Bipartite Graph-Based Projected Clustering With Local Region Guidance for Hyperspectral Imagery

Yongshan Zhang, *Member, IEEE*, Guozhu Jiang, Zhihua Cai, and Yicong Zhou, *Senior Member, IEEE*

***Abstract*—Hyperspectral image (HSI) clustering is challenging to divide all pixels into different clusters because of the absent labels, large spectral variability and complex spatial distribution. Anchor strategy provides an attractive solution to the computational bottleneck of graph-based clustering for large HSIs. However, most existing methods require separated learning procedures and ignore noisy as well as spatial information. In this paper, we propose a bipartite graph-based projected clustering (BGPC) method with local region guidance for HSI data. To take full advantage of spatial information, HSI denoising to alleviate noise interference and anchor initialization to construct bipartite graph are conducted within each generated superpixel. With the denoised pixels and initial anchors, projection learning and structured bipartite graph learning are simultaneously performed in a one-step learning model with connectivity constraint to directly provide clustering results. An alternating optimization algorithm is devised to solve the formulated model. The advantage of BGPC is the joint learning of projection and bipartite graph with local region guidance to exploit spatial information and linear time complexity to lessen computational burden. Extensive experiments demonstrate the superiority of the proposed BGPC over the state-of-the-art HSI clustering methods.**

***Index Terms*—Hyperspectral imagery, bipartite graph, projected clustering, superpixel segmentation.**

I. Introduction

HYPERSPECTRAL imaging technique is well developed to record the scenes from observation areas. Different from RGB images with three channels [1], hyperspectral images (HSIs) obtained by hyperspectral sensors are with hundreds of spectral signatures [2], [3]. They are always denoted by three-order tensors to represent the detailed observations in two spatial dimensions by using numerous wavelengths in one spectral dimension. It is reasonable to analyze HSIs with abundant spectral and spatial information to facilitate diverse applications for precision agriculture, urban planning and military defense. Among diverse applications, the recognition of land-cover materials is a crucial learning task to identify human behaviours in a natural environment [4], [5], [6].

Based on the learning scheme with labels or not, the recognition of land-cover materials can be partitioned into HSI classification and clustering. As a supervised learning scheme, HSI classification develops learning models with labelled training pixels and assigns an exclusive label to every testing pixel [7], [8], [9]. By contrast, as an unsupervised learning scheme, HSI clustering develops learning models with unlabelled pixels and tries to partition all unlabelled pixels into different clusters without any prior knowledge [10], [11], [12]. In reality, it is labor-intensive and time-consuming for HSI annotation. Besides, HSI analysis without label guidance is very difficult because of the large spectral variability and complex spatial structure. Based on these characteristics, HSI clustering is a challenging task in remote sensing community.

According to the learning mechanism, HSI clustering can be separated into centroid-based, density-based, biology-based and graph-based methods [13]. Assume that pixels are distributed in a sphere structure, centroid-based methods obtain the optimal clusters by finding the minimum distance between pixels and cluster centers. Typical methods include K-means [14], fuzzy C-means (FCM) [15] and their variants. Based on the density distribution, density-based methods identify different clusters with high density that are partitioned by pixels with low density. Representative methods are k nearest neighbor density clustering (KNNCLUST) [16] and density peak clustering (DPC) [17]. Inspired by biological theory, biology-based methods pursue the optimal partition via adaptive evolutionary computation in a heuristic manner. Typical examples are multiobjective particle swarm optimization (MOPSO) [18] and unsupervised artificial immune classifier (UAIC) [19]. The above three types of methods generally utilize spectral signatures to generate clusters without considering auxiliary spatial and structural information. To solve this issue, graph-based methods explore the relationships between pair-wise pixels with a weighted graph and then conduct partition on the graph [20], [21]. After graph-cut learning, pixels are divided into disjoint subsets according to their similarities.

