrow and contiguous spectral bands. Therefore for every pixel an entire electromagnetic spectrum is recorded, which gives precise information about the materials present in the scene. This makes HSI widely used in remote sensing applications such as environmental monitoring [1], and agricultural planning [2]. The classification of HSI pixels into distinct categories is a crucial aspect in these domains. However, the labeling process for HSI data can be time-intensive and laborious, leading to the widespread use of clustering for these tasks. In the early stages of research, traditional clustering methods such as K-means [3] and Fuzzy C-Means (FCM) [4] were widely adopted for HSI data clustering due to their efficiency and simplicity. These methods measure the similarity between data points based on the distances between them. In HSI, the pixels are often high-dimensional and sparsely distributed in the ambient space. This sparsity and high dimensionality of the data can cause distance-based methods to encounter difficulties, as the distances between different pixels can become nearly equal.

Subspace clustering is a method of clustering data points that lie in different low-dimensional subspaces within high-dimensional data. It has been applied successful in HSI clustering applications [5]–[13]. The traditional subspace clustering calculate the self-expression matrix based on original data [5]–[9], [14], and usually solve the model by Alternating Direction Method of Multipliers (ADMM) [15]. As deep learning continues to advance, it has also been applied to subspace clustering tasks [10]. Unlike some other approaches that apply deep neural networks to conduct fully data-driven clustering, deep subspace clustering starts from a formulation of a relevant optimization problem. This type of approaches are commonly referred to as model-aware deep learning [16], [17]. Model-aware deep learning is most often associated with inverse imaging problems, like image restoration or compressed sensing. In the case of model-aware deep subspace clustering, deep neural networks are utilised to extract features of original data and calculate the self-expression matrix or subspace basis by gradient back propagation [10]–[12], [18].

Due to the complexity of the problem, the current deep subspace clustering approaches typically treat each data point separately, neglecting thus the spatial context. Several methods that integrate with local spatial constraints [9] and non-local structure [11], [12], demonstrated significant improvements in clustering results. However, these models are combined with self-expression optimization and spectral clustering, and cannot be trained end-to-end. Therefore, existing methods only apply structural information to the self-expression optimization stage, meaning that the resulting structural information can only train a portion of the clustering process. Compared to traditional subspace clustering methods, deep subspace clustering methods are better suited to handle nonlinear structures, making them a popular choice for HSI clustering. However, in self-expression-based model-aware deep subspace learning, the size of the similarity matrix increases quadratically by the size of the data set which limits its real-world application. Recently, the approach of [13] attempted to enhance clustering efficiency by reducing the size of the similarity matrix through the use of superpixels which group image pixels with similar characteristics into larger, more meaningful units. However, this approach resulted in a decrease in accuracy at the superpixel level. Differently, Efficient Deep Embedded Subspace Clustering (EDESC) [18] focuses on learning the

basis of subspace that span the subspace, but not a similarity matrix, which reduces a lot the number of parameters to learn. However, EDESC [18] neglects the spatial structure and feature structure of data, which are critical for HSI clustering tasks.

To obtain accurate HSI clustering results for large-scale HSI data, motivated by the approach in [18], we propose an end-to-end, model-aware deep subspace clustering method for HSI that integrates both local and non-local image data structures. Our contributions are summarized as follows:

- we introduce an enhanced self-supervised loss function with spatial constraints which considers the spatial correlation between pixels and their neighbors. This approach ensures that the clustering result preserves the local structure of the image, which is important for accurate clustering performance.
- we introduce a contrastive loss to take into account the non-local structure of the original data. This loss encourages the model to learn representations that reflect the underlying structure of the data in feature space, which helps to improve the accuracy of the clustering result.
- we propose an end-to-end trainable network for HSI clustering that is scalable to big data. We conduct experiments on 3 datasets which shows the superiority of our method.

The rest of this article is organized as follows: Section II introduces related work on subspace clustering and HSI clustering. Section III presents the proposed network. Section IV describes the results of our method. Finally, Section V concludes the paper.

