remote sensing

MDPI

Article

Scalable Context-Preserving Model-Aware Deep Clustering for Hyperspectral Images

Xianlu Li [1,*], Nicolas Nadisic [1,2], Shaoguang Huang [3], Nikos Deligiannis [4,5] and Aleksandra Pižurica [1]

[1] Department of Telecommunications and Information Processing, Ghent University, Sint-Pietersnieuwstraat 41, 9000 Ghent, Belgium
[2] Royal Institute for Cultural Heritage (KIK-IRPA), Jubelpark 1, 1000 Brussels, Belgium
[3] School of Computer Science, China University of Geosciences, Wuhan 430074, China
[4] ETRO Department, Vrije Universiteit Brussel (VUB), Pleinlaan 2, 1050 Brussels, Belgium; ndeligia@etrovub.be
[5] IMEC, Kapeldreef 75, 3001 Leuven, Belgium
* Correspondence: xianlu.li@ugent.be

Highlights

What are the main findings?

- A mini-cluster-based optimization scheme is proposed to preserve the non-local structure of hyperspectral image (HSI) data.
- A one-stage, end-to-end deep clustering network is designed to learn subspace bases under the joint guidance of local and non-local structures.

What is the implication of the main finding?

- The mini-cluster optimization scheme adaptively models non-local similarity with higher efficiency than manifold-based methods relying on fixed neighbor settings.
- The end-to-end framework enables local and non-local structures to jointly supervise and optimize the entire clustering process, overcoming the limitations of previous two-stage deep subspace methods.

Abstract

Subspace clustering has become widely adopted for the unsupervised analysis of hyperspectral images (HSIs). Recent model-aware deep subspace clustering methods often use a two-stage framework, involving the calculation of a self-representation matrix with complexity of $\mathcal{O}(n^2)$, followed by spectral clustering. However, these methods are computationally intensive, generally incorporating only local or non-local structure constraints, and their structural constraints fall short of effectively supervising the entire clustering process. We propose a scalable, context-preserving deep clustering method based on basis representation, which jointly captures local and non-local structures for efficient HSI clustering. To preserve local structure—i.e., spatial continuity within subspaces—we introduce a spatial smoothness constraint that aligns clustering predictions with their spatially filtered versions. For non-local structure—i.e., spectral continuity—we employ a mini-cluster-based scheme that refines predictions at the group level, encouraging spectrally similar pixels to belong to the same subspace. These two constraints are jointly optimized to reinforce each other. Specifically, our model is designed as a one-stage approach, in which the structural constraints are applied to the entire clustering process. The time and space complexity of our method are $\mathcal{O}(n)$, making it applicable to large-scale HSI data. Experiments on real-world datasets show that our method outperforms state-of-the-art techniques.

Keywords: hyperspectral images; model-aware deep learning; self-representation; basis-representation; structure preservation

check for updates

Academic Editor: Zhenwei Shi

Received: 9 October 2025
Revised: 5 December 2025
Accepted: 12 December 2025
Published: 14 December 2025

Citation: Li, X.; Nadisic, N.; Huang, S.; Deligiannis, N.; Pižurica, A. Scalable Context-Preserving Model-Aware Deep Clustering for Hyperspectral Images. *Remote Sens.* **2025**, *17*, 4030. https://doi.org/10.3390/rs17244030

Copyright: © 2025 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (https://creativecommons.org/licenses/by/4.0/).

1. Introduction

Hyperspectral images (HSIs) record the precise electromagnetic spectrum of the objects in a scene in hundreds of spectral bands, thereby enabling discrimination between objects that are indistinguishable in conventional Red–Green–Blue (RGB) images. As a result, HSIs have been widely applied in fields such as agriculture [1], environmental monitoring [2], and defense and security [3]. Clustering, which categorizes image pixels into different classes without labeled data, plays a crucial role in interpreting HSI data. However, HSI clustering remains challenging due to noise and high spectral variability [4]. Subspace clustering [5,6], which models data as lying in a union of low-dimensional subspaces, has shown strong performance in this context and has gained significant attention in recent years. Subspace clustering assumes that high-dimensional data lie in a few low-dimensional subspaces, where data points within the same subspace are treated as a class. Representative subspace clustering methods can be categorized into model-based subspace clustering [5,7–9] and model-aware deep subspace clustering [10–12]. The model-based subspace clustering methods include sparse subspace clustering (SSC) [5], low-rank representation (LRR) [7], and the joint-sparsity-based sparse subspace clustering (JSSC) method [13]. These approaches are designed based on the self-representation property, which assumes that each data point in a subspace $\mathcal{S}_i$ can be expressed as a linear combination of other points within the same subspace, subject to sparsity or low-rank constraints. The resulting representation matrix effectively reveals the affinities between different data points. It is thus used to construct a similarity matrix, which is further fed to spectral clustering to obtain the clustering result. However, these methods are limited by the matrix-decomposition-based shallow representations, making it difficult to cluster HSIs that are often nonlinearly separable in practice.

To solve this problem, model-aware deep subspace clustering methods leverage the feature extraction capacity of deep neural networks to extract discriminative features and take into account nonlinear interactions. Representative methods include a deep subspace clustering network (DSCNet) [10], a generic deep subspace clustering model (DSC) [11], a self-supervised variant with adaptive initialization (SDSC-AI) [14], a Laplacian-regularized model for hyperspectral images (LRDSC) [15], and a pseudo-supervised extension (PSSC) [12]. Typically, autoencoders are used to project the input data onto a latent feature space, and then a fully connected layer is incorporated within the latent space, positioned between the encoder and decoder, to approximate the self-representation model. Li et al. [14] use clustering results as pseudo labels to train the feature extraction network, enhancing feature discriminability for clustering tasks. They also initialize the self-representation layer with a k-nearest neighbors (KNNs) graph to reduce dictionary redundancy, leading to significant performance improvements. In [16], features from undercomplete and overcomplete autoencoders are fused for subspace clustering, achieving outstanding performance without pre-training. Chen et al. [17] propose leveraging self-attention within the autoencoder to capture long-range dependencies, yielding better results than DSCNet. Benefiting from the improved feature representation in the latent space with the encoders, model-aware deep subspace clustering methods are more effective in handling data of complex structures compared with the aforementioned model-based subspace clustering methods. By optimizing the parameters of the fully connected layer, the self-representation matrix can be obtained for the construction of the similarity matrix. However, they still suffer from the following issues. First, these methods are computationally expensive. This is because the self-representation matrix is of size $n \times n$ (where n is the number of HSI pixels), leading to a training complexity of $\mathcal{O}(n^2)$ that makes large-scale clustering impractical; in addition, the spatial constraints employed further increase the computational burden. Second, the features extracted may not be optimal for clustering. This is because feature extraction and spectral clustering are performed separately, which risks degrading overall performance. Third, these methods struggle to capture the intrinsic cluster structure of HSI. This limitation arises from their focus on either local or non-local dependencies, rather than fully exploiting the spatial relationships present in the data.

In this paper, we propose a scalable context-preserving deep subspace clustering (SCDSC) method, which performs feature extraction and clustering within a unified framework. In contrast to conventional self-representation-based clustering methods that require optimizing a large self-representation matrix, we follow the approach proposed in [18] and instead learn compact subspace bases. Those bases are compact, class-specific subspace dictionaries with fewer parameters in the latent feature space. We then obtain the clustering soft assignment directly by projecting the latent representation onto the subspace bases. The resulting model has low computational complexity, supporting scalability and efficient processing of large HSI.

To capture both local and non-local dependencies in hyperspectral data, we introduce two structural constraints. The local structure constraint enhances spatial homogeneity in the clustering results and improves robustness to noise and spectral variability. We achieve this using a spatial-wise mean filter to smooth the clustering results. The non-local structure constraint promotes consistency among spectrally similar data points, regardless of their spatial distance, by grouping them into mini-clusters and encouraging shared cluster assignments within each group. In contrast to existing local spatial constraints like total variation, which exhibit high computational complexity, our method is computationally efficient. The improved local homogeneity can be propagated to non-local data points through the non-local constraint, and conversely, the non-local constraint can also enhance local homogeneity, creating a mutually reinforcing relationship. To the best of our knowledge, this is the first attempt to develop an end-to-end, scalable deep subspace clustering method for HSIs. Experimental results on four benchmark datasets show that our method consistently outperforms several state-of-the-art methods, both model-based and deep learning-based. A preliminary version of this work was presented in [19], where we applied spatial filtering to embed spatial continuity into the soft assignment optimization and used contrastive learning in the feature space to preserve the non-local structure. Compared with that preliminary work, we develop a novel approach to modeling non-local similarities by means of a mini-cluster grouping. Moreover, we provide a more detailed presentation, a deeper analysis of the overall approach, and critical discussions. We also present a more extensive experimental study.

The remainder of the paper is as follows: Section 2 provides a comprehensive analysis of model-based subspace clustering and deep clustering methods, including purely data-driven and model-aware approaches. Section 3 describes our main contribution, a context-preserving deep subspace clustering method. Section 4 evaluates it using four real-world hyperspectral datasets. Finally, Section 5 concludes this paper.

2. Related Work

In this section, we introduce the key concepts and models that form the foundation of our proposed method. Then, we briefly review the existing approaches for HSI clustering.

2.1. Agglomerative Hierarchical Clustering

In our work, we use agglomerative hierarchical clustering to generate mini-clusters, as detailed in Section 3. Agglomerative hierarchical clustering is a method that groups data points by gradually merging clusters. It starts with each data point as its own separate cluster, and then at each iteration, it merges clusters based on a defined rule, called a linkage criterion. As a variation of this method, a first-integer-neighbor clustering method (FINCH) [20] uses the first neighbor of each sample to identify long neighbor chains and uncover groups within the data. This method shows high performance in clustering complex data with a complexity of $\mathcal{O}(n \log n)$, where n is the number of data points. In every iteration of FINCH, the first nearest neighbor adjacency matrix is created as follows:

$$\mathbf{A}(i,j) = \begin{cases} 1 & \text{if } j = \kappa_i^1 \text{ or } \kappa_j^1 = i \text{ or } \kappa_i^1 = \kappa_j^1 \\ 0 & \text{otherwise} \end{cases} \tag{1}$$

where κ_i^1 denotes the first nearest neighbor of point i. During the merging process, data points connected by the same neighbor chain are grouped, and a new data point is generated as the mean of these points. The generated data points are merged in the same way in the next iterations. In the initial few iterations, the relationships between data points are simpler, allowing the algorithm to merge data points accurately. Compared to methods that require a predefined number of neighbors, FINCH is a parameter-free algorithm, and it allows clusters of different sizes. This results in a more flexible and adaptive capture of non-local data structures.

2.2. Model-Based Subspace Clustering

Subspace clustering has become a major approach in analyzing high-dimensional data because it efficiently identifies meaningful low-dimensional structures within high-dimensional data. Classical model-based subspace clustering methods such as SSC [5] and LRR [7] optimize self-representation-based models to reveal the affinities between different data points, as shown in Figure 1.

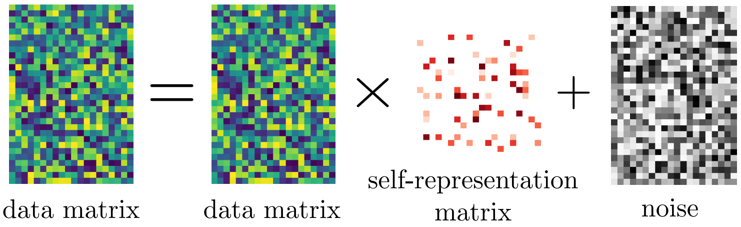


Figure 1. An illustration of the self-representation model.

However, they rely solely on spectral representations to identify relationships between HSI pixels, making the models sensitive to noise. Therefore, many extensions attempt to incorporate local and non-local information to improve robustness against noise.

Zhang et al. [21] propose a spectral–spatial sparse subspace clustering method (S4C) that incorporates spatial information by applying a 2D mean filter to the representation matrix. Huang et al. [22] impose a joint constraint on local neighborhoods obtained via superpixel segmentation to reduce feature variability within clusters. Other methods, including a locally constrained collaborative representation-based Fisher's LDA (LCR-FLDA) [23], incorporate non-local structure by imposing Laplacian constraints. An alternative to traditional self-representation-based models is dictionary-based approaches, which are generally more computationally efficient. Representative methods include a sketch-based subspace clustering method with total variation (Sketch-TV) [24], a dictionary learning method with adaptive regularization (IDLSC) [25], a total variation regularized collaborative representation clustering method with a locally adaptive dictionary (TV-CRC-LAD) [26], and a structural prior-guided subspace clustering method (SPGSC) [27]. Although these methods achieve notable improvements over traditional subspace clustering approaches for HSI, their underlying assumptions still pose significant limitations. Specifically, they rely on the premise that each data point can be represented as a linear combination of points from a single linear subspace or a given dictionary. This linearity assumption does not hold when the data lie on nonlinear manifolds whose intrinsic geometry cannot be approximated using relatively few global linear subspaces, leading to degraded performance.

2.3. Purely Data-Driven Deep Clustering Methods

Data-driven deep clustering approaches focus on learning from the inherent structure and distribution of data and can be broadly categorized into two types. The first type involves feature learning followed by traditional clustering, where neural networks are used for feature extraction, after which conventional clustering algorithms are applied. For example, the deep embedding network for clustering DEN [28] employs an autoencoder with local similarity and sparsity constraints, followed by K-means clustering on the extracted features. Similarly, deep subspace clustering with sparsity prior (PARTY) [29] trains an autoencoder with structure-prior regularization and then applies subspace clustering. The deep spectral clustering method SpectralNet [30] first trains a neural network to learn an embedding that approximates the eigenvectors of the graph Laplacian and then applies K-means clustering in the learned embedding space. In a similar manner, the manifold-based deep clustering method N2D [31] begins with autoencoder-based feature learning, then performs manifold embedding, and finally applies a traditional clustering algorithm to obtain cluster labels. The second type is the joint optimization of feature learning and clustering, where traditional clustering algorithms are integrated into the neural network's loss function, allowing for simultaneous clustering and feature learning during network training. For instance, the unsupervised deep embedding method for clustering (DEC) [32] learns a latent representation with an autoencoder and refines cluster assignments using a soft assignment based on the Student's t-distribution together with a Kullback–Leibler (KL) divergence clustering objective. Building on this, Nalepa et al. [33] employ a three-dimensional (3D) convolutional network to better capture spectral structure, further improving performance. Another method, a deep semantic clustering method based on partition confidence maximization (PICA) [34], maximizes the confidence level of each data point being assigned to a cluster, thereby enhancing clustering results. While purely data-driven deep clustering methods are flexible and can adapt to data with complex structures or noise, they often require large datasets, lack interpretability, and are prone to overfitting.