Graph-based clustering is able to provide competitive results for HSIs. As one of the promising styles, subspace clustering constructs a favorable affinity matrix to capture the geometric structure of data with their self-expressive relationships [22], [23], [24]. In [22], Zhai et al. proposed an ℓ_2-norm

Manuscript received 3 April 2023; revised 25 March 2024; accepted 20 April 2024. Date of publication 30 April 2024; date of current version 18 September 2024. This work was supported in part by the National Natural Science Foundation of China under Grant 62106241, in part by the Knowledge Innovation Program of Wuhan-Shuguang under Grant 2023010201020335, in part by Science and Technology Development Fund, Macau SAR File no. 0049/2022/A1, and in part by the University of Macau File no. MYRG2022-00072-FST. The associate editor coordinating the review of this manuscript and approving it for publication was Prof. Sanghoon Lee. *(Corresponding author: Yicong Zhou.)*

Yongshan Zhang, Guozhu Jiang, and Zhihua Cai are with the School of Computer Science, China University of Geosciences, Wuhan 430074, China (e-mail: yszhang.cug@gmail.com; jiangguozhu@cug.edu.cn; zhcai@cug.edu.cn).

Yicong Zhou is with the Department of Computer and Information Science, University of Macau, Macau 999078, China (e-mail: yicongzhou@um.edu.mo).

The source code is released at https://github.com/ZhangYongshan/BGPC.

Digital Object Identifier 10.1109/TMM.2024.3394975

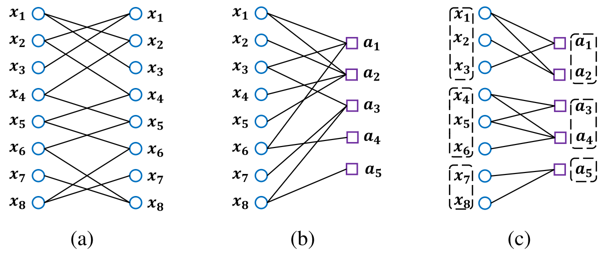


Fig. 1. Different types of graph representations. A circle denotes a sample and a square indicates an anchor. (a) Traditional graph. (b) Bipartite graph. (c) Structured bipartite graph with three connected components.

regularized sparse subspace clustering (L2-SSC) method by introducing spatial information. Further, Huang et al. [23] presented an adaptive regularized dictionary learning subspace clustering (IDLSC) method to learn a concise dictionary from HSIs. Based on structural information, Cai et al. [24] designed two graph convolutional subspace clustering (GCSC) methods in the original feature space and kernel space. In addition, deep subspace clustering methods are emerging by introducing different components or strategies into autoencoder, such as contrastive learning [25], low-rank graph convolution [26], collaborative constrained multi-scale learning [27]. Given a dataset with N samples and B features, these methods take $\mathcal{O}(N^2)$ for storage and require $\mathcal{O}(N^2B)$ for computation to obtain the affinity matrix. To identify C different partitions on an $N \times N$ graph shown in Fig. 1(a), they need at least $\mathcal{O}(N^2C)$ or $\mathcal{O}(N^3)$ for computation based on the eigen decomposition of the related Laplacian matrix. Although these methods achieves promising results, they still suffer from both heavy computation burden and huge memory consumption when solving the problems with large-scale data.

To deal with this dilemma, efficient graph-based clustering has been put forward in recent years. With different strategies, efficient methods show their advantages in both time complexity and memory requirement [28], [29], [30]. In [28], Cai et al. proposed a neighborhood contrastive subspace clustering (NCSC) network with superpixel pooling autoencoder to learn superpixel-level representations for clustering. To leverage the geometric structures, Zhao et al. [31] presented a global and local similarity graph-based clustering (SGLSC) method by introducing both global and local information at superpixel level. Even though these superpixel-level methods achieve superior performance at certain datsets, they heavily rely on the generated superpixels because pixel's label assignment is achieved with pixel-to-superpixel relationships. To perform efficient clustering at pixel level, anchor strategy shown in Fig. 1(b) is applied and takes the linear time complexity of $\mathcal{O}(NMB)$, where M is the number of anchors and $M \ll N$. For instance, in [29], Wang et al. proposed a fast spectral clustering with anchor graph (FSCAG) method based on the original pixels and the denoised pixels from mean filtering. Further, Wang et al. [32] presented a spatial-spectral clustering with anchor graph (SSCAG) method based on the original pixels and the denoised pixels from multi-scale weighted mean filtering. These methods first learn the graph between pixels and anchors and then separately perform clustering with spectral analysis. Their performance is compromised because of the separated graph learning and clustering. To solve this problem, in [33], Yang et al. presented a fast spectral embedded clustering with structured graph learning (FSECSGL) method to directly output clustering results from the embedded data. To assign the soft labels for pixels, Yang et al. [30] further designed a fuzzy embedded clustering with bipartite graph (FECBG) method to obtain the clustering outcome by fuzzy clustering. These methods first learn low-dimensional representations and then directly provide clustering results based on graph learning. However, they ignore spatial information and separate feature learning from clustering. Based on these observations, it is challenging to simultaneously perform efficient projection and clustering at pixel level with spatial information in a one-step learning model.