II. Related work

A. Traditional subspace clustering

The traditional subspace clustering typically involves computing the self-expression matrix, which captures the representation relationships between data points, based on the raw data. The assumption is that each data point can be expressed as a linear combination of other data points in the same subspace. The problem is formulated as follows in subspace clustering:

$$\min_{\mathbf{Z}} \frac{\lambda}{2}||\mathbf{X}-\mathbf{X}\mathbf{Z}||_F^2 + ||\mathbf{Z}||_p \quad \text{s.t.} \quad \text{diag}(\mathbf{Z}) = 0, \tag{1}$$

where $\mathbf{X} \in \mathbb{R}^{D\times N}$ is the image data, N and D are respectively the number of image pixels and the dimension of every pixel. $\mathbf{Z}$ represents the self-expression matrix which captures the relationships between different data points, $\text{diag}(\mathbf{Z}) = 0$ prevents data expressed by itself and λ is used to balance the different terms. The choice of p leads to different types of subspace clustering methods[1]. By reducing the value of p from infinity to zero, the sparsity of $\mathbf{Z}$ will increase. In subspace clustering, the p is usually set to 1 or 2. Recently some work were proposed [9], [14], [19] that reduces the redundancy of the self-expression matrix, which improves the efficiency of this model. However, this model may still not be effective in handling linear non-separable data.

[1]The l_p-norm is defined as $||Z||_p \triangleq (\sum_{i,j}|Z_{ij}|^p)^{\frac{1}{p}}$.

B. Deep subspace clustering

These methods map raw data into a deep feature space and usually obtain a self-expression matrix through iterative updates of linear fully connected layers [10]. Typically, a CNN-based autoencoder is used to extract the deep feature of the raw data with a reconstruction loss to ensure that the feature contains sufficient information, as shown below:

$$L_{ae} = \frac{1}{2N}||\mathbf{X}_q - \hat{\mathbf{X}}_q||_F^2, \tag{2}$$

where $\mathbf{X}_q \in \mathbb{R}^{N\times u\times v\times c}$ is the input data, u and v are the width and height of the patch-size, c is the channel of image patch. $\hat{\mathbf{X}}_q$ is the reconstruction of $\mathbf{X}_q$. To obtain a high-quality self-expression matrix, similar to traditional subspace clustering, the self-expression loss is formulated as follows:

$$L_{se} = \lambda_1||\mathbf{H}-\mathbf{H}\mathbf{C}||_F^2 + \lambda_2||\mathbf{C}||_p, \quad \text{s.t.} \quad \text{diag}(\mathbf{C}) = 0. \tag{3}$$

Here, $\mathbf{H}$ is the deep representation of $\mathbf{X}_q$, and $\mathbf{C}$ is the self-expression parameter from the linear fully connected layers. λ_1 and λ_2 are parameters to balance different terms. Some works incorporate additional supervised learning modules [20], and structure preservation losses [11], [12] to improve performance, but these methods have high memory consumption and are not scalable to large datasets. Recently, the approach of [13] combined superpixel and contrastive learning to make it more efficient in processing large-scale images, but its accuracy is at the superpixel level rather than the pixel level. Differently, a recent method EDESC [18] focuses on learning a high-quality basis for each subspace, which reduces the number of training parameters. It comprises two main steps. The first step involves initializing the model, in which a CNN-based autoencoder feature extraction network is trained to obtain a deep representation of the network. K-means++ [21] and Singular Value Decomposition (SVD) are then applied to get the initial basis of each subspace. In the second step, the entire network is trained, and the prediction of the class label is obtained based on the projection value of the data point into each subspace, which is formulated as follows:

$$s_{ij} = \frac{||h_i^T\mathbf{B}^{(j)}||_F^2 + \eta d}{\sum_j(||h_i^T\mathbf{B}^{(j)}||_F^2 + \eta d)}. \tag{4}$$