2.4. Model-Aware Deep Subspace Clustering

Model-aware deep learning, which integrates mathematical modeling with deep neural networks to harness the advantages of both domains, has been widely adopted in various image inverse problems, including image denoising [35], image reconstruction [36], and compressed sensing [37]. In remote sensing, this paradigm has been extensively explored through deep unfolding and plug-and-play (PnP) frameworks. For example, deep unfolding has been applied to tasks such as satellite image super-resolution [38] and pan-sharpening [39], where iterative optimization algorithms are unrolled into deep networks, offering both interpretability and improved performance. In contrast, PnP strategies embed learned priors into traditional optimization pipelines, as demonstrated in hyperspectral unmixing [40]. Similarly, model-aware deep learning techniques have shown success in subspace clustering. A pioneering work is DSCNet [10], which employs a deep autoencoder to map data nonlinearly into a latent space and then applies a fully connected layer on the latent representation to mimic the self-representation model.

To improve robustness to noise, many extensions have been proposed that incorporate non-local structure during model optimization. For example, Zeng et al. [15] employ a Laplacian regularizer on the self-representation matrix to directly impose non-local structure preservation. In [41], the Laplacian regularizer is applied to the self-representation matrix within a residual network. In [42], the hypergraph-structured autoencoder (HyperAE) imposes a hypergraph regularizer on the latent representation, thereby maintaining the non-local structure in the self-representation matrix. In summary, although these methods perform well, they have several limitations. They require significant computational resources due to the large matrices involved; their two-stage design prevents integrated structure preservation, and they do not fully capture the spatial dependencies present in the data.

2.5. Summary and Discussion

In Table 1, we summarize the above-described categorization of HSI clustering approaches, with representatives of model-based and deep learning-based methods, along with their main advantages and limitations. While these methods have achieved remarkable progress in hyperspectral image clustering, they typically incur high computational costs, rely on either linear assumptions or large amounts of data, and often lack the structural priors required to capture the nonlinear spectral–spatial patterns of hyperspectral images.

Table 1. Summary of representative clustering methods for hyperspectral images.

Category	Sub-Category	Algorithms	Remarks
Model-based	Self-representation	SSC [5], LRR [7], JSSC [13], S4C [21], LCR-FLDA [43]	Learn global or structured self-representation coefficients; effective for capturing subspace structure; performance degrades on nonlinear manifolds; do not scale well.
	Dictionary-based	Sketch-TV [24], IDLSC [25], SPGSC [27], (TV-CRC-LAD) [26]	Use compact or structured dictionaries to improve scalability; suitable for large HSIs but still rely on linear reconstruction assumptions.
Deep learning-based	Data-driven	DEC [32], DEN [28], PICA [34], PARTY [29] SpectralNet [30], N2D [31]	Learn latent representations in an end-to-end manner; flexible and scalable, but rely heavily on network design and sufficient data.
	Model-aware	DSCNet [10], SDSC-AI [14], HyperAE [42], LRDSC [15] PSSC [12], DSC [11]	Incorporate model priors into deep networks to improve accuracy; usually involve complex architectures and higher computational cost.

The emerging basis-representation-based subspace clustering method [18] attempts to learn the basis of subspaces to obtain accurate assignments of data points. This method builds on the property that all vectors within a subspace can be expressed as linear combinations of that subspace's basis vectors, as illustrated in Figure 2. It achieves comparable performance to self-representation-based methods in image and text-level tasks with linear complexity. However, this method shows limited performance in HSI clustering, as image- and text-level clustering tasks treat each data point as an entire image or document, without considering spatial relationships, rendering spatial context irrelevant. In contrast, HSI clustering involves data points corresponding to individual pixels or image patches, where spatial continuity and non-local structure are critical for accurate clustering.

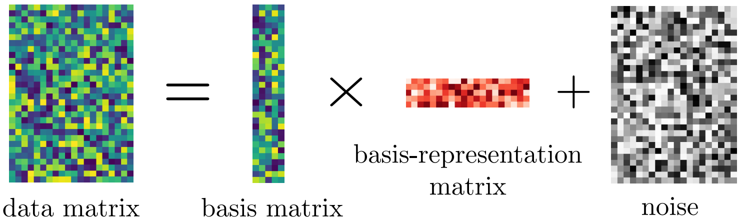


Figure 2. An illustration of the basis-representation model.

Our contribution is motivated by the extension of this basis-representation model to hyperspectral data through the integration of structural constraints that explicitly preserve both spatial continuity and non-local structure.

3. Proposed Method

In this section, we present our main contribution, that is, a scalable and context-preserving deep subspace clustering method. To address the high computational complexity of traditional self-representation-based clustering approaches, we propose a novel strategy that avoids learning a self-representation matrix. Instead, our method directly learns a compact subspace basis, effectively integrating both local and non-local structural information inherent to HSI data. In this section, we first formally define the problem tackled, then we detail the structure constraints that are central to our approach, and finally we introduce our end-to-end training strategy. The overall framework of our proposed method is depicted in Figure 3.

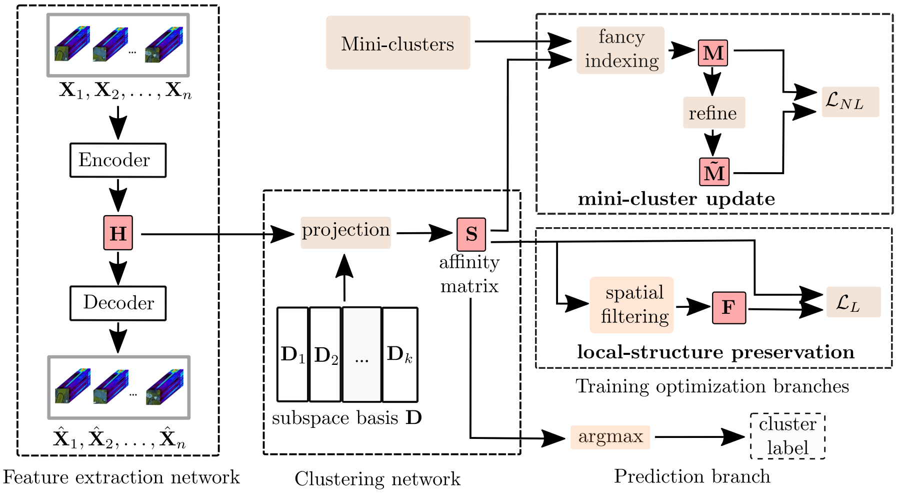

$\mathbf{X}_1, \mathbf{X}_2, \ldots, \mathbf{X}_n$
Encoder
H
Decoder
$\hat{\mathbf{X}}_1, \hat{\mathbf{X}}_2, \ldots, \hat{\mathbf{X}}_n$
Feature extraction network
Mini-clusters
fancy indexing
M
refine
$\tilde{\mathbf{M}}$
$\mathcal{L}_{NL}$
mini-cluster update
projection
S
affinity matrix
$\mathbf{D}_1$ $\mathbf{D}_2$ ... $\mathbf{D}_k$
subspace basis D
Clustering network
spatial filtering
F
$\mathcal{L}_L$
local-structure preservation
Training optimization branches
argmax
cluster label
Prediction branch

Figure 3. Structure of the proposed method. The autoencoder maps the input data nonlinearly to a latent representation **H**. This representation is then projected onto various subspace bases to form the subspace affinity matrix **S**. To enhance the quality of the subspace basis, the following optimization modules are applied: (1) the mini-cluster updating module, which generates mini-cluster assignment **M** and updates it by minimizing the KL divergence loss to a refined version $\tilde{\mathbf{M}}$; (2) the local-structure-preserving module, which encourages the subspace affinity matrix **S** to be similar to its smooth version **F**.

3.1. Problem Formulation

Let $\mathcal{X} = \{\mathbf{X}_1, \mathbf{X}_2, \ldots, \mathbf{X}_n\}$ denote a hyperspectral image under investigation, divided into n overlapping patches, where each patch $\mathbf{X}_i \in \mathbb{R}^{a \times a \times b}$ is centered on a pixel and represents a 3D region of the hyperspectral image, extending over a small $a \times a$ spatial neighborhood and all b spectral bands. The goal of HSI clustering is to group these patches into k distinct classes in an unsupervised manner. In this study, k is assumed to be known in advance, which enables consistent comparison with the ground-truth labels. Note that in the subsequent discussion, the term "pixel" refers to the target center pixel of each patch, and both share the same class assignment. This patch-overlap strategy is widely adopted in hyperspectral clustering [42,44], as it enables spatial context modeling for each target pixel. Despite the induced redundancy, it provides consistent context cues that improve learning robustness.

Traditional methods often rely on a two-stage framework involving an $n \times n$ self-representation matrix, which results in $\mathcal{O}(n^2)$ computational complexity. Moreover, the self-representation matrix calculation and spectral clustering are performed separately, with structure preservation applied only during the first stage, thereby limiting its impact on the overall clustering process.

In this work, we propose a scalable context-preserving deep subspace clustering method for HSIs. Specifically, we aim to learn the bases of different subspaces to cluster the HSI patches. Based on the learned subspace bases, we compute the affinity matrix $\mathbf{S} \in \mathbb{R}^{n \times k}$, where each entry $s_{i,j}$ denotes the soft assignment of the ith pixel to the jth subspace. Let $\mathbf{H} = [\mathbf{h}_1, \ldots, \mathbf{h}_n]^\top \in \mathbb{R}^{n \times d}$ be the embedded representation of all pixels with dimension d produced by the encoder of a deep autoencoder and $\mathbf{D} = \{\mathbf{D}_1, \ldots, \mathbf{D}_k\}$ be the set of the learned subspace bases, with each $\mathbf{D}_j \in \mathbb{R}^{d \times r}$ representing the basis of the jth subspace (with r basis vectors). The soft assignment between pixel i and subspace j is then computed as

$$s_{i,j} = \frac{\|\mathbf{h}_i^\top \mathbf{D}_j\| + \theta r}{\sum_j (\|\mathbf{h}_i^\top \mathbf{D}_j\| + \theta r)}, \tag{2}$$

where θ is a smoothing constant set to 5 following [18]. Collecting all assignments yields the vector $\mathbf{s}_i = [s_{i,1}, \ldots, s_{i,k}]$, the ith row of $\mathbf{S}$. Each pixel is finally assigned to the subspace with the largest soft assignment, i.e., $\operatorname{argmax}_j s_{i,j}$.

To achieve optimal clustering performance on HSI data, the learned bases must satisfy the following criteria: (1) each subspace should have a distinct basis; (2) the bases should exhibit strong discriminative power to distinguish data points from different subspaces; (3) the bases should capture the intrinsic geometry (that is, spectral properties) of the data for the non-local structure to be preserved during clustering; and (4) the bases should leverage the local structure (that is, spatial continuity) to enhance robustness during feature-basis alignment. Similar to model-aware deep subspace clustering methods, basis learning is formulated within the latent space of an autoencoder.

Based on these requirements, we formulate the optimization problem. To make the optimization targets explicit, let Θ denote the trainable parameters of the deep autoencoder. Since the latent representation matrix $\mathbf{H}$ and the reconstructions $\hat{\mathcal{X}} = \{\hat{\mathbf{X}}_i\}_{i=1}^n$ are outputs of the network, they are computed from the input patches $\mathcal{X} = \{\mathbf{X}_i\}_{i=1}^n$ by the autoencoder parameterized by Θ. Consequently, we formulate the joint optimization problem with respect to the subspace bases $\mathbf{D}$ and the network parameters Θ as follows:

$$\min_{\mathbf{D},\Theta} \frac{1}{2n} \sum_{i=1}^{n} ||\mathbf{X}_i - \hat{\mathbf{X}}_i||_F^2 + \beta\phi(\mathbf{D}) + \beta_1\eta(\mathbf{HD}) + \beta_2\Psi(\mathbf{HD}), \tag{3}$$

which consists of the data fidelity term and three regularization terms: $\phi(\mathbf{D})$—promoting discriminative power of the learned bases, $\eta(\mathbf{HD})$—non-local (spectral) similarities, and $\Psi(\mathbf{HD})$—local (spatial) consistency within the clusters. $\hat{\mathbf{X}}_i$ is the reconstructed version of $\mathbf{X}_i$, obtained from the corresponding latent code $\mathbf{h}_i$ by the decoder of the same autoencoder, which constrains the latent representation $\mathbf{H}$ to retain essential spatial–spectral information, and the improved latent representation in turn leads to better estimation of the subspace bases $\mathbf{D}$ during training. The basis dissimilarity term $\phi(\mathbf{D})$ ensures that the learned subspace bases are distinct, enabling distinguishing data points from different subspaces. The non-local structure preservation term $\eta(\mathbf{HD})$ encourages data points to have a prediction similar to those of their nearest neighbors in the spectral space. The local structure preservation term $\Psi(\mathbf{HD})$ maintains spatial dependencies, ensuring that data points share similar affinities to the subspace basis as their spatial neighbors. These constraints are discussed in detail in the following sections. The positive constants β, β_1, and β_2 balance the contributions of the different terms of the objective function, and their values are discussed in the experimental section (see Section 4).

In contrast to self-representation-based methods, our model does not require maintaining a self-representation matrix of size $n \times n$. Instead, it uses a basis-representation matrix of size $rk \times n$, reducing the computational complexity from $\mathcal{O}(n^2)$ to $\mathcal{O}(krn)$. Here, the number of clusters k and the number of basis vectors per cluster r are typically small constants independent of the sample size n. Since $kr \ll n$ for large-scale HSI data, the effective complexity simplifies to $\mathcal{O}(n)$. Moreover, our model follows a one-stage approach, where both local and non-local structure constraints jointly optimize the entire clustering process end to end, thereby offering stronger guidance for model optimization.

3.2. Basis Dissimilarity Constraint

In subspace clustering, the bases of each subspace must be distinct, and orthogonality is often employed to reinforce this distinction, which has been shown to be beneficial for clustering performance in prior work [18]. Additionally, the bases are kept on the same scale to ensure more consistent and effective evaluation. This reduces overlap between subspaces, enhances the discriminative power of the bases, and ultimately leads to more accurate and robust clustering results. To ensure these properties are maintained during basis learning, we adopt a basis dissimilarity constraint $\phi(\mathbf{D})$, similar to that in [18], as described below:

$$\phi(\mathbf{D}) = \|\mathbf{D}^\top \mathbf{D} \odot \mathbf{O}\|_F^2 + \|\mathbf{D}^\top \mathbf{D} \odot \mathbf{I} - \mathbf{I}\|_F^2, \tag{4}$$

where $\odot$ represents the Hadamard product, $\mathbf{O} \in \mathbb{R}^{kr \times kr}$ is a matrix with diagonal blocks of size $r \times r$ are 0, others 1, and $\mathbf{I}$ is the identity matrix of appropriate dimensions. This constraint effectively enforces each subspace to have orthonormal bases (unit-norm and mutually orthogonal) and pushes bases from different subspaces apart. In other words, $\phi(\mathbf{D})$ helps $\mathbf{D}$ behave like a block-wise orthogonal basis set, enhancing subspace separation and stability.

3.3. Non-Local Structure Preservation

Image pixels with similar spectral responses are likely to belong to the same land cover class, regardless of their spatial location. Making use of these non-local spectral similarities helps to preserve the correct non-local structure in a clustering map. Existing works often apply a Laplacian matrix to maintain this structure in the self-representation matrix, which is computationally expensive and limits scalability for large datasets.