In this paper, we propose a bipartite graph-based projected clustering (BGPC) method with local region guidance for HSI data. Based on the spatial structure, the HSI is partitioned into multiple superpixels to reflect homogenous regions. To make good use of spatial information, HSI denoising is achieved with the pixel reconstruction by a weighted summation of nearest neighboring pixels and anchor initialization is realized with the average of all denoised pixels within each superpixel. Compared to denoising across the entire HSI, the presented spatial denoising strategy constrains the selection range of neighbors from the same superpixel with similar properties, thereby better preserving the edge information and mitigating its deterioration. Inspired by anchor strategy, projection learning and structured bipartite graph learning are simultaneously conducted in a unified model with connectivity constraint to achieve efficient HSI clustering at pixel level. An alternating optimization algorithm is devised to solve the formulated problem. As shown in Fig. 1(c), the structured bipartite graph with exactly connected components is learned to directly provide clustering results after one-step learning in linear time complexity. Extensive experiments demonstrate the superiority of our BGPC over the state-of-the-art HSI clustering methods.

The contributions of this paper are highlighted as follows.

1) We propose a BGPC method for efficient HSI clustering. With local region guidance, projection learning and structured bipartite graph learning are simultaneously conducted to directly obtain optimal clustering with exact pixel partition in a one-step learning model.
2) We introduce an effective denoising manner to reduce noise interference and a useful anchor selection strategy to cover all hyperspectral pixels to facilitate efficient clustering. They take full advantage of plentiful spatial information within the generated superpixels.
3) We present an alternating optimization strategy to solve the proposed formulation and show its high efficiency with linear time complexity.

The rest of this paper is structured as follows. Section II introduces the related work. Section III presents the proposed BGPC method for HSI clustering. Section IV describes the optimization strategy and analyzes the computational complexity. Extensive experiments and comprehensive comparisons on HSI datasets are provided in Section V. Finally, Section VI concludes the paper.

II. Related Works

A. Traditional Graph-Based Clustering

Traditional graph-based clustering methods are effective for HSI data by exploring the relationships between pair-wise pixels [20], [21], [34]. Their performance depends on the quality of the learned affinity matrix. In [20], Zhang et al. proposed a spectral-spatial sparse subspace clustering method to learn a more favorable affinity matrix from HSIs. Based on the partitioned spectral bands and spatial features, Tian et al. [21] presented a spatial-spectral multi-view low-rank sparse subspace clustering method to analyze HSI data. To explore the nonlinear relationships, Li et al. [35] devised a deep mutual information subspace clustering network by means of information and deep learning theory. Apart from the graph-based methods in a subspace clustering style, there are other effective methods in different styles. For example, Murphy et al. [34] proposed a spectral-spatial diffusion learning method to integrate geometric estimation with spectral and spatial information. To perform HSI clustering, Kong et al. [36] desgined an unsupervised broad learning method with a graph-regularized autoencoder.

Even though traditional graph-based methods yield superior results for HSIs, they always ask for at least the quadratic time complexity and memory requirement of $\mathcal{O}(N^2)$. Thus, they are inapplicable to the problems with large-scale data.