Here, h_i is the deep representation of the i_{th} data point, $\mathbf{B}^{(j)}$ is the basis of the j_{th} subspace, η controls the smoothness, d is the dimension of subspace, and s_{ij} is the probability that the i_{th} data point belongs to the j_{th} subspace. To achieve network convergence, a self-supervising loss is applied as follows:

$$\tilde{s}_{ij} = \frac{s_{ij}^2/\sum_i s_{ij}}{\sum_j(s_{ij}^2/\sum_i s_{ij})}. \tag{5}$$

The refinement in (5) rescales the entries s_{ij} to give more importance to the high-probability assignments. The self-supervised loss is defined as follows:

$$L_{sp} = \text{KL}(\tilde{\mathbf{S}}||\mathbf{S}), \tag{6}$$

where $\mathbf{S}$ is the prediction matrix, $\tilde{s}_{ij}$ represents the element of the i_{th} row and j_{th} column of $\mathbf{S}$, and $\tilde{\mathbf{S}}$ is the refined $\mathbf{S}$ obtained using (5). KL is the Kullback–Leibler divergence defined as

$$\mathrm{KL}(\tilde{\mathbf{S}}||\mathbf{S}) = \sum_i \sum_j \tilde{S}_{ij} \log \frac{\tilde{S}_{ij}}{S_{ij}}.$$

Additionally, to obtain a good quality subspace allocation and data projection, the following constraints are used:

$$B_{cons} = \epsilon(\frac{1}{2}||\mathbf{B}^T\mathbf{B} \odot \mathbf{I} - \mathbf{I}||_F^2 + \frac{1}{2}||\mathbf{B}^T\mathbf{B} \odot \mathbf{O}||_F^2). \quad (7)$$

Here, $\odot$ represents the Hadamard product, $\mathbf{I}$ is the identity matrix of size kd by kd, and $\mathbf{O}$ is a matrix in which all diagonal block elements of size d are 0 while the others are 1. The first constraint requires normalizing each basis to have a length of 1 to maintain the quality of the projection. The second constraint mandates that each subspace must have distinct bases to ensure optimal subspace allocation, and the value of ϵ is 10^{-3}. Inspired by EDESC [18], this article introduces a Structured Efficient Subspace Clustering Network for Hyperspectral Images. Our proposed approach leverages both local and non-local structures of the hyperspectral image to supervise the entire clustering process and improve the clustering results in large-scale HSI images.

III. PROPOSED METHOD

We build on the model EDESC [18], in the sense that we use a CNN-Based autoencoder to learn the deep feature of original data with the loss defined in (2) and the basis constraint of (7) to maintain the quality of basis. Instead of using a refined prediction to supervise the training of the basis learning module, we propose an enhanced self-supervised loss that considers the spatial correlation of image data based on both refined prediction and smoothed refined prediction. In order to maintain the non-local structure of data, we introduce a contrastive loss, which encourages data that have similar spectral features to have the same prediction. In Figure 1, it can be observed that two sets of image patches were generated from the original HSI block. The image patches with a smaller patch size were used for feature extraction and clustering, while the image patches with a larger patch size were used to capture the non-local structure of the data. The network is composed of a CNN-based autoencoder and a basis layer. The autoencoder is utilized to study the features that contain sufficient information from the original data. On the other hand, the basis layer is utilized to learn a good basis for subspace clustering.

A. The local structure preserving module

Based on the analysis of the image properties, each pixel is likely to have the same class label as its neighboring pixels. However, most previous subspace clustering methods, including EDESC [18], cluster data solely based on their own characteristics without taking into account the prediction of neighboring data points. Based on this observation, a filter-based post-processing technique is applied. During this