To address this issue, we propose a mini-cluster updating scheme that ensures the subspace bases align with this non-local structure of the data while efficiently preserving it in the final clustering map. Specifically, the original data points are first grouped into mini-clusters using the algorithm FINCH (see Section 2.1 for details). Let $\mathcal{C} = \{\mathcal{C}_1, \mathcal{C}_2, \mathcal{C}_3, \ldots, \mathcal{C}_l\}$ represent the set of l mini-clusters generated by FINCH, where each mini-cluster comprises a group of neighboring data points. Formally, each mini-cluster $\mathcal{C}_p$ contains the indices of the data points belonging to this mini-cluster.

We expect data points in each mini-cluster $\mathcal{C}_p$ to align more strongly with a common basis $\mathbf{D}_j$ than with any other $\mathbf{D}_e$, as shown below:

$$\left\|\mathbf{h}_q^\top \mathbf{D}_j\right\| \gg \max_{e \neq j} \left\|\mathbf{h}_q^\top \mathbf{D}_e\right\| \quad \text{for all } q \in \mathcal{C}_p, \tag{5}$$

where $\mathbf{h}_q \in \mathbb{R}^d$ represents the latent representation of the qth data point. According to the definition of soft assignment in (2), the subspace basis affinity in (5) can be mapped to soft assignments, resulting in similar assignments for all data points within the same mini-cluster; that is, for a given mini-cluster $\mathcal{C}_p$,

$$\mathbf{s}_q \approx \mathbf{s}_o \text{ for all } q \text{ and } o \in \mathcal{C}_p. \tag{6}$$

To preserve the underlying structure, we optimize the soft assignment at the mini-cluster level to encourage data points within the same mini-cluster to share the same assignment. During this processing, the soft assignments of data points within each mini-cluster are extracted using their mini-cluster index through fancy indexing, a vectorized technique that enables efficient, loop-free index-based value extraction on GPUs. The soft assignments of mini-clusters are represented as $\mathbf{M} \in \mathbb{R}^{l \times k} = [\mathbf{m}_1, \mathbf{m}_2, \ldots, \mathbf{m}_l]^\top$. The soft assignment for the pth mini-cluster $\mathbf{m}_p \in \mathbb{R}^k$ is obtained by averaging the assignments of the data points it contains as follows:

$$\mathbf{m}_p = \frac{1}{|\mathcal{C}_p|} \sum_{q \in \mathcal{C}_p} \mathbf{s}_q. \tag{7}$$

Since the soft assignment matrix $\mathbf{S}$ depends on the feature embedding $\mathbf{H}$ and the cluster subspace bases $\mathbf{D}$, this target matrix $\mathbf{M}$ is implicitly a function of $(\mathbf{H}, \mathbf{D})$. To further enhance the distribution of these assignments, we adopt a refined soft assignment $\tilde{\mathbf{M}} \in \mathbb{R}^{l \times k}$ whose entries are defined as

$$\tilde{m}_{p,j} = \frac{m_{p,j}^2 / \sum_p m_{p,j}}{\sum_j (m_{p,j}^2 / \sum_p m_{p,j})}, \tag{8}$$

where $m_{p,j}$ represents the soft assignment of the pth mini-cluster to class j and $\tilde{m}_{p,j}$ is its refined soft assignment. This refinement process improves cluster purity by emphasizing high-confidence predictions and mitigating distortions caused by large clusters [32]. By aligning the initial mini-cluster predictions with their refined versions, the quality of soft assignments is enhanced, which in turn strengthens the discriminative power of the subspace bases. Based on the above analysis, we define the non-local structure preservation constraint $\eta(\mathbf{HD})$ as follows:

$$\eta(\mathbf{HD}) = \mathrm{KL}(\tilde{\mathbf{M}} \| \mathbf{M}) = \sum_p \sum_j \tilde{m}_{p,j} \log \frac{\tilde{m}_{p,j}}{m_{p,j}}, \tag{9}$$

where $\tilde{\mathbf{M}}$ represents the refined soft assignment affinity matrix of the mini-clusters. During training, the refinement calculation increasingly emphasizes class distinctions, causing the refined mini-cluster soft assignment $\tilde{\mathbf{m}}_i$ to converge towards a state where one class dominates, with its value approaching 1, while the values for other classes approach 0. As $\eta(\mathbf{HD})$ decreases over time, the mini-cluster is assigned more confidently to a single class.

In parallel, the soft assignment of data points within the mini-cluster follows the same optimization trajectory as the mini-cluster itself as follows:

$$\frac{\partial \eta(\mathbf{HD})}{\partial \mathbf{s}_q} = \frac{\partial \eta(\mathbf{HD})}{\partial \mathbf{m}_p} \frac{\partial \mathbf{m}_p}{\partial \mathbf{s}_q} = \frac{1}{|\mathcal{C}_p|} \frac{\partial \eta(\mathbf{HD})}{\partial \mathbf{m}_p}, \quad \text{for all } q \in \mathcal{C}_p. \tag{10}$$

This means that as the mini-cluster moves toward being classified as a particular class, all the data points within it also shift toward the same class assignment. As training progresses, this optimization direction causes all the points in the mini-cluster to converge to the same class assignment. The mini-cluster updating scheme enhances the model's performance in two ways. First, it optimizes the mini-cluster soft assignments, which are more representative and robust than instance assignments, because of the assignment averaging. Second, it promotes consistency among data points within the same mini-cluster, ensuring they share the same prediction and preserving the non-local structure during optimization. Unlike previous methods that relied on a Laplacian matrix with complexity $\mathcal{O}(n^2)$ to preserve the non-local structure, our approach is significantly more efficient. By leveraging a mini-cluster index vector of size n and using fancy indexing with complexity

$\mathcal{O}(n)$, which aligns with Graphics Processing Unit (GPU) implementation efficiency, we achieve efficient non-local structure preservation. Moreover, our method integrates non-local structure preservation throughout the entire clustering process, rather than limiting it to a single stage. This provides stronger guidance and improves the overall clustering performance.

3.4. Local Structure Preservation

In real-world scenarios, neighboring areas often belong to the same land cover class, a property known as spatial continuity. This relationship is a common phenomenon in many types of data, including HSI, where adjacent pixels are likely to belong to the same class. To leverage this property, our model integrates spatial neighborhood information into the basis learning process. This not only improves noise robustness when measuring the alignment between feature vectors and basis vectors but also ensures that the clustering results maintain the intrinsic spatial continuity of HSI. Specifically, as with the non-local structure preservation constraint, we directly apply spatial filtering to the soft assignments of pixels, as illustrated in Figure 4.

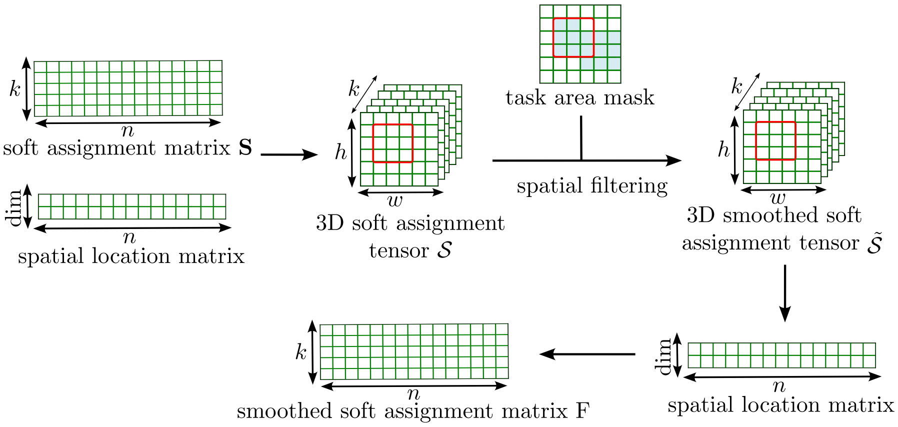

k
n
soft assignment matrix **S**
dim
n
spatial location matrix
k
h
w
3D soft assignment tensor $\mathcal{S}$
task area mask
spatial filtering
k
h
w
3D smoothed soft assignment tensor $\tilde{\mathcal{S}}$
k
n
smoothed soft assignment matrix F
dim
n
spatial location matrix

Figure 4. Calculation process of the smoothed soft assignment matrix **F**. First, the soft assignments of image pixels are arranged into a 3D tensor based on their spatial locations. Next, a local window is applied to perform mean filtering for each point. Specifically, the filtering is calculated within the task area defined by a mask **T**, where labeled data are available, to facilitate effective evaluation of the results.

As defined in the figure, $\mathcal{S} \in \mathbb{R}^{w \times h \times k}$ represents a 3D tensor derived from the 2D soft assignment matrix, where w and h are the width and height of the HSI. Each $\mathcal{S}_{x,y} \in \mathbb{R}^k$ denotes the soft assignment vector for the pixel at spatial location (x, y). By incorporating spatial information, this tensor facilitates a spatially aware prediction of cluster membership. A spatial filtering operation is subsequently applied to each layer of $\mathcal{S}$, leveraging relationships among neighboring pixels to produce a smoothed 3D tensor $\tilde{\mathcal{S}}$, expressed as

$$\tilde{\mathcal{S}}_{x,y} = \frac{\sum_{(u,v) \in \mathbf{W}_{x,y}} t_{u,v} \cdot \mathcal{S}_{u,v}}{\sum_{(u,v) \in \mathbf{W}_{x,y}} t_{u,v}}, \tag{11}$$

where $\tilde{\mathcal{S}}_{x,y}$ is the smoothed soft assignment vector at spatial location (x, y), with x and y denoting the row and column indices, respectively, and $\mathbf{W}_{x,y}$ is a fixed smoothing window centered at (x, y). The mask $\mathbf{T} \in \mathbb{R}^{w \times h}$ assigns a value of 1 to pixels within the cluster region and 0 otherwise, where $t_{x,y}$ denotes the value at the corresponding location in $\mathbf{T}$. We then extract the final smoothed soft assignment matrix $\mathbf{F} \in \mathbb{R}^{n \times k}$ by flattening $\tilde{\mathcal{S}}$ according to pixel ordering.

To incorporate this refined local structure into the original predictions, we minimize the KL divergence loss between the original and the smoothed predictions. The local structure preservation function $\Psi(\mathbf{HD})$ is defined as follows:

$$\Psi(\mathbf{HD}) = \mathrm{KL}(\mathbf{F}\|\mathbf{S}) = \sum_i \sum_j f_{i,j} \log \frac{f_{i,j}}{s_{i,j}}, \tag{12}$$

where $\mathbf{F} \in \mathbb{R}^{n\times k}$ is the smoothed assignment matrix, $f_{i,j}$ represents the element at position (i,j), and $\mathbf{S} \in \mathbb{R}^{n\times k}$ is the original soft assignment matrix computed from the embedded features $\mathbf{H}$ and the subspace bases $\mathbf{D}$ via Equation (2).

Existing methods typically apply spatial constraints, such as spatial smoothing, on a self-representation 3D tensor $\mathcal{M} \in \mathbb{R}^{w\times h\times n}$, where w and h denote the width and height of the image, resulting in a computational complexity of $\mathcal{O}(n^2)$ [21]. Other approaches apply total variation regularization on a dictionary representation tensor $\mathcal{T} \in \mathbb{R}^{w\times h\times n'}$, where $n' \gg k$ is the dictionary dimension, leading to a complexity of $\mathcal{O}(nn' \log n)$ [24].

In contrast, our method directly applies spatial filtering to the clustering soft assignment $\mathcal{S} \in \mathbb{R}^{w\times h\times k}$. Denoting the size of the smoothing window as $|\mathbf{W}|$, the computational complexity is $\mathcal{O}(|\mathbf{W}| \times n \times k)$, which simplifies to $\mathcal{O}(n)$, making it much more efficient. More importantly, our spatial constraint optimizes the entire clustering process, providing stronger guidance for network optimization.

3.5. Objective Function and Training Strategy

The overall objective function consists of multiple loss terms that jointly optimize the reconstruction of the autoencoder, basis dissimilarity, and both local and non-local structure preservation. Specifically, the reconstruction loss is defined as

$$\mathcal{L}_R = \frac{1}{2n}\sum_{i=1}^{n} ||\mathbf{X}_i - \hat{\mathbf{X}}_i||_F^2, \tag{13}$$

which measures the reconstruction error between the input $\mathbf{X}_i$ and its reconstruction $\hat{\mathbf{X}}_i$. This term encourages the network to preserve the essential spectral information of each pixel during feature learning, ensuring that the latent representation retains sufficient fidelity for downstream clustering. Based on the definition of $\phi(\mathbf{D})$ in Equation (4), the basis dissimilarity loss is formulated as

$$\mathcal{L}_D = \phi(\mathbf{D}), \tag{14}$$

which enforces diversity among the learned basis vectors to avoid redundancy in representation. To preserve both spatial and spectral consistency, we further introduce two structure-preserving constraints. The non-local preservation loss is defined as

$$\mathcal{L}_{NL} = \eta(\mathbf{HD}), \tag{15}$$

where $\eta(\mathbf{HD})$ (introduced in Equation (9)) enforces non-local spectral consistency between spectrally similar but spatially distant pixels. Similarly, the local preservation loss is given by

$$\mathcal{L}_L = \Psi(\mathbf{HD}), \tag{16}$$

where $\Psi(\mathbf{HD})$ (defined in Equation (12)) preserves spatial smoothness by encouraging neighboring pixels to exhibit consistent affinities. Finally, the total objective function is given by

$$\mathcal{L}_{\text{total}} = \mathcal{L}_R + \beta\mathcal{L}_D + \beta_1\mathcal{L}_{NL} + \beta_2\mathcal{L}_L, \tag{17}$$

where β, β_1, and β_2 are trade-off parameters controlling the relative importance of each loss term. By minimizing this objective function, both the autoencoder parameters Θ and the learnable subspace bases $\mathbf{D}$ are jointly updated through back-propagation. This optimization forms a coordinated dual mechanism: the gradients arising from the soft assignments $\mathbf{S}$ update the subspace bases $\mathbf{D}$ and further propagate through $\mathbf{H}$ to the encoder, guiding the latent features to align with the underlying subspace structures. Meanwhile, the reconstruction loss between the input $\mathcal{X}$ and the output $\hat{\mathcal{X}}$ anchors this evolution, ensuring that the evolving representation $\mathbf{H}$ remains grounded in the intrinsic spectral–spatial information needed for faithful data reconstruction.

As shown in Algorithm 1, the network training involves four main steps (Steps 1–4), followed by a final assignment step (Step 5). First, the mini-clusters are generated by FINCH. Next, the autoencoder is pre-trained to obtain initial latent representations of the input data. Then, the K-means algorithm is applied to generate the initial clustering results, which are subsequently processed using SVD to obtain the initial subspace basis for each cluster. Following work [18], we select five main basis vectors, i.e., the top 5 singular vectors with the largest singular values, to capture the subspace structure within each cluster. In the joint optimization step (Step 4), the autoencoder and the subspace bases are jointly optimized, where their parameters are updated under the supervision of all constraints.