B. Efficient Graph-Based Clustering

Efficient graph-based clustering methods are emerging to solve the bottleneck of traditional graph-based clustering [37], [38], [39], [40]. They are superior in not only clustering performance but also time complexity and memory consumption. In [41], Ding et al. presented a graph convolution embedding clustering network to reduce the learning scale to superpixel level. Based on tree node partition and binary clustering, Shahi et al. [38] proposed a hierarchical clustering method to efficiently analyze HSI data at different levels. Anchor strategy provides a practical solution to learn clustering results at pixel level. To handle large HSIs, Hassanzadeh et al. [39] devised a sequential spectral clustering method with sequential singular value decomposition and bipartite graph represention. Utilizing both spectral and spatial information, Huang et al. [40] presented a coclustering method based on bipartite graph partition with joint sparsity over dictionary learning. Based on Nyström extension, Zhao et al. [42] designed a fast spectral clustering method with the selected anchors. Without relying on additional clustering, Wang et al. [37] proposed a scalable clustering method with anchor graph and nonnegative relaxation.

Although efficient graph-based clustering methods with different strategies are flexible for large HSIs, they still show the deficiency in some aspects, such as clustering with pixel-to-superpixel memberships, neglecting spatial information and separated learning procedures. To address the above-mentioned problems, our proposed method belonging to efficient graph-based clustering can simultaneously perform projection and clustering at pixel level with spatial information in a one-step learning model.

III. Proposed Method

A. Overview

The overall framework of the proposed BGPC is illustrated in the Fig. 2. Based on the spatial distribution, the HSI is segmentated into diverse superpixels with homogeneity. To alleviate noise interference, HSI denoising is achieved by reconstructing pixels with a weighted summation of nearest neighboring pixels within a same superpixel. To learn a bipartite graph, anchor initialization is realized by averaging all denoised pixels within each superpixel. For effective and efficient HSI clustering at pixel level, projection learning and bipartite graph learning are simultaneously conducted using the denoised pixels and initial anchors in a unified optimization model with connectivity constraint. With local region guidance, spatial information can be well exploited during clustering. After iterative optimization, clustering results can be directly provided for all pixels according to the exactly connected components in the learned bipartlte graph.

B. Weighted Local Region Denoising

Due to the wicked observation condition and water absorption, noise interference may be introduced during HSI data collection [29]. It is inevitable that the introduced noises could lead to spectral distortion and degraded performance in subsequent learning [32]. To eliminate the noises and improve the representation capability, it is reasonable to perform HSI denoising with spatial information in local regions. Here, we present a weighted local region denoising strategy containing local region generation and weighted spatial denoising.

1) Local Region Generation: In general, pixels with a same label are always distributed in a homogenous region with high probability [43]. Recent studies have shown that superpixel segmentation technique is effective to uncover the spectral-spatial structures of HSIs [44]. It is better to localize the homogeneity of HSI data for further analysis. Among many effective segmentation methods, we adopt the entropy rate superpixel (ERS) segmentation method [45] to adaptively generate homogenous regions with adaptive shapes and sizes based on the complex spatial distribution and texture complexity. The ERS segmentation method is originally designed for RGB image segmentation. During the segmentation, ERS firstly converts RGB images to grayscale images, and then conducts segmentation on the grayscale images. That is to say, segmentation using ERS is conducted on the image with one component instead of more components. When there are more components within an HSI, it is necessary to get the only one component before segmentation. Therefore, we only use the first principal component with the primary knowledge of the HSI for segmentation. Assume that an HSI is denoted by a third-order tensor $\boldsymbol{\mathcal{X}} \in \mathbb{R}^{W \times H \times B}$ with $W \times H$ pixels and B bands, we reshape it into a matrix denoted by $\widehat{\mathbf{X}} \in \mathbb{R}^{B \times N}$ $(N = W \times H)$. To achieve efficient HSI segmentation, we first conduct principal component analysis