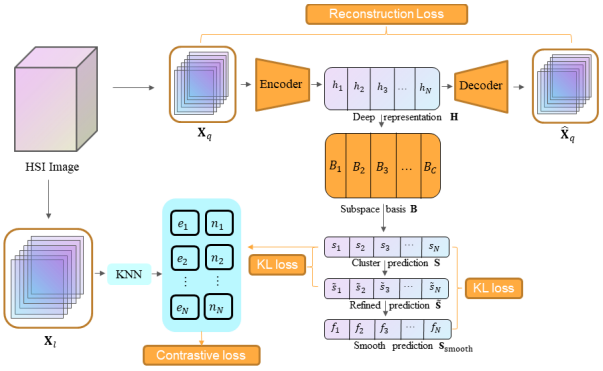


Fig. 1. The architecture of our clustering network. The deep representations $\mathbf{H}$ learned from the image patches $\mathbf{X}_q$ are projected through the subspace basis $\mathbf{B}$ to produce the prediction $\mathbf{S}$. During optimization, the KL divergence loss is minimized between $\mathbf{S}$ and its refined predictions $\tilde{\mathbf{S}}$ as well as the smooth prediction $\mathbf{S}_{smooth}$ to preserve the local structure. The non-local structures captured from $\mathbf{X}_l$ are preserved by minimizing the contrastive loss.

post-processing, the refined result will be assigned to the corresponding location of the original image. Meanwhile, a mask will assign 1 to the same location. We will mean-filter the refined prediction and then divide it by the mean-filtered mask to remove the effect of the irrelevant area. This processing results in a smoothed refined prediction $\mathbf{S}_{smooth}$. We extend the self-supervised loss from (6) to incorporate this smoothed prediction. In particular, we define a local structure preserving loss function L_{sp} as follows:

$$L_{sp} = \lambda_{sp}\,\mathrm{KL}(\tilde{\mathbf{S}}||\mathbf{S}) + (1-\lambda_{sp})\,\mathrm{KL}(\mathbf{S}_{smooth}||\mathbf{S}), \quad (8)$$

where λ_{sp} is the parameter used to control the smoothness of the refined prediction.

B. The non-local structure preserving module

We propose an original way to incorporate the non-local structure in the clustering process in an end-to-end manner, inspired by the successful application of contrastive learning in subspace clustering tasks [13], [22], [23]. Differently, we apply this loss directly in the clustering result, allowing for joint optimization of the feature extraction network and basis learning layers. Firstly, positive and negative samples are selected by K-Nearest Neighbors (KNN) based on the feature distance of the original data. The e_i and n_i are the positive sample sets and negative sample sets of i_{th} data points. After obtaining the samples, the loss is minimized as follows to optimize the entire network:

$$L_{con} = \frac{1}{N}\sum_{i=1}^{N} -\log\left(\frac{\sum_{j\in e_i}\exp\left(g_{ij}/\tau\right)}{\sum_{t\in(e_i\cup n_i)}\exp\left(g_{it}/\tau\right)}\right), \quad (9)$$

where τ is the temperature parameter, g_{ij} is the similarity of the prediction between i_{th} and j_{th} data points defines as follows:

$$g_{ij} = \hat{s}_i \cdot \hat{s}_j, \quad (10)$$

$\hat{s}_i$ is the normalized prediction of i_{th} data point. By minimizing this loss, the pixels with small feature distances

are encouraged to have the same cluster prediction which maintains the non-local structure of data.

C. Objective function and training strategy

Our model's training process consists of three steps, which can be summarized in pseudo-code in Algorithm 1.

Algorithm 1 Training Strategy

Step 1: Pre-train the feature extraction network and initialize the basis; $Loss = L_{ae}$;
Step 2: Train the whole network with the local structure constraint with weight β; $Loss = L_{ae} + B_{cons} + \beta\ L_{sp}$;
Step 3: Train the whole network with the local and non-local structure constraint with weight β and λ_{con} respectively; $Loss = L_{ae} + B_{cons} + \beta\ L_{sp} + \lambda_{con}\ L_{con}$;

IV. Experimental Evaluation

A. Datasets and experiment settings

We conduct our experiment on three hyperspectral image datasets: Trento, Houston, and PaviaU. Trento had 63 spectral bands and 6 classes, with an image size of 600×166 and 30,214 samples. Houston had 144 bands, 7 classes, and a size of 130×130 with 6,104 samples. PaviaU had 115 bands, 9 classes, and a size of 610×340 with 42,776 samples.