Algorithm 1 Training Process for Hyperspectral Image Clustering.

```
Input: Hyperspectral image patches: $\mathcal{X} = \{\mathbf{X}_1, \mathbf{X}_2, \dots, \mathbf{X}_n\}$
Output: Cluster labels
Step 1: Mini-cluster generation (FINCH)
Generate mini-clusters: $\mathcal{C} = \{\mathcal{C}_1, \mathcal{C}_2, \mathcal{C}_3, \dots, \mathcal{C}_l\}$
Step 2: Pre-training
Perform data preprocessing and pre-train the deep autoencoder; obtain initial $\mathbf{H}$.
Minimize the reconstruction loss:
                         $\mathcal{L} = \mathcal{L}_R$
Step 3: Initial subspace basis construction
Apply K-means clustering on $\mathbf{H}$ to generate initial clusters.
Initialize the subspace basis $\mathbf{D}$ for the initial clusters using Singular Value Decomposition (SVD).
Step 4: Joint Optimization (End-to-End)
for epoch = 1 to MaxEpochs do
    Forward Pass (Compute Variables):
        Compute latent features: $\mathbf{H} \leftarrow \mathrm{Encoder}(\mathcal{X})$.
        Compute reconstruction: $\hat{\mathcal{X}} \leftarrow \mathrm{Decoder}(\mathbf{H})$.
        Compute soft assignments $\mathbf{S}$ via Equation (2) with $\mathbf{H}$ and $\mathbf{D}$.
        Update target distributions $\mathbf{M}$, $\tilde{\mathbf{M}}$, and $\mathbf{F}$ based on current $\mathbf{S}$.
        Compute total loss $\mathcal{L}_{\text{total}}$ via Equation (17).
    Backward Pass (Update Model Parameters):
        Update subspace bases $\mathbf{D}$ using gradients of $\mathcal{L}_{\text{total}}$.
        Update autoencoder parameters $\Theta$ via back-propagation of $\mathcal{L}_{\text{total}}$.
end for
Step 5: Final Assignment
Assign each data point to a cluster based on the highest value in its soft assignment vector:
                         $\mathrm{label}_i = \underset{j}{\mathrm{argmax}}\, s_{i,j}$.
```

Our model jointly optimizes these components, enabling local homogeneity to propagate to non-local data points through the non-local constraint and vice versa. In contrast, self-representation-based subspace clustering methods have a complexity of $\mathcal{O}(n^2)$ for computing the self-representation matrix, maintaining the non-local structure with Laplacian regularization, and preserving spatial dependency on the self-representation matrix. Our method achieves a complexity of $\mathcal{O}(n)$ for all key operations, including calculating the soft assignment, maintaining the non-local structure, and preserving the local structure, making it scalable for large datasets. Furthermore, the proposed structural constraints supervise the entire clustering process, providing stronger guidance for optimization.

4. Experiments and Results

In this section, we evaluate the proposed method on four real-world hyperspectral images. We compare it against several popular clustering algorithms. Then, we perform an ablation study to understand the impact of the local and non-local structure preservation constraints.

4.1. Datasets

We conducted experiments on four real-world hyperspectral image datasets. The details of these datasets are as follows:

1. Trento dataset: This dataset was acquired using the Compact Airborne Spectrographic Imager (CASI) sensor and contains 63 spectral bands. It is divided into 6 classes. The image size is 600 × 166 pixels, with 30,214 labeled samples.
2. Houston dataset: The Houston dataset was collected using the ITRES Compact Airborne Spectrographic Imager (ITRES-CASI) sensor, which captures high-resolution hyperspectral imagery across 144 spectral bands. It is categorized into 7 classes. The image size is 130 × 130 pixels, containing 6104 labeled samples.
3. PaviaU dataset: The PaviaU dataset was acquired with the Reflective Optics System Imaging Spectrometer (ROSIS-3) sensor, providing 103 spectral bands. It is classified into 9 classes. The image size is 610 × 340 pixels, with 42,776 labeled samples.
4. HYPSO-1 dataset: The HYPSO-1 dataset originates from the Hyperspectral Small Satellite for Ocean Observation (HYPSO-1) CubeSat mission, which provides hyperspectral imagery covering sea, land, and cloud regions with approximately 120 spectral bands. For our experiments, a 150 × 150 spatial region was selected from one labeled scene, containing three major classes (sea, land, and cloud) with 22,500 labeled samples.

These datasets feature a variety of sensor types, numbers of spectral bands, class categories, and image dimensions, providing a diverse experimental platform to validate the effectiveness of our proposed method.

4.2. Experimental Setting

During training, the model is trained for 400 epochs without early stopping. To balance memory consumption and enrich the feature representation of each HSI pixel, we use a patch size of 7 × 7 for all compared methods. When generating mini-clusters, we employ a larger patch size of 17 × 17 to incorporate more detailed neighborhood information. Parameter β is fixed at 10^{-3}. To ensure statistical robustness, all experiments are repeated 10 times with different random seeds. We report the mean ± standard deviation of all performance metrics. Statistical significance is evaluated using the Wilcoxon signed-rank test to compare our method against each baseline. Detailed statistical results are provided in the Appendix A. All implementation details, including hyperparameter settings and training schedule, are available in our released code. The code and data are publicly available, see the "Data Availability Statement" section for details.

We compare our method with several clustering approaches, including centroid-based methods such as K-means [45] and Fuzzy C-means (FCM) [46]; the graph-based method spectral clustering (SC) [47]; data-driven deep clustering methods such as improved deep embedded clustering (IDEC) [48], SpectralNet (SN) [30], N2D [31], and deep embedded K-means (DEKM) clustering [49]; self-representation-based deep subspace clustering method HyperAE [42] and the nearest neighbor-based method FINCH [20]. Additionally, we compare our method with our preliminary model-aware deep learning (MADL) framework [19], which preserves non-local structures through contrastive learning.

To assess the performance of our model, we employ three widely used metrics: Overall Accuracy (OA), Normalized Mutual Information (NMI), and Cohen's Kappa ($\mathcal{K}$). The OA quantifies the proportion of correctly classified samples relative to the total number of samples, computed by

$$\mathrm{OA} = \frac{1}{n}\sum_{i=1}^{n}\mathbb{I}(g_i = \hat{g}_i) \tag{18}$$

where n is the total number of samples, g_i is the true label of the ith sample, $\hat{g}_i$ is the predicted label of the ith sample, and $\mathbb{I}(\cdot)$ is the indicator function that equals 1 if the condition inside is true and 0 otherwise. The NMI measures the mutual information between the clustering results and the true labels, normalized by the average of their entropies, defined by

$$\mathrm{NMI} = \frac{2 \times \mathrm{I}(\mathbf{G};\hat{\mathbf{G}})}{\mathrm{H}(\mathbf{G}) + \mathrm{H}(\hat{\mathbf{G}})} \tag{19}$$

where $\mathrm{I}(\mathbf{G};\hat{\mathbf{G}})$ is the mutual information between the true labels $\mathbf{G}$ and the clustering labels $\hat{\mathbf{G}}$ and $\mathrm{H}(\mathbf{G})$ and $\mathrm{H}(\hat{\mathbf{G}})$ are the entropies of $\mathbf{G}$ and $\hat{\mathbf{G}}$, respectively. $\mathcal{K}$ assesses the agreement between the clustering results and the true labels while accounting for the possibility of agreement occurring by chance, calculated by

$$\mathcal{K} = \frac{P_o - P_e}{1 - P_e} \tag{20}$$

where P_o is the observed agreement among raters and P_e is the expected agreement by chance. For the three evaluation metrics, a higher value indicates better performance. Additionally, we report the computational time in seconds to compare the efficiency of different methods.

4.3. Performance Analysis

4.3.1. Houston Dataset

The clustering results of different methods on the Houston dataset are presented in Table 2 and Figure 5. We set $\beta_1 = 3$, $\beta_2 = 8$, and use a filter window size of 3×3. We observe that only the deterministic method FINCH [20] produces fully consistent results across runs and metrics. To provide a fair comparison, we further conduct a Wilcoxon signed-rank analysis among different methods, as summarized in Table A1. The results show that most pairwise differences with our method are statistically significant ($p_w < 0.05$, many at $p_w < 0.01$), confirming the reliability of the observed improvements. For the few cases where the differences are not significant, such as HyperAE and MADL, our model still achieves higher mean accuracies and smaller deviations, showing more stable and consistent performance. IDEC also attains a comparable mean in NMI but with larger variance, while FINCH, as a deterministic algorithm, produces fixed outputs without statistical variance.

Table 2. Quantitative evaluation of different clustering methods on the dataset Houston.

Class	K-Means [45]	FCM [46]	SC [47]	IDEC [48]	FINCH [20]	DEKM [49]	SN [30]	HyperAE [42]	N2D [31]	MADL [19]	SCDSC
1	46.50	46.50	**62.46**	47.20	53.50	45.53	41.40	47.16	51.84	47.20	48.60
2	**100**	**100**	**100**	**100**	**100**	**100**	**100**	99.62	99.98	**100**	**100**
3	58.05	0.0	35.66	**88.94**	70.61	24.30	32.11	70.03	67.76	74.70	79.66
4	**100**	**100**	**100**	96.37	**100**	**100**	76.44	99.36	63.10	99.96	99.65
5	94.77	0.70	90.00	94.54	**100**	85.69	80.00	100.00	38.92	69.38	60.00
6	0	27.07	0	0	**70.94**	0	1.54	0	10.85	23.05	30.70
7	0	0.46	0	9.97	0	4.00	59.16	36.60	**73.84**	39.56	47.97
OA(%) Mean	64.50	63.23	65.86	67.58	72.10	61.01	60.77	70.36	63.61	72.33	**74.41**
OA Std	0.08	0.01	5.24	1.20	0.00	3.00	3.79	5.50	2.22	5.85	4.21
NMI Mean	0.6973	0.5935	0.6181	0.7851	0.7702	0.7011	0.6439	0.7697	0.7465	0.7656	**0.7902**
NMI Std	0.0006	0.0002	0.0692	0.0269	0.00	0.0484	0.0475	0.0400	0.0213	0.0266	0.0329
$\mathcal{K}$ Mean	0.5354	0.5250	0.5441	0.5859	0.6424	0.4922	0.5057	0.6225	0.5679	0.6492	**0.6759**
$\mathcal{K}$ Std	0.0011	0.0001	0.0715	0.0204	0.00	0.0412	0.0568	0.0700	0.0276	0.0793	0.0590
Time (sec) Mean	2.21	7.32	2.68	42.96	**0.73**	67.88	16.75	419.40	26.40	60.52	33.43
Time Std	0.17	3.06	0.30	0.48	0.16	9.31	0.37	42.10	0.80	1.07	1.11

The best results are highlighted in **bold**, and the second-best results are underlined.

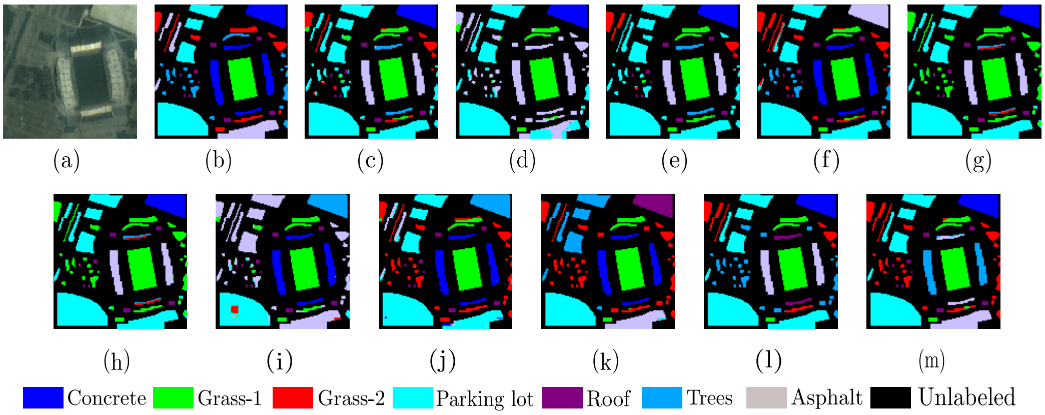


Figure 5. Houston dataset. (**a**) False-color image. (**b**) Ground truth and the clustering results obtained by (**c**) K-means, (**d**) FCM, (**e**) SC, (**f**) IDEC, (**g**) FINCH, (**h**) DEKM, (**i**) SpectralNet, (**j**) HyperAE, (**k**) N2D, (**l**) MADL, and (**m**) SCDSC.

Overall, our method achieves the best performance across all three metrics. Compared to our preliminary work [19], our method gains an additional 2.08% in OA, demonstrating the effectiveness of the mini-cluster updating scheme in enhancing representation stability and clustering accuracy. Compared with HyperAE, which relies on global self-representation reconstruction, our basis-representation model achieves better accuracy with far less computational cost. Meanwhile, data-driven deep clustering methods such as N2D, DEKM, and SpectralNet exhibit inferior and less stable results due to their dependence on feature distribution learning without explicit structural constraints. FINCH also performs strongly among non-deep methods, reflecting its ability to model complex local relationships through a nearest neighbor chain mechanism.

In terms of runtime, FINCH remains the fastest, and traditional clustering methods are generally more efficient than deep learning–based approaches. Nevertheless, our method completes the clustering task in approximately 33 s, whereas HyperAE requires more than 400 s due to its $\mathcal{O}(n^2)$ self-representation complexity. This clearly highlights the computational advantage of adopting a compact basis representation instead of full self-representation modeling.

From the cluster map, we observe that our method best aligns with the ground truth. Specifically, our approach accurately distinguishes between parking lot and asphalt areas, while other methods often merge them. Only SpectralNet and N2D partially recognize these distinctions.

4.3.2. Trento Dataset

The clustering results of various methods on the Trento dataset are presented in Table 3 and Figure 6. We set $\beta_1 = 3$, $\beta_2 = 1$, and use a filter window size of 7×7. We observe that most methods achieve accuracies above 70%, which may be attributed to class imbalance, as several dominant classes occupy the majority of samples. Due to this imbalance, most approaches, including ours, fail to recognize the third class. According to the Wilcoxon signed-rank analysis in Table A1, our method achieves statistically significant improvements ($p_w < 0.05$) over nearly all baselines on OA and $\mathcal{K}$, confirming its reliability and robustness. Compared with our preliminary work [19], the proposed model achieves an additional 2.31% improvement in OA (90.61% vs. 88.30%) while maintaining almost identical NMI (0.9101 vs. 0.9144). This indicates that the new non-local structure preservation scheme is more adaptive than using a fixed number of contrastive neighbors. Among the compared methods, HyperAE fails on this dataset because its self-representation step involves $\mathcal{O}(n^2)$ memory usage, which leads to out-of-memory errors. Meanwhile, our basis-representation formulation maintains linear scalability and achieves better results than MADL.

Table 3. Quantitative evaluation of different clustering methods on the dataset Trento. "–" indicates out-of-memory during execution.