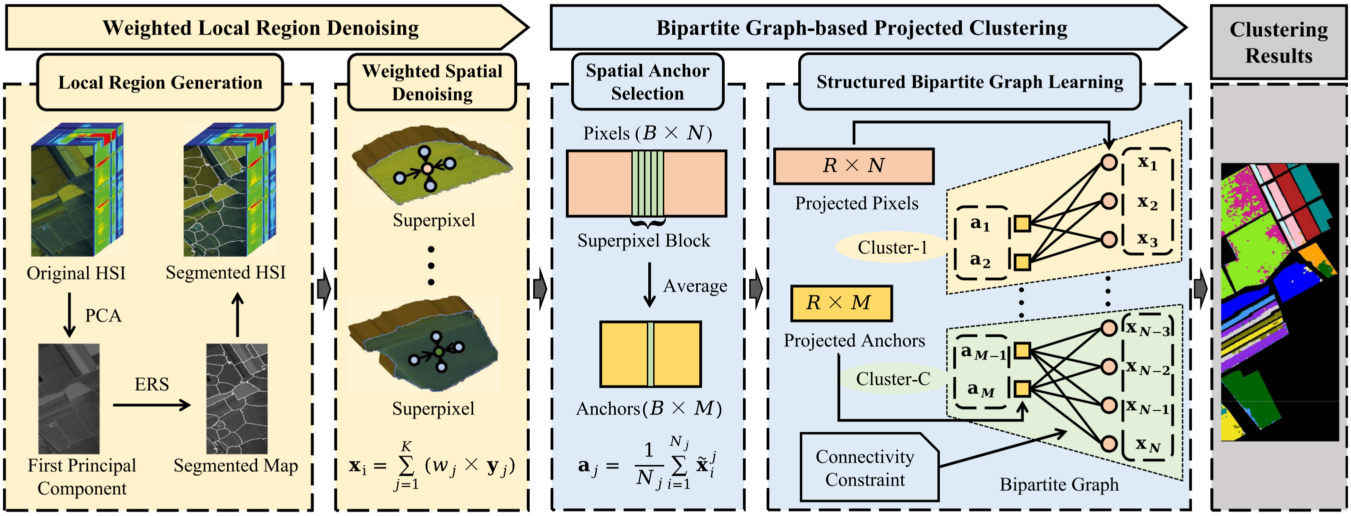


Fig. 2. Overall framework of the proposed BGPC for HSI clustering. It performs superpixel segmentation on the HSI to obtain the segmented HSI. Based on the generated superpixels, HSI denoising is first conducted and anchor initialization is then determined with spatial information guidance. Given the denoised pixels and initial anchors, projection learning and structured bipartite graph learning are simultaneously conducted in a unified model with connectivity constraint to directly provide clustering results after iterative optimization.

(PCA) [46] on $\widehat{\mathbf{X}}$ to describe major information on the first principal component I_f, and then perform ERS on I_f to generate M homogenous regions as follows:

$$I_f = \bigcup_{i=1}^{M} \mathcal{H}_i, \quad \text{s.t.} \quad \mathcal{H}_i \cap \mathcal{H}_j = \varnothing, \quad (i \neq j), \tag{1}$$

where $\mathcal{H}_i$ is the i-th local region with the same segmentation label. The constraint $\mathcal{H}_i \cap \mathcal{H}_j = \varnothing$ means that any two generated local regions are non-overlapping. Aligning the above segmentation results to original HSI, we can acquire the segmented HSI with all spectral bands to reflect multiple superpixels.

2) Weighted Spatial Denoising: Denoising is an effective manner to reduce noise interference. It is harmful to directly perform denoising on the entire HSI because the misleading information of different land covers may be introduced into the denoised pixels. For HSIs, mean filtering is a traditional strategy to perform denoising by averaging pixels within a small square window [29]. It is obvious that such manner does not work well because the complex spatial structures cannot be effectively extracted by a square window. Based on the homogeneity of HSIs, the pixels within a same superpixel exhibit similar property. Compared to denoising on the entire HSI and denoising within a square window, it is more effective to perform denoising within a superpixel based on the adaptive local region information. It can smooth the homogenous regions and preserve their edge information for HSIs. Given an original pixel $\widehat{\mathbf{x}}_i \in \mathbb{R}^{B\times 1}$ within $\widehat{\mathbf{X}}$, we confirm its K nearest spatial neighbors within the same superpixel and denoted them by $\mathbf{Y}_i = \{\mathbf{y}_1, \mathbf{y}_2, \ldots, \mathbf{y}_K\} \in \mathbb{R}^{B\times K}$. To perform effective denoising, we first calculate the weight to measure the spectral similarity between $\widehat{\mathbf{x}}_i$ and $\mathbf{y}_j$ in $\mathbf{Y}_i$ by

$$w_j = \frac{\exp\left(-\|\widehat{\mathbf{x}}_i - \mathbf{y}_j\|_2^2/2t^2\right)}{h}, \quad j = 1, 2, \ldots, K, \tag{2}$$