We implement our model based on a CNN-based autoencoder. We provide our code and test scripts in an online repository[2]. Our model requires different hyperparameters for each image since each image has a unique local and non-local structure and feature, which can significantly impact the performance of the model if not optimized properly. According to the approach proposed in [18], the value of β ranges from 0 to 1. The λ_{sp} parameter also ranges from 0 to 1, where a value of 0 implies that the model does not consider local structure, while a value of 1 indicates that the model takes full smoothness into account. Similarly, λ_{con} ranges from 0 to 1 and its specific value may vary depending on the dataset. In our experiments, we set β to 1 for the Trento and Houston datasets, and to 0.2 for the PaviaU dataset. The λ_{sp} was set to 0 for Trento and Houston, and to 0.5 for PaviaU, while λ_{con} was set to 1 for Trento and Houston, and to 0.7 for PaviaU.

B. Comparison with state-of-art methods

In Table I, we present clustering results for different representative methods applied to the Houston, Trento, and PaviaU datasets, respectively. We also showed the clustering map of Trento dataset in Figure 2. Based on our findings, distance-based methods like Kmeans++ [21], and FCM [4] are scalable for large datasets but have lower accuracy compared to pixel-level deep subspace clustering approaches such as HyperAE. While the superpixel-based method can improve the efficiency of self-expression-based methods, it may decrease accuracy. The hierarchical-based clustering techniques like [24] can achieve an efficient and relatively good result. DEC [25] can achieve good performance for the Houston dataset but may not for others and also requires relatively high memory resources. Compared to these methods, our approach is scalable for large-scale data and can achieve superior accuracy than many existing methods.

[2]https://github.com/lxlscut/SEDESC

C. Ablation study

In this section, we investigate the influence of the proposed loss function on clustering performance. We conduct experiments with three models, each using a different combination of loss functions, and compare their results to the base model built on EDESC. The results of our ablation study are presented in Table II, which indicates that incorporating both the local and non-local structure modules leads to improved clustering performance.

V. Conclusion

In this paper we presented an efficient deep subspace clustering model for HSI clustering tasks. Specifically, we proposed an enhanced supervised loss function that considers the local structure information of the original data. Additionally, in order to maintain the non-local structure of the HSI, we proposed a contrastive loss that encourages pixels with small feature distances to have the same prediction value. The ablation experiment result shows that both local and non-local modules can effectively improve the clustering accuracy. Extensive experiments on three real-world datasets show that our method outperforms the state-of-the-art methods.

References

[1] M. B. Stuart, A. J. McGonigle, and J. R. Willmott, "Hyperspectral imaging in environmental monitoring: A review of recent developments and technological advances in compact field deployable systems," *Sensors*, vol. 19, no. 14, p. 3071, 2019.

[2] B. Lu, P. D. Dao, J. Liu, Y. He, and J. Shang, "Recent advances of hyperspectral imaging technology and applications in agriculture," *Remote Sensing*, vol. 12, no. 16, p. 2659, 2020.

[3] J. A. Hartigan and M. A. Wong, "Algorithm as 136: A k-means clustering algorithm," *Journal of the Royal Statistical Society. series C (applied statistics)*, vol. 28, no. 1, pp. 100–108, 1979.

[4] J. C. Bezdek, *Pattern recognition with fuzzy objective function algorithms*. Springer Science & Business Media, 2013.

[5] E. Elhamifar and R. Vidal, "Sparse subspace clustering: Algorithm, theory, and applications," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 35, no. 11, pp. 2765–2781, 2013.

[6] H. Zhang, H. Zhai, L. Zhang, and P. Li, "Spectral–spatial sparse subspace clustering for hyperspectral remote sensing images," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 54, no. 6, pp. 3672–3684, 2016.

[7] H. Zhai, H. Zhang, L. Zhang, P. Li, and A. Plaza, "A new sparse subspace clustering algorithm for hyperspectral remote sensing imagery," *IEEE Geoscience and Remote Sensing Letters*, vol. 14, no. 1, pp. 43–47, 2016.