Class	K-Means [45]	FCM [46]	SC [47]	IDEC [48]	FINCH [20]	DEKM [49]	SN [30]	HyperAE [42]	N2D [31]	MADL [19]	SCDSC
1	71.37	0	0	89.50	93.95	99.39	73.69	–	99.62	99.98	**100**
2	14.33	1.2	7.82	35.38	0	31.85	39.84	–	26.32	10	**87.98**
3	0	1.1	0	**53.90**	0	0	0	–	8.79	0	0
4	99.29	95.96	99.52	99.16	99.54	99.91	99.41	–	82.08	**100**	**100**
5	92.19	99.48	95.54	76.92	91.96	76.44	92.55	–	54.78	**100**	**100**
6	84.72	14.00	**92.82**	70.47	67.13	80.04	69.99	–	73.88	86.05	36.80
OA(%) Mean	81.83	65.16	73.76	80.28	76.23	81.47	83.20	–	67.55	88.30	**90.61**
OA Std	3.15	3.00	1.32	5.21	0.00	6.57	7.16	–	6.05	0.19	2.87
NMI Mean	0.7717	0.5685	0.7612	0.8234	0.8200	0.8257	0.7835	–	0.7568	**0.9144**	0.9101
NMI Std	0.0155	0.0529	0.0148	0.0411	0.00	0.0388	0.0831	–	0.0246	0.0044	0.0024
$\mathcal{K}$ Mean	0.7566	0.4885	0.6353	0.7430	0.6963	0.7607	0.7720	–	0.5978	0.8434	**0.8746**
$\mathcal{K}$ Std	0.0423	0.0499	0.0164	0.0694	0.00	0.0811	0.1046	–	0.0690	0.0025	0.0392
Time (sec) Mean	12.81	**5.34**	154.15	211.69	16.99	291.12	81.56	–	267.49	230.38	148.87
Time Std	0.85	0.49	9.40	1.36	0.37	83.83	1.02	–	52.23	7.58	2.99

The best results are highlighted in **bold**, and the second-best results are underlined.

Overall, our method achieves the best performance in OA and $\mathcal{K}$ while maintaining a competitive NMI close to the top-performing MADL. Although N2D conceptually bridges deep and shallow clustering, it performs relatively poorly in this dataset because its clustering is conducted on fixed autoencoded features without feedback to the encoder. The autoencoder and clustering stages are not jointly optimized, so the learned representations are driven purely by reconstruction quality rather than clustering separability. DEKM and SpectralNet exhibit moderate performance but higher variance across trials. However, K-means also performs remarkably well and significantly better than FCM. This can be attributed to the clear boundaries in the Trento data, where hard assignments are more suitable than fuzzy partitioning. Additionally, K-means outperforms FINCH and spectral clustering, suggesting that the Trento dataset has a structure closer to spherical clusters, where centroid-based models are advantageous.

In terms of efficiency, our method requires around 149 s on Trento, which is substantially faster than MADL, DEKM, and IDEC. More importantly, the runtime increases almost proportionally to data volume compared with Houston, where the dataset size is about five times smaller. This consistency demonstrates the linear scalability of the proposed basis-representation formulation with respect to data size while maintaining superior clustering performance.

From the cluster maps, our method best aligns with the ground truth, accurately recognizing the Vineyard and Wood classes, while other methods tend to mix them with surrounding areas. Moreover, our predictions are much smoother, reflecting better spatial consistency. The inclusion of non-local structure preservation also improves the distinction between buildings and roads, further enhancing spatial detail fidelity.

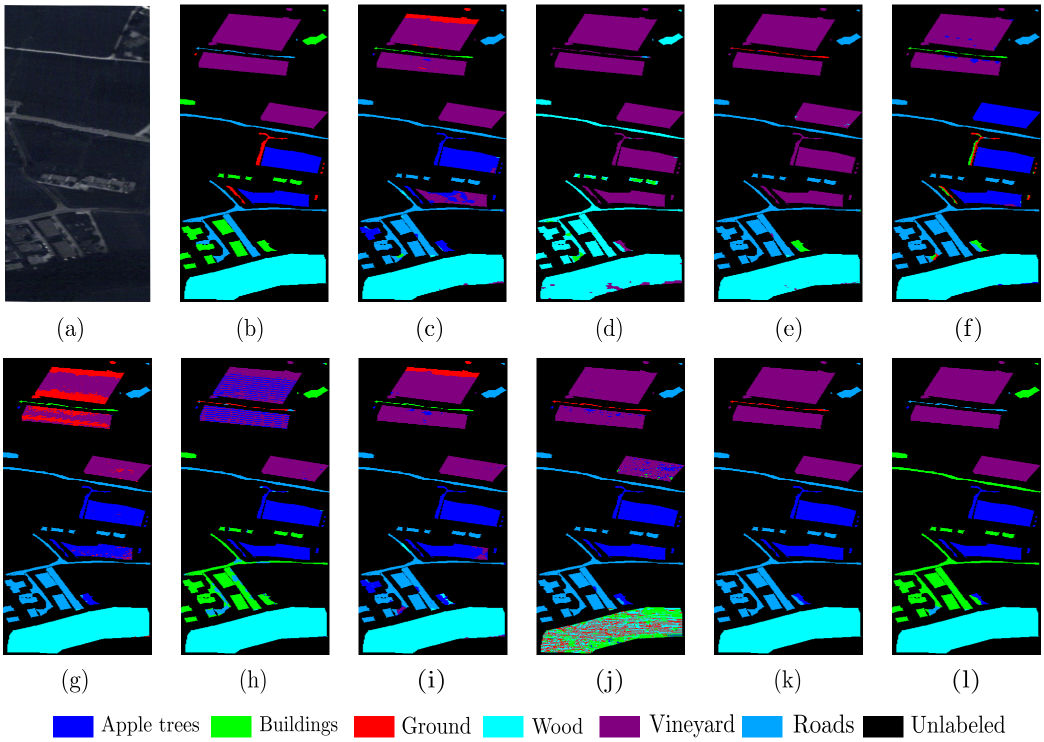


Figure 6. Trento dataset. (**a**) False-color image. (**b**) Ground truth and the clustering results obtained by (**c**) K-means, (**d**) FCM, (**e**) SC, (**f**) IDEC, (**g**) FINCH, (**h**) DEKM, (**i**) SpectralNet, (**j**) N2D, (**k**) MADL, and (**l**) SCDSC.

4.3.3. PaviaU Dataset

The clustering results of various methods on the PaviaU dataset are presented in Table 4 and Figure 7. We set $\beta_1 = 3$, $\beta_2 = 7$, and use a filter window size of 7×7. According to the Wilcoxon signed-rank analysis in Table A1, our method achieves statistically significant improvements ($p_w < 0.05$) over most baselines in terms of OA and $\mathcal{K}$, with higher mean values across both metrics. More importantly, it achieves significant improvements in NMI over all compared methods, showing a clear margin in both mean and stability, which highlights the consistency and reliability of the learned representation even under severe class imbalance.

Table 4. Quantitative evaluation of different clustering methods on the dataset PaviaU. "–" indicates out-of-memory during execution.

Class	K-Means [45]	FCM [46]	SC [47]	IDEC [48]	FINCH [20]	DEKM [49]	SN [30]	HyperAE [42]	N2D [31]	MADL [19]	SCDSC
1	92.44	99.98	**100**	96.53	99.98	77.04	95.17	–	88.66	99.55	93.64
2	47.75	88.99	**100**	38.48	58.88	75.33	63.97	–	32.15	70.83	57.67
3	0	0	0	0	0	10.57	0	–	**95.28**	0	0
4	72.63	0.16	73.76	95.59	89.49	65.43	77.44	–	80.40	97.24	**98.98**
5	66.40	0	39.18	76.68	**100**	93.42	**100**	–	80.00	99.96	**100**
6	3.92	0.02	0	53.16	44.04	22.13	14.94	–	55.30	18.65	**85.46**
7	0	0	0	0	0	**17.32**	0.28	–	15.04	0	0
8	93.69	0	0	99.33	0	82.31	54.66	–	97.11	79.84	**99.98**
9	0	0	76.24	14.20	0	68.00	**94.41**	–	89.50	9.10	42.20
OA(%) Mean	50.97	54.31	67.30	56.11	55.90	64.66	59.89	–	52.88	65.69	**69.48**
OA Std	0.01	0.02	0.00	3.39	0.00	4.94	4.87	–	5.66	3.07	8.02
NMI Mean	0.5926	0.3459	0.6905	0.6722	0.6510	0.6480	0.6408	–	0.6262	0.6511	**0.7490**
NMI Std	0.0001	0.0002	0.00	0.0193	0.00	0.0196	0.0313	–	0.0223	0.0260	0.0358
$\mathcal{K}$ Mean	0.3918	0.3491	0.5290	0.4811	0.4762	0.5408	0.4805	–	0.4551	0.5521	**0.6250**
$\mathcal{K}$ Std	0.0004	0.0003	0.0001	0.0333	0.00	0.0509	0.0625	–	0.0285	0.0385	0.0892
Time (sec) Mean	22.09	**5.34**	297.12	301.53	38.01	839.60	115.77	–	376.29	391.85	320.33
Time Std	0.77	0.49	9.20	3.48	0.64	75.04	0.84	–	54.09	17.96	5.31

The best results are highlighted in **bold**, and the second-best results are underlined.

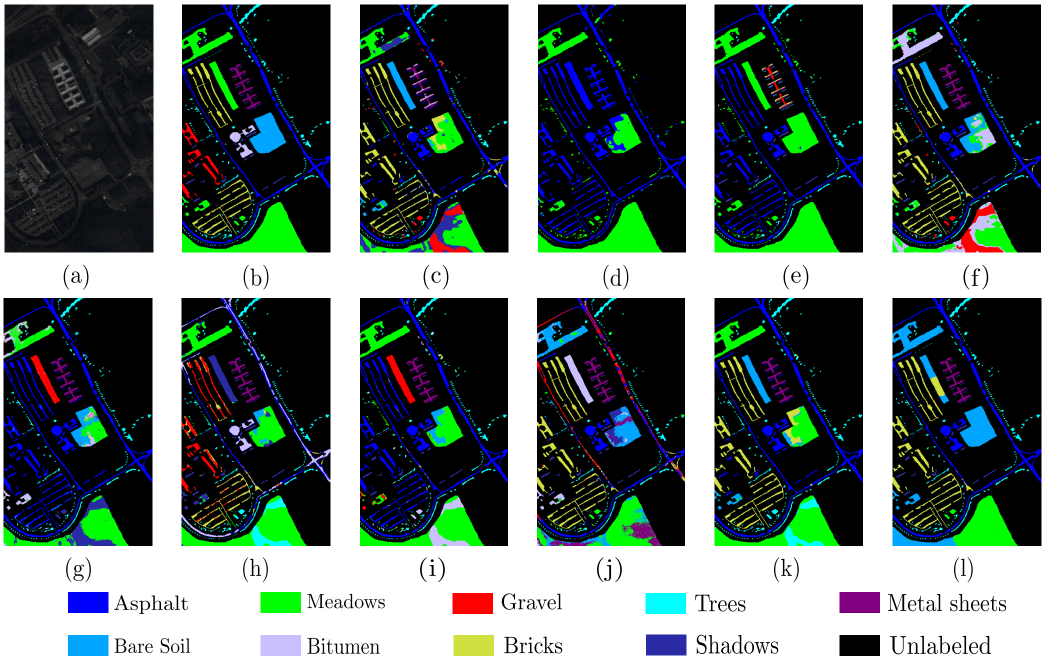


Figure 7. PaviaU dataset. (**a**) False-color image. (**b**) Ground truth and the clustering results obtained by (**c**) K-means, (**d**) FCM, (**e**) SC, (**f**) IDEC, (**g**) FINCH, (**h**) DEKM, (**i**) SpectralNet, (**j**) N2D, (**k**) MADL, and (**l**) SCDSC.

Overall, our method achieves the best mean performance across OA, NMI, and $\mathcal{K}$, with a 3.79% improvement in OA compared with our preliminary work [19]. This demonstrates the effectiveness of the model-aware basis representation and the structure-preserving design in improving both discriminative power and clustering robustness. Due to the dataset's complex and highly mixed structure, all methods achieve OA values below 70%, since accuracy-based metrics are easily affected by dominant-class bias. In contrast, our superior NMI result indicates stronger global consistency and better inter-cluster separation, making it a more reliable indicator on this dataset.

Spectral clustering achieves the second best performance, outperforming several deep methods despite the large data volume. However, due to class imbalance, it tends to detect only a few dominant clusters while ignoring minority ones. Our method also fails to recognize two small classes for the same reason. Among deep models, data-driven approaches such as DEKM, SpectralNet, and N2D show moderate results but less stability. N2D performs particularly poorly since its clustering is conducted on fixed autoencoded features without feedback to the encoder, relying solely on reconstruction rather than clustering optimization. HyperAE again fails due to its $\mathcal{O}(n^2)$ self-representation complexity, whereas our linear basis-representation formulation remains efficient and scalable.

From the cluster maps, our method best aligns with the ground truth, accurately distinguishing between bare soil and tree classes, while other methods often confuse these classes. Benefiting from both local and non-local structure preservation, our results exhibit smoother and more spatially consistent cluster boundaries. In terms of runtime, our method runs faster than other deep clustering baselines such as MADL and DEKM while maintaining a nearly linear growth trend relative to data size, consistent with the observations on the previous datasets.

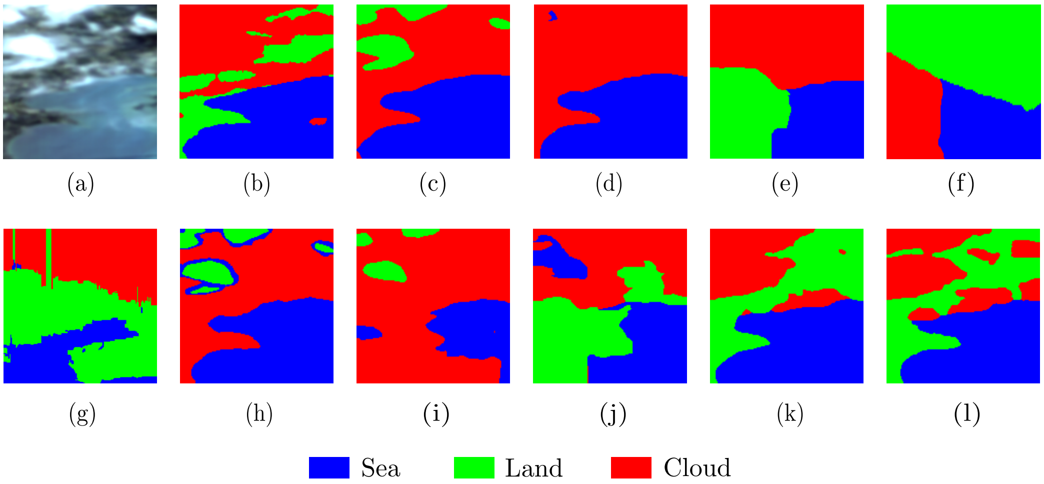
4.3.4. HYPSO-1 Dataset

The clustering results of various methods on the HYPSO-1 dataset are presented in Table 5 and Figure 8. We set $\beta_1 = 3$, $\beta_2 = 0.001$, and use a filter window of size 3×3. We choose a very small weight for the local structure term because the cloud and water regions exhibit relatively simple spatial patterns. Moreover, cloud areas often contain small spots or holes that can be easily removed by spatial filtering, so assigning a large spatial weight may distort these regions rather than preserve meaningful structure. According to the Wilcoxon signed-rank analysis in Table A1, our method achieves statistically significant improvements ($p_w < 0.05$) over all baselines in terms of OA and $\mathcal{K}$, with a noticeably large margin of over 3%. For NMI, although the difference is not statistically significant compared with IDEC and MADL, our method still attains a higher mean value with a low standard deviation of around 0.02. Although this standard deviation is larger than that of MADL, it remains small overall, indicating stable clustering performance.