[8] S. Huang, H. Zhang, and A. Pižurica, "Joint sparsity based sparse subspace clustering for hyperspectral images," in *2018 25th IEEE International Conference on Image Processing (ICIP)*. IEEE, 2018, pp. 3878–3882.

[9] S. Huang, H. Zhang, Q. Du, and A. Pižurica, "Sketch-based subspace clustering of hyperspectral images," *Remote Sensing*, vol. 12, no. 5, p. 775, 2020.

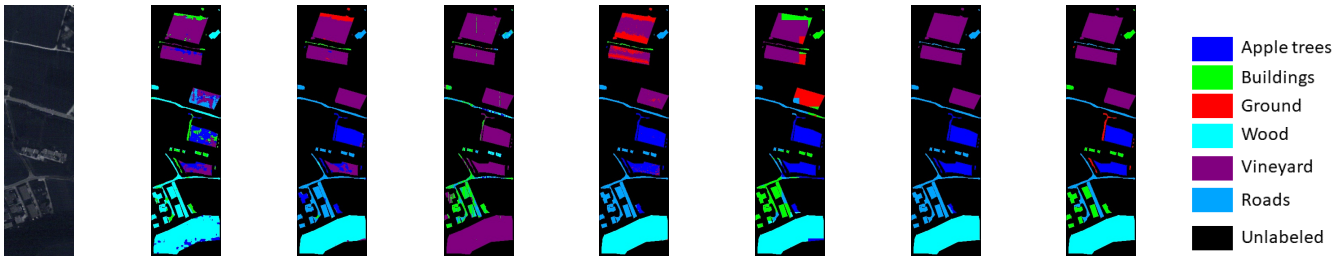
[10] P. Ji, T. Zhang, H. Li, M. Salzmann, and I. Reid, "Deep subspace clustering networks," *Advances in Neural Information Processing Systems*, vol. 30, 2017.

[11] Y. Cai, Z. Zhang, Z. Cai, X. Liu, and X. Jiang, "Hypergraph-structured autoencoder for unsupervised and semisupervised classification of hyperspectral image," *IEEE Geoscience and Remote Sensing Letters*, vol. 19, pp. 1–5, 2021.

TABLE I
RESULT OF EXPERIMENTS ON CLUSTERING HYPERSPECTRAL IMAGES

Dataset	Metrics	Algorithms						
		FCM [4]	Kmeans++ [21]	DEC [25]	Finch [24]	HyperAE [11]	NCSC [13]	Ours
Houston	ACC	0.6633	0.6450	0.7244	0.7210	0.7135	0.6114	**0.7266**
	NMI	0.6204	0.6973	**0.7726**	0.7702	0.7700	0.6371	0.7681
	Kappa	0.5531	0.5354	0.6463	0.6424	0.6328	0.5361	**0.6524**
Trento	ACC	0.6413	0.8183	0.5546	0.7622	OOM	0.7415	**0.8846**
	NMI	0.5900	0.7717	0.5600	0.8200	OOM	0.6784	**0.9136**
	Kappa	0.5049	0.7566	0.3730	0.6963	OOM	0.6718	**0.8455**
PaviaU	ACC	0.5659	0.5097	OOM	0.5590	OOM	OOM	**0.6549**
	NMI	0.4626	0.5926	OOM	0.6015	OOM	OOM	**0.6382**
	Kappa	0.4257	0.3919	OOM	0.4454	OOM	OOM	**0.5471**

Note: Due to an out-of-memory (OOM) problem, some methods can not run in some large-scale datasets. The best results are marked in **bold**.



(a) (b) OA=64.63% (c) OA=80.30% (d) OA=48.50% (e) OA=76.22% (f) OA=75.92% (g) OA=88.80% (h) GT

Fig. 2. Clustering maps for Trento datasets. (a) False color image. (b) FCM. (c) Kmeans++. (d) DEC. (e) Finch. (f) Ncsc. (g) Ours. (h) Ground Truth.