Table 5. Quantitative evaluation of different clustering methods on the HYPSO-1 dataset. "–" indicates out-of-memory during execution.

Class	K-Means [45]	FCM [46]	SC [47]	IDEC [48]	FINCH [20]	DEKM [49]	SN [30]	HyperAE [42]	N2D [31]	MADL [19]	SCDSC
1	**99.87**	99.01	66.45	88.73	96.98	97.75	42.32	–	74.77	99.13	93.91
2	0.00	0.12	52.92	83.23	66.26	53.21	19.98	–	72.50	92.99	**95.75**
3	69.04	93.65	89.72	64.19	**94.04**	37.06	76.25	–	61.52	60.59	71.83
OA(%) Mean	67.61	77.66	73.64	77.34	75.79	63.59	52.30	–	68.75	81.81	**84.96**
OA Std	0.00	0.02	0.01	6.30	0.00	7.91	6.82	–	10.95	0.41	2.43
NMI Mean	0.5892	0.6102	0.4546	0.6064	0.4662	0.4250	0.2725	–	0.4221	0.6353	**0.6380**
NMI Std	0.00	0.0017	0.00	0.0566	0.00	0.1246	0.0753	–	0.0882	0.0048	0.0195
$\mathcal{K}$ Mean	0.4762	0.6237	0.5931	0.6592	0.5971	0.4430	0.2458	–	0.5308	0.7269	**0.7739**
$\mathcal{K}$ Std	0.00	0.0003	0.00	0.1049	0.00	0.1373	0.1067	–	0.1647	0.0057	0.0340
Time (sec) Mean	**6.88**	30.77	21.35	141.43	16.87	262.37	63.10	–	110.59	145.00	131.10
Time Std	0.37	16.55	1.33	1.11	0.35	48.72	1.93	–	0.55	2.09	3.66

The best results are highlighted in **bold**, and the second-best results are underlined.


(a) (b) (c) (d) (e) (f)

(g) (h) (i) (j) (k) (l)

Figure 8. HYPSO-1 dataset. (**a**) False-color image. (**b**) Ground truth and the clustering results obtained by (**c**) K-means, (**d**) FCM, (**e**) SC, (**f**) IDEC, (**g**) FINCH, (**h**) DEKM, (**i**) SpectralNet, (**j**) N2D, (**k**) MADL, and (**l**) SCDSC.

Overall, our method achieves the best average performance across OA, NMI, and $\mathcal{K}$. We also observe that spectral clustering performs relatively well, while SpectralNet obtains the worst results. This is perhaps because SpectralNet constructs its KNN graph within mini-batches and approximates the spectral embedding using a neural network; such a procedure makes its neighborhood structure unstable on high-dimensional HSI data, resulting in substantially poorer performance than the global, closed-form spectral clustering. The K-means and DEKM methods also fail to achieve good performance, likely because the data distribution deviates from a spherical assumption and the thin cloud and ground regions are highly mixed, leading to ambiguous class boundaries. Consequently, FCM yields noticeably better performance in this scenario. IDEC, MADL, and our method all perform relatively well, suggesting that refined probability estimation can effectively capture the underlying data distribution. Our method further improves performance owing to the more effective modeling of HSI structural characteristics.

From the clustering maps, we can see that, due to the spatially homogeneous nature of regions such as water and cloud, most methods already produce relatively smooth results. Even so, our method aligns with the ground truth most closely, particularly in the ground and cloud regions in the upper-right area. In terms of runtime, our method maintains linear computational complexity and runs faster than IDEC, DEKM, and MADL, demonstrating both its effectiveness and efficiency.

4.4. Ablation Study

In the ablation study, we analyze the effect of different parts of our model. Specifically, we apply the basis-representation clustering model proposed in [18] as the baseline. We then enhance this model by introducing a mini-cluster-based update scheme to preserve the spectral structure. Finally, we integrate a local preservation module to further smooth out noise and improve robustness. The parameter settings of our model are shown in Table 6; the results are presented in Table 7.

Table 6. Hyperparameter settings for each dataset.

Dataset	**Learning Rate**	β_1	β_2	**Smooth Window**
Houston	0.0001	3	8	3×3
Trento	0.005	3	1	7×7
PaviaU	0.005	3	7	7×7
HYPSO-1	0.005	3	0.001	3×3

Table 7. Results of the ablation study.

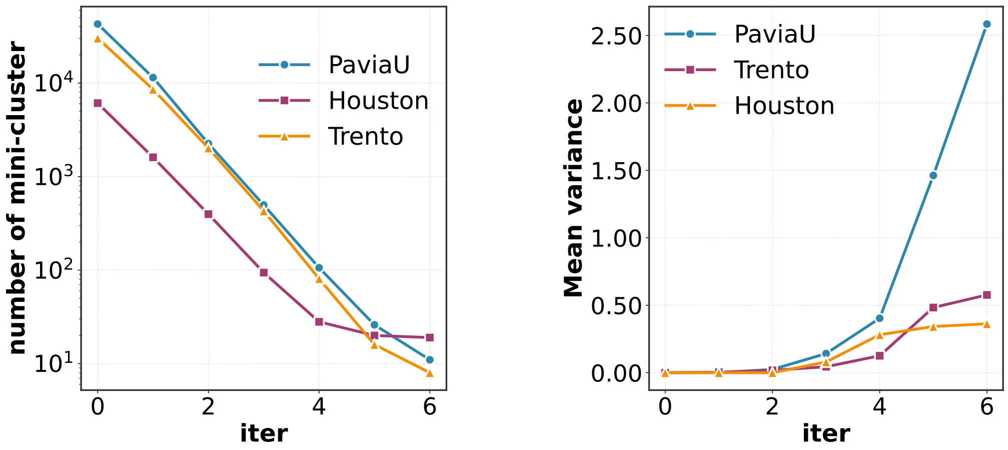
Model	**Houston**			**Trento**			**PaviaU**			**HYPSO-1**		
	OA(%)	**NMI**	$\mathcal{K}$	**OA(%)**	**NMI**	$\mathcal{K}$	**OA(%)**	**NMI**	$\mathcal{K}$	**OA(%)**	**NMI**	$\mathcal{K}$
Baseline	72.53	0.7538	0.6555	81.98	0.8607	0.7701	54.19	0.6340	0.4638	83.72	0.6305	0.7563
L	72.47	0.7619	0.6545	81.85	0.8611	0.7651	60.13	0.6744	0.5282	83.68	0.6302	0.7558
MC	73.92	0.7843	0.6698	90.25	0.8898	0.8704	61.78	0.6743	0.5286	84.89	0.6369	0.7729
MC&L	**74.41**	**0.7902**	**0.6759**	**90.61**	**0.9101**	**0.8746**	**69.48**	**0.7490**	**0.6250**	**84.96**	**0.6380**	**0.7739**

The best results are highlighted in **bold**. Baseline is the original basis representation model proposed in [18]; L is the baseline model with local structure preservation; MC is the baseline model with non-local structure preservation; MC&L is the baseline model with non-local and local structure preservation constraints.

The results from the ablation study indicate that applying only local structure preservation improves NMI on the Houston, Trento, and PaviaU datasets. However, it slightly reduces accuracy on the Houston, Trento, and HYPSO-1 datasets, possibly due to the occasional smoothness in the original predictions. In contrast, applying only non-local structure preservation enhances performance across all datasets and metrics. Combining local and non-local structure preservation further boosts performance, as local structure preservation helps propagate the benefits of non-local preservation. Considering both structures allows for more effective learning of HSI structure compared to using either one alone.

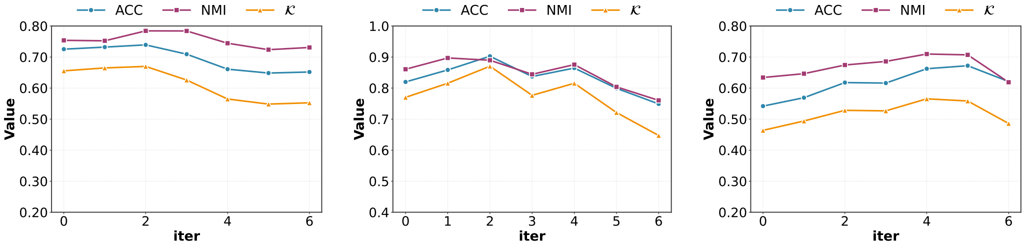
4.4.1. Impact of the Number of FINCH Iterations

In our mini-cluster updating part, the algorithm FINCH [20] is applied for mini-cluster generation. The number of mini-clusters decreases as the number of FINCH iterations increases, as shown in Figure 9a. Meanwhile, the variance within each mini-cluster increases with the number of FINCH iterations. The mean within-cluster variance within every mini-cluster is calculated as shown in Figure 9b. We observe that from the second iteration, the mean within-cluster variance starts to increase, indicating that there are more outliers within the mini-clusters. Specifically, the clustering performance of mini-clusters generated from different iterations is shown in Figure 10. We observe that the accuracy increases at first and then decreases. Specifically, for the Houston and Trento datasets, we have the best performance after two iterations, and then it decreases, which corroborates the findings related to variance. In conclusion, to ensure the quality of the mini-clusters and enhance clustering performance, the mini-clusters generated after the second iteration are applied.



(**a**) (**b**)

Figure 9. Mini-cluster generation across FINCH iterations. (**a**) Mini-cluster number by number of FINCH iterations, (**b**) Mini-cluster variance by number of FINCH iterations.

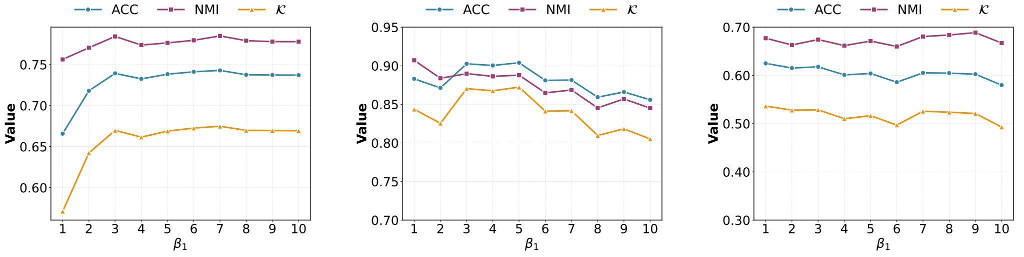


(**a**) (**b**) (**c**)

Figure 10. Performance with mini-clusters generated by the number of FINCH iterations. (**a**) Houston, (**b**) Trento, (**c**) PaviaU.

4.4.2. Impact of Mini-Cluster Updating

The parameter β_1 controls the updating of the mini-cluster soft assignment, which is important in the optimization. Here we compare the performance with different β_1 values on various datasets. The clustering results are shown in Figure 11.



(**a**) (**b**) (**c**)

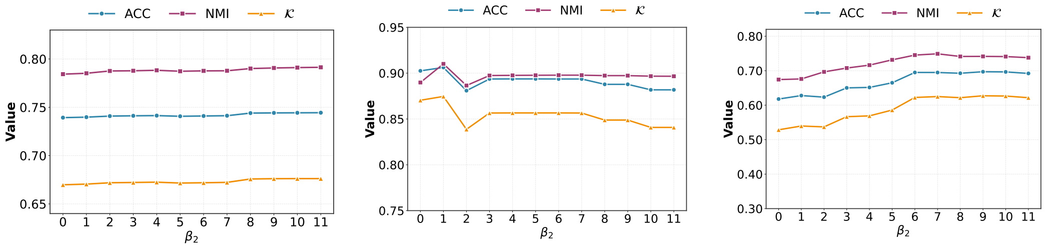
Figure 11. The impact of mini-cluster updating in clustering results. (**a**) Houston, (**b**) Trento, (**c**) PaviaU.

From the graph, it is evident that either too large or too small values of β_1 adversely affect clustering performance. For example, a small β_1 decreases the performance on the Houston dataset. Meanwhile, a large β_1 leads to a lower accuracy on the Trento dataset.

Since different datasets exhibit varying sensitivities to β_1, it is advisable to choose β_1 values between 3 and 5. In our experiments, we set β_1 to three across all datasets.

4.4.3. Impact of Local Structure Preservation

To accommodate the diverse inner local structures of different datasets, distinct settings are required for the local structure preservation module. To evaluate the influence of this module, we integrate it into a mini-cluster optimization model with varying weights. The clustering results with different local structure weights are shown in Figure 12.



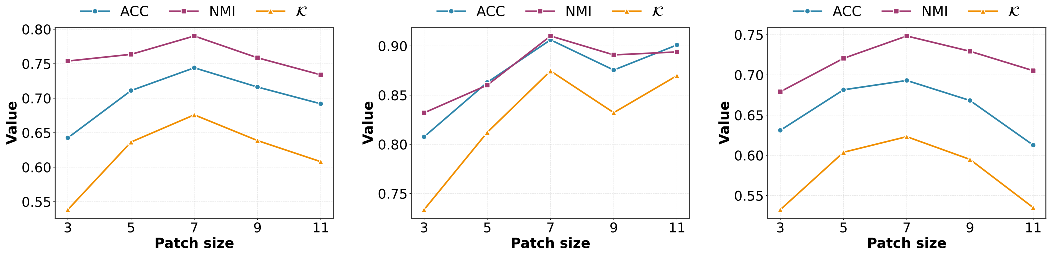
(**a**) (**b**) (**c**)

Figure 12. The impact of local structure preservation in clustering results. (**a**) Houston, (**b**) Trento, (**c**) PaviaU.

The results show that the local structure preservation module effectively enhances clustering performance. Additionally, the optimal weight parameters vary across different datasets. For example, with simpler datasets like Trento, a small weight is adequate to achieve smooth clustering. On the other hand, for more complex datasets like Houston or PaviaU, a larger weight may be needed, typically within the range of [6, 10].

4.4.4. Impact of Patch Size

We conducted an additional ablation study (Figure 13) to analyze the influence of different patch sizes [3, 5, 7, 9, 11]. The results indicate that moderate patch sizes yield better performance, while excessively large patches lead to a decline in performance. Notably, all datasets achieved their best performance when the patch size was set to 7 × 7. This validates the effectiveness of the chosen patch configuration and highlights its role in achieving optimal clustering performance.