TABLE II
RESULTS OF ABLATION STUDY

Method	Trento			Houston			PaviaU		
	ACC	NMI	Kappa	ACC	NMI	Kappa	ACC	NMI	Kappa
Baseline	0.8825	0.9077	0.8430	0.7178	0.7629	0.6411	0.6190	0.5973	0.5053
w/s	0.8843	0.9113	0.8451	0.7247	0.7660	0.6497	0.6212	0.6008	0.5080
$w/s\&f$	**0.8846**	**0.9136**	**0.8455**	**0.7266**	**0.7681**	**0.6524**	**0.6549**	**0.6382**	**0.5471**

Note: Baseline refers to EDESC, w/s refers to EDESC with local structure, and $w/s\&f$ refers to EDESC with both local and non-local structure.

[12] M. Zeng, Y. Cai, X. Liu, Z. Cai, and X. Li, "Spectral-spatial clustering of hyperspectral image based on laplacian regularized deep subspace clustering," in *IGARSS 2019-2019 IEEE International Geoscience and Remote Sensing Symposium*. IEEE, 2019, pp. 2694–2697.

[13] Y. Cai, Z. Zhang, P. Ghamisi, Y. Ding, X. Liu, Z. Cai, and R. Gloaguen, "Superpixel contracted neighborhood contrastive subspace clustering network for hyperspectral images," *IEEE Transactions on Geoscience and Remote Sensing*, 2022.

[14] S. Huang, H. Zhang, and A. Pižurica, "Subspace clustering for hyperspectral images via dictionary learning with adaptive regularization," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 60, pp. 1–17, 2021.

[15] S. Boyd, N. Parikh, E. Chu, B. Peleato, J. Eckstein *et al.*, "Distributed optimization and statistical learning via the alternating direction method of multipliers," *Foundations and Trends® in Machine learning*, vol. 3, no. 1, pp. 1–122, 2011.

[16] V. Monga, Y. Li, and Y. C. Eldar, "Algorithm unrolling: Interpretable, efficient deep learning for signal and image processing," *IEEE Signal Processing Magazine*, vol. 38, no. 2, pp. 18–44, 2021.

[17] N. Shlezinger, J. Whang, Y. C. Eldar, and A. G. Dimakis, "Model-based deep learning," *Proceedings of the IEEE, year=2023*.

[18] J. Cai, J. Fan, W. Guo, S. Wang, Y. Zhang, and Z. Zhang, "Efficient deep embedded subspace clustering," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 1–10.

[19] P. A. Traganitis and G. B. Giannakis, "Sketched subspace clustering," *IEEE Transactions on Signal Processing*, vol. 66, no. 7, pp. 1663–1675, 2017.

[20] K. Li, Y. Qin, Q. Ling, Y. Wang, Z. Lin, and W. An, "Self-supervised deep subspace clustering for hyperspectral images with adaptive self-expressive coefficient matrix initialization," *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, vol. 14, pp. 3215–3227, 2021.

[21] D. Arthur and S. Vassilvitskii, "k-means++: The advantages of careful seeding," Stanford, Tech. Rep., 2006.

[22] X. Hu, T. Li, T. Zhou, and Y. Peng, "Deep spatial-spectral subspace clustering for hyperspectral images based on contrastive learning," *Remote Sensing*, vol. 13, no. 21, p. 4418, 2021.

[23] B. Peng and W. Zhu, "Deep structural contrastive subspace clustering," in *Asian Conference on Machine Learning*. PMLR, 2021, pp. 1145–1160.

[24] S. Sarfraz, V. Sharma, and R. Stiefelhagen, "Efficient parameter-free clustering using first neighbor relations," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2019, pp. 8934–8943.

[25] J. Xie, R. Girshick, and A. Farhadi, "Unsupervised deep embedding for clustering analysis," in *International conference on machine learning*. PMLR, 2016, pp. 478–487.