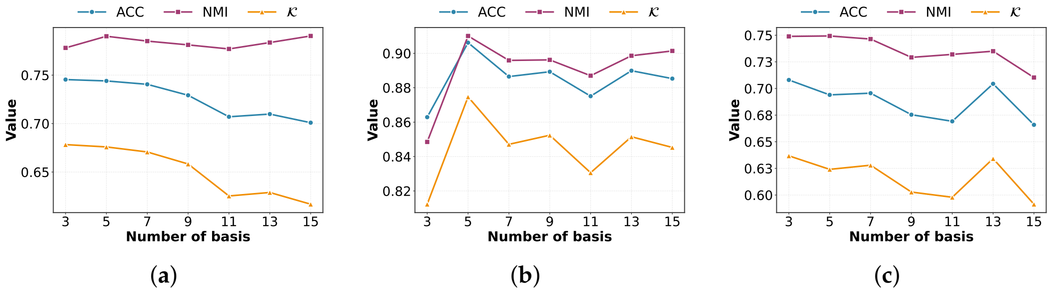


(**a**) (**b**) (**c**)

Figure 13. Impact of patch size on clustering performance across different datasets. (**a**) Houston, (**b**) Trento, (**c**) PaviaU.

4.4.5. Impact of the Number of Basis Vectors

We also conducted an ablation study (Figure 14) to evaluate the impact of different numbers of basis vectors (3, 5, 7, 9, 11, 13, and 15). The results indicate that although the optimal choice varies slightly across datasets, using five basis vectors consistently yields performance close to the best in all cases. This validates the effectiveness of the adopted setting and provides insight into how the basis dimension influences clustering quality.



(**a**) (**b**) (**c**)

Figure 14. Impact of the number of basis components on clustering performance across datasets. (**a**) Houston, (**b**) Trento, (**c**) PaviaU.

4.4.6. Visualization of Embedded Representation and Soft Assignment

The t-distributed Stochastic Neighbor Embedding (t-SNE) [50] visualization of the latent representation for the Houston and PaviaU datasets is shown in Figure 15. We compare the original latent representation produced by the autoencoder with the representation obtained by our proposed method. For the Houston dataset, the t-SNE visualization of the autoencoder's latent representation reveals significant overlap between some classes, resulting in unclear decision boundaries. Additionally, the intra-class points are loosely distributed, lacking tight clustering, which decreases the overall discriminative power. Our proposed method enhances the representation quality, reducing class overlap and improving intra-class compactness. For example, Class 2 is well separated as highlighted in the red circle, leading to a more distinct and organized representation. Meanwhile, the latent representation for the PaviaU dataset from the autoencoder shows unclear decision boundaries, with many classes being mixed. In contrast, our method generates a more discriminative representation, reducing the mixing of different classes, as illustrated by the red circle, ultimately leading to better class separation.

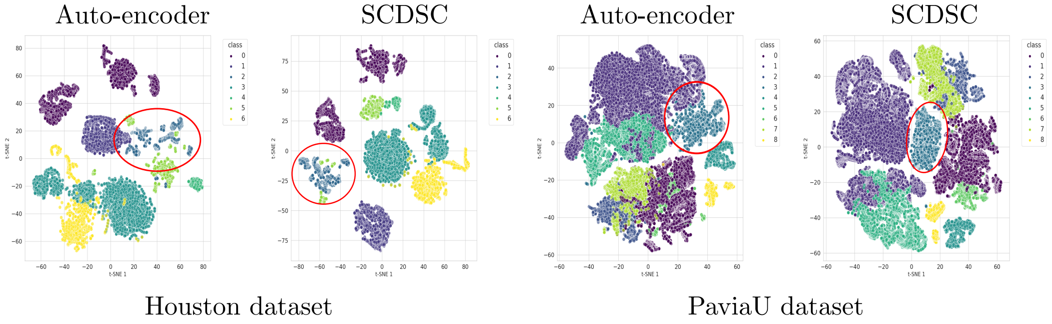
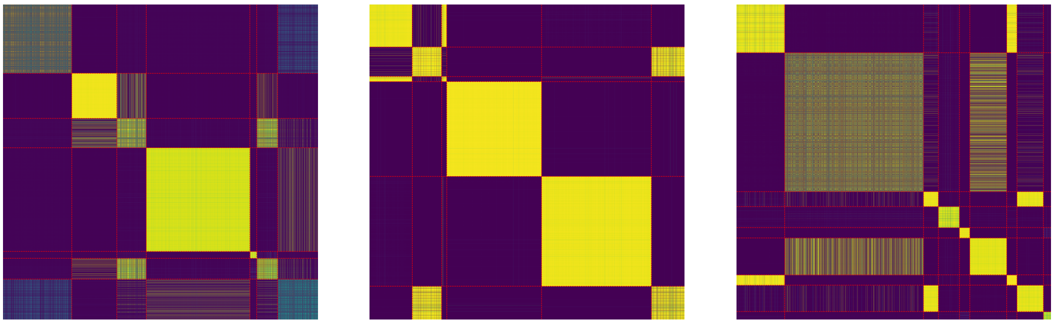


Figure 15. Visualization of the latent representation with t-SNE on the Houston and PaviaU datasets.

We also visualize the confusion matrix of the soft assignment matrix **S**, as shown in Figure 16. Distinct block-diagonal structures can be observed, indicating that the proposed model produces compact clusters with clear inter-cluster boundaries. A small amount of mixing is observed between several clusters, which is reasonable and can be attributed to class imbalance, high spectral similarity, or similar material compositions among adjacent classes.

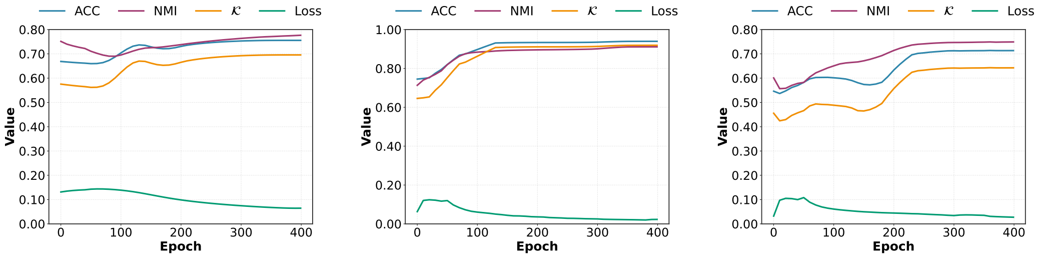


(**a**) (**b**) (**c**)

Figure 16. Visualization of **S** matrix on three datasets. (**a**) Houston, (**b**) Trento, (**c**) PaviaU.

4.4.7. Convergence Analysis

To validate our model's convergence, we show the training curve on the datasets. The results are shown in Figure 17. From these curves, we observe that as the training iterations increase, the training loss decreases while the training accuracy increases, eventually plateauing. This indicates that our model converges well.



(**a**) (**b**) (**c**)

Figure 17. Training loss and accuracy curves on three datasets. (**a**) Houston, (**b**) Trento, (**c**) PaviaU.

5. Conclusions

In this paper, we present a concise review of model-based and deep clustering methods, including both purely data-driven and model-aware approaches. Our primary contribution is a scalable context-preserving model-aware deep clustering approach for hyperspectral images. The proposed method learns the subspace basis under the supervision of both local and non-local structures inherent to hyperspectral image data, allowing these structures to mutually reinforce each other during training. Our approach achieves clustering with a computational complexity of $\mathcal{O}(n)$, making it scalable for large-scale data. Unlike previous state-of-the-art methods, in our method, both the local and non-local structure preservation constraints optimize the entire clustering process in an end-to-end manner and provide stronger guidance for model optimization. Experimental results on four benchmark hyperspectral datasets demonstrate that our method outperforms state-of-the-art approaches in terms of clustering performance.

Author Contributions: Conceptualization, X.L., S.H. and A.P.; methodology, X.L., S.H. and A.P.; software, X.L.; validation, X.L.; formal analysis, X.L., A.P., N.N. and N.D.; investigation, X.L.; data curation, X.L.; writing—original draft preparation, X.L.; writing—review and editing, S.H., N.N., N.D. and A.P.; supervision, A.P. and N.N.; All authors have read and agreed to the published version of the manuscript.

Funding: This research has been partially supported by the Flanders AI Research Programme grant no. 174B09119. This work was also supported by the Bijzonder Onderzoeksfonds (BOF) under Grant BOF.24Y.2021.0049.01, the Research Foundation—Flanders under Grant G094122N (project SPYDER), and the China Scholarship Council under Grant 202106150007. Additionally, this work was supported in part by the National Natural Science Foundation of China under Grant 42301425 and the China Postdoctoral Science Foundation under Grant 2023M743299.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The implementation code and experimental scripts are publicly available at https://github.com/lxlscut/SCDSC (accessed on 10 December 2025). The hyperspectral datasets analyzed in this study are openly available from public repositories as follows: the Trento dataset (https://github.com/A-Piece-Of-Maple/TrentoDateset, accessed on 10 December 2025), the Pavia University dataset on the Hyperspectral Remote Sensing Scenes website (https://www.ehu.eus/ccwintco/index.php/Hyperspectral_Remote_Sensing_Scenes, accessed on 10 December 2025, "Pavia University scene" section), the Houston dataset from the IEEE GRSS Data Fusion Contest (https://www.grss-ieee.org/community/technical-committees/2013-ieee-grss-data-fusion-contest/, accessed on 10 December 2025), and the HYPSO-1 dataset (https://ntnu-smallsat-lab.github.io/hypso1_sea_land_clouds_dataset/, accessed on 10 December 2025).

Acknowledgments: The authors thank the providers of the hyperspectral datasets used in this work. N. Deligiannis is supported by the "Onderzoeksprogramma Artificiele Intelligentie (AI) Vlaanderen" programme and by the ERC Consolidator Grant IONIAN (No. 101171240, DOI: 10.3030/101171240). Funded by the European Union. Views and opinions expressed are however those of the author(s) only and do not necessarily reflect those of the European Union or the European Research Council Executive Agency. Neither the European Union nor the granting authority can be held responsible for them.

Conflicts of Interest: The authors declare no conflicts of interest.

Appendix A. Overall Wilcoxon Signed-Rank Results Across Datasets

We report two-sided Wilcoxon signed-rank p-values (p_w) from paired comparisons of each method against our proposed *SCDSC* baseline (Target) over repeated trials. Smaller p_w indicates stronger evidence of a difference from *SCDSC*.

Table A1. Wilcoxon signed-rank p-values (p_w) versus the proposed SCDSC method. Values are formatted in scientific notation ($p < 0.05$ is marked with *, $p < 0.01$ with **).

Dataset	Metric	K-Means	FCM	SC	IDEC	FINCH	DEKM	SN	HyperAE	N2D	MADL
Houston	OA	$\mathbf{1.95\times10^{-3}}^{**}$	$\mathbf{1.95\times10^{-3}}^{**}$	$\mathbf{5.86\times10^{-3}}^{**}$	$\mathbf{5.86\times10^{-3}}^{**}$	1.05×10^{-1}	$\mathbf{1.95\times10^{-3}}^{**}$	$\mathbf{1.95\times10^{-3}}^{**}$	2.75×10^{-1}	$\mathbf{1.95\times10^{-3}}^{**}$	1.93×10^{-1}
	NMI	$\mathbf{1.95\times10^{-3}}^{**}$	$\mathbf{1.95\times10^{-3}}^{**}$	$\mathbf{1.95\times10^{-3}}^{**}$	6.95×10^{-1}	6.45×10^{-2}	$\mathbf{3.91\times10^{-3}}^{**}$	$\mathbf{1.95\times10^{-3}}^{**}$	3.75×10^{-1}	$\mathbf{3.91\times10^{-3}}^{**}$	$\mathbf{4.88\times10^{-2}}^{*}$
	$\mathcal{K}$	$\mathbf{1.95\times10^{-3}}^{**}$	$\mathbf{1.95\times10^{-3}}^{**}$	$\mathbf{5.86\times10^{-3}}^{**}$	$\mathbf{9.77\times10^{-3}}^{**}$	1.05×10^{-1}	$\mathbf{1.95\times10^{-3}}^{**}$	$\mathbf{1.95\times10^{-3}}^{**}$	2.75×10^{-1}	$\mathbf{3.91\times10^{-3}}^{**}$	1.31×10^{-1}
Trento	OA	$\mathbf{1.95\times10^{-3}}^{**}$	$\mathbf{1.95\times10^{-3}}^{**}$	$\mathbf{1.95\times10^{-3}}^{**}$	$\mathbf{3.91\times10^{-3}}^{**}$	$\mathbf{1.95\times10^{-3}}^{**}$	$\mathbf{3.91\times10^{-3}}^{**}$	$\mathbf{3.91\times10^{-3}}^{**}$	–	$\mathbf{1.95\times10^{-3}}^{**}$	$\mathbf{5.86\times10^{-3}}^{**}$
	NMI	$\mathbf{1.95\times10^{-3}}^{**}$	$\mathbf{1.95\times10^{-3}}^{**}$	$\mathbf{1.95\times10^{-3}}^{**}$	$\mathbf{1.95\times10^{-3}}^{**}$	$\mathbf{1.95\times10^{-3}}^{**}$	$\mathbf{1.95\times10^{-3}}^{**}$	$\mathbf{1.95\times10^{-3}}^{**}$	–	$\mathbf{1.95\times10^{-3}}^{**}$	$\mathbf{5.86\times10^{-3}}^{**}$
	$\mathcal{K}$	$\mathbf{1.95\times10^{-3}}^{**}$	$\mathbf{1.95\times10^{-3}}^{**}$	$\mathbf{1.95\times10^{-3}}^{**}$	$\mathbf{3.91\times10^{-3}}^{**}$	$\mathbf{1.95\times10^{-3}}^{**}$	$\mathbf{9.77\times10^{-3}}^{**}$	$\mathbf{3.91\times10^{-3}}^{**}$	–	$\mathbf{1.95\times10^{-3}}^{**}$	$\mathbf{5.86\times10^{-3}}^{**}$
PaviaU	OA	$\mathbf{1.95\times10^{-3}}^{**}$	$\mathbf{1.95\times10^{-3}}^{**}$	3.22×10^{-1}	$\mathbf{3.91\times10^{-3}}^{**}$	$\mathbf{1.95\times10^{-3}}^{**}$	2.32×10^{-1}	$\mathbf{1.95\times10^{-2}}^{*}$	–	$\mathbf{1.95\times10^{-3}}^{**}$	2.32×10^{-1}
	NMI	$\mathbf{1.95\times10^{-3}}^{**}$	$\mathbf{1.95\times10^{-3}}^{**}$	$\mathbf{1.95\times10^{-3}}^{**}$	$\mathbf{1.95\times10^{-3}}^{**}$	$\mathbf{1.95\times10^{-3}}^{**}$	$\mathbf{1.95\times10^{-3}}^{**}$	$\mathbf{1.95\times10^{-3}}^{**}$	–	$\mathbf{1.95\times10^{-3}}^{**}$	$\mathbf{1.95\times10^{-3}}^{**}$
	$\mathcal{K}$	$\mathbf{1.95\times10^{-3}}^{**}$	$\mathbf{1.95\times10^{-3}}^{**}$	$\mathbf{2.73\times10^{-2}}^{*}$	$\mathbf{3.91\times10^{-3}}^{**}$	$\mathbf{1.95\times10^{-3}}^{**}$	1.05×10^{-1}	$\mathbf{9.77\times10^{-3}}^{**}$	–	$\mathbf{1.95\times10^{-3}}^{**}$	$\mathbf{4.88\times10^{-2}}^{*}$
HYPSO-1	OA	$\mathbf{1.95\times10^{-3}}^{**}$	$\mathbf{1.95\times10^{-3}}^{**}$	$\mathbf{1.95\times10^{-3}}^{**}$	$\mathbf{3.91\times10^{-3}}^{**}$	$\mathbf{1.95\times10^{-3}}^{**}$	$\mathbf{1.95\times10^{-3}}^{**}$	$\mathbf{1.95\times10^{-3}}^{**}$	–	$\mathbf{1.95\times10^{-3}}^{**}$	$\mathbf{1.37\times10^{-2}}^{*}$
	NMI	$\mathbf{1.95\times10^{-3}}^{**}$	$\mathbf{5.86\times10^{-3}}^{**}$	$\mathbf{1.95\times10^{-3}}^{**}$	1.93×10^{-1}	$\mathbf{1.95\times10^{-3}}^{**}$	$\mathbf{1.95\times10^{-3}}^{**}$	$\mathbf{1.95\times10^{-3}}^{**}$	–	$\mathbf{1.95\times10^{-3}}^{**}$	4.92×10^{-1}
	$\mathcal{K}$	$\mathbf{1.95\times10^{-3}}^{**}$	$\mathbf{1.95\times10^{-3}}^{**}$	$\mathbf{1.95\times10^{-3}}^{**}$	$\mathbf{3.91\times10^{-3}}^{**}$	$\mathbf{1.95\times10^{-3}}^{**}$	$\mathbf{1.95\times10^{-3}}^{**}$	$\mathbf{1.95\times10^{-3}}^{**}$	–	$\mathbf{1.95\times10^{-3}}^{**}$	$\mathbf{1.37\times10^{-2}}^{*}$

Significance markers: * $p < 0.05$, ** $p < 0.01$. All significant p_w are shown in **bold**.

References

1. Liu, Y.; Pu, H.; Sun, D.W. Hyperspectral imaging technique for evaluating food quality and safety during various processes: A review of recent applications. *Trends Food Sci. Technol.* **2017**, *69*, 25–35.
2. Stuart, M.B.; McGonigle, A.J.; Willmott, J.R. Hyperspectral imaging in environmental monitoring: A review of recent developments and technological advances in compact field deployable systems. *Sensors* **2019**, *19*, 3071.
3. Briottet, X.; Boucher, Y.; Dimmeler, A.; Malaplate, A.; Cini, A.; Diani, M.; Bekman, H.; Schwering, P.; Skauli, T.; Kasen, I.; et al. Military applications of hyperspectral imagery. In *Proceedings of the Targets and backgrounds XII: Characterization and Representation*; SPIE: Bellingham, WA, USA, 2006; Volume 6239, pp. 82–89.
4. Huang, S.; Zhang, H.; Zeng, H.; Pižurica, A. From Model-Based Optimization Algorithms to Deep Learning Models for Clustering Hyperspectral Images. *Remote Sens.* **2023**, *15*, 2832.

5. Elhamifar, E.; Vidal, R. Sparse subspace clustering: Algorithm, theory, and applications. *IEEE Trans. Pattern Anal. Mach. Intell.* **2013**, *35*, 2765–2781.
6. Vidal, R. Subspace clustering. *IEEE Signal Process. Mag.* **2011**, *28*, 52–68.
7. Liu, G.; Lin, Z.; Yan, S.; Sun, J.; Yu, Y.; Ma, Y. Robust recovery of subspace structures by low-rank representation. *IEEE Trans. Pattern Anal. Mach. Intell.* **2013**, *35*, 171–184.
8. Wang, Y.X.; Xu, H.; Leng, C. Provable subspace clustering: When LRR meets SSC. *Adv. Neural Inf. Process. Syst.* **2013**, *26*, 64–72.
9. Tian, L.; Du, Q.; Kopriva, I. L 0-motivated low rank sparse subspace clustering for hyperspectral imagery. In *Proceedings of the IGARSS 2020-2020 IEEE International Geoscience and Remote Sensing Symposium*; IEEE: New York, NY, USA, 2020; pp. 1038–1041.
10. Ji, P.; Zhang, T.; Li, H.; Salzmann, M.; Reid, I. Deep subspace clustering networks. *Adv. Neural Inf. Process. Syst.* **2017**, *30*, 24–33.
11. Peng, X.; Feng, J.; Zhou, J.T.; Lei, Y.; Yan, S. Deep subspace clustering. *IEEE Trans. Neural Netw. Learn. Syst.* **2020**, *31*, 5509–5521.
12. Lv, J.; Kang, Z.; Lu, X.; Xu, Z. Pseudo-Supervised Deep Subspace Clustering. *IEEE Trans. Image Process.* **2021**, *30*, 5252–5263. https://doi.org/10.1109/TIP.2021.3079800.
13. Huang, S.; Zhang, H.; Pižurica, A. Joint sparsity based sparse subspace clustering for hyperspectral images. In *Proceedings of the 2018 25th IEEE International Conference on Image Processing (ICIP)*; IEEE: New York, NY, USA, 2018; pp. 3878–3882.
14. Li, K.; Qin, Y.; Ling, Q.; Wang, Y.; Lin, Z.; An, W. Self-supervised deep subspace clustering for hyperspectral images with adaptive self-expressive coefficient matrix initialization. *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.* **2021**, *14*, 3215–3227.
15. Zeng, M.; Cai, Y.; Liu, X.; Cai, Z.; Li, X. Spectral-spatial clustering of hyperspectral image based on Laplacian regularized deep subspace clustering. In *Proceedings of the IGARSS 2019-2019 IEEE International Geoscience and Remote Sensing Symposium*; IEEE: New York, NY, USA, 2019; pp. 2694–2697.
16. Valanarasu, J.M.J.; Patel, V.M. Overcomplete deep subspace clustering networks. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*; IEEE: New York, NY, USA, 2021; pp. 746–755.
17. Chen, Z.; Ding, S.; Hou, H. A novel self-attention deep subspace clustering. *Int. J. Mach. Learn. Cybern.* **2021**, *12*, 2377–2387.
18. Cai, J.; Fan, J.; Guo, W.; Wang, S.; Zhang, Y.; Zhang, Z. Efficient Deep Embedded Subspace Clustering. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*; IEEE: New York, NY, USA, 2022, pp. 21–30.
19. Li, X.; Nadisic, N.; Huang, S.; Deligiannis, N.; Pižurica, A. Model-Aware Deep Learning for the Clustering of Hyperspectral Images with Context Preservation. In *Proceedings of the 2023 31st European Signal Processing Conference (EUSIPCO)*; IEEE: New York, NY, USA, 2023; pp. 885–889.
20. Sarfraz, S.; Sharma, V.; Stiefelhagen, R. Efficient parameter-free clustering using first neighbor relations. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*; IEEE: New York, NY, USA, 2019; pp. 8934–8943.
21. Zhang, H.; Zhai, H.; Zhang, L.; Li, P. Spectral–Spatial Sparse Subspace Clustering for Hyperspectral Remote Sensing Images. *IEEE Trans. Geosci. Remote Sens.* **2016**, *54*, 3672–3684. https://doi.org/10.1109/TGRS.2016.2524557.
22. Huang, S.; Zhang, H.; Pižurica, A. Semisupervised Sparse Subspace Clustering Method With a Joint Sparsity Constraint for Hyperspectral Remote Sensing Images. *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.* **2019**, *12*, 989–999. https://doi.org/10.1109/JSTARS.2019.2895508.
23. Xu, J.; Fowler, J.E.; Xiao, L. Hypergraph-Regularized Low-Rank Subspace Clustering Using Superpixels for Unsupervised Spatial–Spectral Hyperspectral Classification. *IEEE Geosci. Remote Sens. Lett.* **2021**, *18*, 871–875. https://doi.org/10.1109/LGRS.2020.2985981.
24. Huang, S.; Zhang, H.; Du, Q.; Pižurica, A. Sketch-based subspace clustering of hyperspectral images. *Remote Sens.* **2020**, *12*, 775.
25. Huang, S.; Zhang, H.; Pižurica, A. Subspace Clustering for Hyperspectral Images via Dictionary Learning With Adaptive Regularization. *IEEE Trans. Geosci. Remote Sens.* **2022**, *60*, 1–17. https://doi.org/10.1109/TGRS.2021.3127536.
26. Zhai, H.; Zhang, H.; Zhang, L.; Li, P. Total variation regularized collaborative representation clustering with a locally adaptive dictionary for hyperspectral imagery. *IEEE Trans. Geosci. Remote Sens.* **2019**, *57*, 166–180.
27. Huang, S.; Zeng, H.; Chen, H.; Zhang, H. Spatial and Cluster Structural Prior-Guided Subspace Clustering for Hyperspectral Image. *IEEE Trans. Geosci. Remote Sens.* **2024**, *62*, 1–15. https://doi.org/10.1109/TGRS.2024.3375922.
28. Huang, P.; Huang, Y.; Wang, W.; Wang, L. Deep embedding network for clustering. In *Proceedings of the 2014 22nd International Conference on Pattern Recognition*; IEEE: New York, NY, USA, 2014; pp. 1532–1537.
29. Peng, X.; Xiao, S.; Feng, J.; Yau, W.Y.; Yi, Z. Deep subspace clustering with sparsity prior. In *Proceedings of the IJCAI*; AAAI Press: Washington, DC, USA, 2016; pp. 1925–1931.
30. Shaham, U.; Stanton, K.; Li, H.; Nadler, B.; Basri, R.; Kluger, Y. SpectralNet: Spectral Clustering Using Deep Neural Networks. *arXiv* **2018**, arXiv:1801.01587.
31. McConville, R.; Santos-Rodríguez, R.; Piechocki, R.J.; Craddock, I. N2D: (Not Too) Deep Clustering via Clustering the Local Manifold of an Autoencoded Embedding. In *Proceedings of the 2020 25th International Conference on Pattern Recognition (ICPR)*; IEEE: New York, NY, USA, 2021; pp. 5145–5152. https://doi.org/10.1109/ICPR48806.2021.9413131.
32. Xie, J.; Girshick, R.; Farhadi, A. Unsupervised deep embedding for clustering analysis. In *Proceedings of the International Conference on Machine Learning*, PMLR: New York, NY, USA, 2016; pp. 478–487.

33. Nalepa, J.; Myller, M.; Imai, Y.; Honda, K.I.; Takeda, T.; Antoniak, M. Unsupervised Segmentation of Hyperspectral Images Using 3-D Convolutional Autoencoders. *IEEE Geosci. Remote Sens. Lett.* **2020**, *17*, 1948–1952. https://doi.org/10.1109/LGRS.2019.2960945.
34. Huang, J.; Gong, S.; Zhu, X. Deep semantic clustering by partition confidence maximisation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, IEEE: New York, NY, USA, 2020; pp. 8849–8858.
35. Zeng, H.; Cao, J.; Feng, K.; Huang, S.; Zhang, H.; Luong, H.; Philips, W. Degradation-Noise-Aware Deep Unfolding Transformer for Hyperspectral Image Denoising. *arXiv* **2023**, arXiv:2305.04047.
36. Chen, X.; Xia, W.; Yang, Z.; Chen, H.; Liu, Y.; Zhou, J.; Wang, Z.; Chen, Y.; Wen, B.; Zhang, Y. SOUL-net: A sparse and low-rank unrolling network for spectral CT image reconstruction. *IEEE Trans. Neural Netw. Learn. Syst.* **2024**, *35*, 18620–18634.
37. Kouni, V.; Paraskevopoulos, G.; Rauhut, H.; Alexandropoulos, G.C. ADMM-DAD net: A deep unfolding network for analysis compressed sensing. In *Proceedings of the ICASSP 2022-2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*; IEEE: New York, NY, USA, 2022; pp. 1506–1510.
38. Wang, J.; Shao, Z.; Huang, X.; Lu, T.; Zhang, R. A deep unfolding method for satellite super resolution. *IEEE Trans. Comput. Imaging* **2022**, *8*, 933–944.
39. Tao, H.; Li, J.; Hua, Z.; Zhang, F. DUDB: Deep unfolding-based dual-branch feature fusion network for pan-sharpening remote sensing images. *IEEE Trans. Geosci. Remote Sens.* **2024**, *62*, 1–17.
40. Zhao, M.; Wang, X.; Chen, J.; Chen, W. A plug-and-play priors framework for hyperspectral unmixing. *IEEE Trans. Geosci. Remote Sens.* **2022**, *60*, 1–13.
41. Cai, Y.; Zeng, M.; Cai, Z.; Liu, X.; Zhang, Z. Graph regularized residual subspace clustering network for hyperspectral image clustering. *Inf. Sci.* **2021**, *578*, 85–101.
42. Cai, Y.; Zhang, Z.; Cai, Z.; Liu, X.; Jiang, X. Hypergraph-structured autoencoder for unsupervised and semisupervised classification of hyperspectral image. *IEEE Geosci. Remote Sens. Lett.* **2022**, *19*, 1–5.
43. Liu, S.; Huang, N.; Xiao, L. Locally Constrained Collaborative Representation Based Fisher's LDA for Clustering of Hyperspectral Images. In *Proceedings of the IGARSS 2020-2020 IEEE International Geoscience and Remote Sensing Symposium*; IEEE: New York, NY, USA, 2020; pp. 1046–1049.
44. Cai, Y.; Zhang, Z.; Ghamisi, P.; Ding, Y.; Liu, X.; Cai, Z.; Gloaguen, R. Superpixel contracted neighborhood contrastive subspace clustering network for hyperspectral images. *IEEE Trans. Geosci. Remote Sens.* **2022**, *60*, 1–13.
45. Hartigan, J.A.; Wong, M.A. Algorithm AS 136: A k-means clustering algorithm. *J. R. Stat. Soc. Ser. C (Appl. Stat.)* **1979**, *28*, 100–108.
46. Bezdek, J.C. *Pattern Recognition with Fuzzy Objective Function Algorithms*; Springer Science & Business Media: Berlin/Heidelberg, Germany, 2013.
47. Von Luxburg, U. A tutorial on spectral clustering. *Stat. Comput.* **2007**, *17*, 395–416.
48. Guo, X.; Gao, L.; Liu, X.; Yin, J. Improved deep embedded clustering with local structure preservation. In *Proceedings of the IJCAI*; AAAI Press: Washington, DC, USA, 2017; Volume 17, pp. 1753–1759.
49. Guo, W.; Lin, K.; Ye, W. Deep Embedded K-Means Clustering. In *Proceedings of the 2021 International Conference on Data Mining Workshops (ICDMW)*, IEEE: New York, NY, USA, 2021; pp. 686–694. https://doi.org/10.1109/ICDMW53433.2021.00090.
50. Van der Maaten, L.; Hinton, G. Visualizing data using t-SNE. *J. Mach. Learn. Res.* **2008**, *9*, 2579–2605.

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.