where the two control parameters are empirically determined by $t = (1/K)\sum_{j=1}^{K}\|\widehat{\mathbf{x}}_i - \mathbf{y}_j\|_2^2$ and $h = \sum_{j=1}^{K}\exp(-\|\widehat{\mathbf{x}}_i - \mathbf{y}_j\|_2^2/2t^2)$. Based on the reconstruction weights learned by (2), we perform denoising for $\widehat{\mathbf{x}}_i$ with its K nearest spatial neighbors by

$$\mathbf{x}_i = \sum_{j=1}^{K} w_j \times \mathbf{y}_j, \; i = 1, 2, \ldots, N, \tag{3}$$

where $\mathbf{x}_i$ is the denoised and reconstructed pixel of $\widehat{\mathbf{x}}_i$ by a weighted summation of the K nearest spatial neighbors in a same superpixel. After denoising, the spatial distribution of pixels is in a good consistency and the representation capability of data is greatly improved. Thus, we utilize the denoised data $\mathbf{X} = \{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_N\} \in \mathbb{R}^{B\times N}$ instead of the original data $\widehat{\mathbf{X}}$ to pursue better performance in subsequent learning.

C. Bipartite Graph-Based Projected Clustering

Traditional graph-based clustering methods are effective for HSI analysis [22]. However, most of them are inapplicable to large HSI data due to the high dimensionality and computation burden [24]. To solve these problems, we formulate a bipartite graph-based projected clustering model with local region guidance. It can directly learn a much smaller structured graph for efficient and effective HSI clustering based on spatial anchor selection and structured bipartite graph learning.

1) Spatial Anchor Selection: For anchor-based methods, it is vital to select a smaller number of anchors to uncover the distribution of all samples [47]. The random and K-means strategies are commonly used for anchor selection [37]. The random strategy arbitrarily selects some original samples to be anchors, while the K-means strategy empirically learns the cluster centroids as representative anchors. In reality, the two strategies are unable to select high-quality anchors for HSIs, because they ignore the inherent spatial distribution of pixels. To solve this issue, we select informative anchors for the HSI based on the denoised superpixels. By doing so, anchors generated from the corresponding superpixels contain spatial information and are able to cover all pixels in the entire HSI, thus leading to better results. Specifically, we denote an anchor matrix containing a set of anchors by $\mathbf{A} = \{\mathbf{a}_1, \mathbf{a}_2, \dots, \mathbf{a}_M\} \in \mathbb{R}^{B\times M}$. Given the denoised superpixels, we determine M anchors by averaging all denoised pixels in the same superpixel as follows:

$$\mathbf{a}_j = \frac{1}{N_j}\sum_{i=1}^{N_j} \tilde{\mathbf{x}}_i^j, \ \ j = 1, 2, \dots, M, \tag{4}$$

where $\tilde{\mathbf{x}}_i^j$ denotes the i-th pixel in the j-th superpixel and N_j indicates the number of pixels within the j-th superpixel. Thus, $N = \sum_{j=1}^{M} N_j$. Compared to the random and K-means strategies, (4) provides an advanced strategy to determine informative anchors with spatial information based on the generated superpixels.

2) Structured Bipartite Graph Learning: Bipartite graph-based clustering is a promising learning paradigm to partition samples into a group of clusters in an efficient way [47]. Compared to traditional graph-based clustering required the time complexity of $\mathcal{O}(N^2)$ or $\mathcal{O}(N^3)$, bipartite graph-based clustering is more efficient with the time complexity of $\mathcal{O}(N)$. Thus, it is more applicable and flexible for large-scale HSI data. To learn a bipartite graph $\mathbf{Z} \in \mathbb{R}^{N\times M}$ for HSIs, the i-th denoised pixel $\mathbf{x}_i \in \mathbf{X}$ should be connected to the j-th anchor $\mathbf{a}_j \in \mathbf{A}$ with a probability z_{ij} to describe their "pixel-to-anchor" neighboring relationship. Specifically, z_{ij} is an element in $\mathbf{Z}$ to denote the connection probability between $\mathbf{x}_i$ and $\mathbf{a}_j$. It is common knowledge that the greater value of z_{ij} corresponds to the smaller distance between $\mathbf{x}_i$ and $\mathbf{a}_j$. Thus, bipartite graph learning for $\mathbf{Z}$ can be achieved by adaptive assigning pixels to neighboring anchors as follows:

$$\begin{aligned} &\min_{\mathbf{A},\mathbf{Z}} \sum_{i=1}^{N}\sum_{j=1}^{M} \|\mathbf{x}_i - \mathbf{a}_j\|_2^2 \, z_{ij} + \gamma \|\mathbf{Z}\|_F^2 \\ &\text{s.t.} \quad \mathbf{z}_i^T \mathbf{1} = 1, z_{ij} \geq 0, \end{aligned} \tag{5}$$

where $\|\mathbf{x}_i - \mathbf{a}_j\|_2^2$ represents the square of Euclidean distance between $\mathbf{x}_i$ and $\mathbf{a}_j$, $\|\mathbf{Z}\|_F^2$ indicates the regularization term for $\mathbf{Z}$, and γ is an adaptive regularization parameter for the connection sparsity control between pixels and anchors and avoiding the trivial solution. Besides, $\mathbf{z}_i^T\mathbf{1} = 1$ and $z_{ij} \geq 0$ are the normalization and non-negative constraints on $\mathbf{Z}$ for sparsity control.

Compared to the images with serveral channels, HSIs with hundreds of spectral channels may suffer from noise interference and the curse of dimensionality. To alleviate these problems, it is reasonable to introduce projection learning into the clustering process. By doing so, the data can be projected into a low-dimensional feature space to reduce noise interference and enhance feature representation, leading to better separability during clustering. Thus, we introduce a projection matrix $\mathbf{W} \in \mathbb{R}^{B\times R}$ into bipartite graph learning of (5) to reduce the spectral dimensionality from B to R in a new subspace as follows:

$$\begin{aligned} &\min_{\mathbf{W},\mathbf{A},\mathbf{Z}} \sum_{i=1}^{N}\sum_{j=1}^{M} \left\|\mathbf{W}^T\mathbf{x}_i - \mathbf{W}^T\mathbf{a}_j\right\|_2^2 z_{ij} + \gamma\|\mathbf{Z}\|_F^2 \\ &\text{s.t.} \quad \mathbf{z}_i^T\mathbf{1} = 1, z_{ij} \geq 0, \mathbf{W}^T\mathbf{S}_t\mathbf{W} = \mathbf{I}, \mathbf{S}_t = \mathbf{X}\mathbf{X}^T, \end{aligned} \tag{6}$$

where $\mathbf{W}^T\mathbf{S}_t\mathbf{W} = \mathbf{I} \in \mathbb{R}^{R\times R}$ indicates the subspace constraint making that the projected data are statistically uncorrelated and easily separated, and $\mathbf{S}_t = \mathbf{X}\mathbf{X}^T \in \mathbb{R}^{B\times B}$ denotes the total scatter matrix to decrease the space complexity. Equation (6) is formulated to learn bipartite graph in a projected subspace. By introducing projection learning, clustering performance can be further improved and convergence state can be achieved with less learning iterations.

For HSI clustering, we desire to directly partition pixels into C clusters. So far, bipartite graph projected learning of (6) relies on an additional clustering step, so that it cannot directly provide clustering results. Compared to two-step learning, one-step learning is more able to provide satisfactory results in a specific task [48]. To achieve one-step learning, one ideal situation is that the optimal neighboring anchor assignment leads to the C exactly connected components in $\mathbf{Z}$. As stated above, we denote the augmented graph by $\mathbf{P} \in \mathbb{R}^{(N+M)\times(N+M)}$ associated with $\mathbf{Z}$ to show the block structure by

$$\mathbf{P} = \begin{bmatrix} & \mathbf{Z} \\ \mathbf{Z}^T & \end{bmatrix}. \tag{7}$$

Based on $\mathbf{P}$, the degree matrix $\mathbf{D} \in \mathbb{R}^{(N+M)\times(N+M)}$ consists of diagonal elements $d_{ii} = \sum_{j=1}^{N+M} p_{ij}$. Thus, the corresponding normalized Laplacian matrix is denoted by

$$\mathbf{L}_Z = \mathbf{I} - \mathbf{D}^{-\frac{1}{2}}\mathbf{P}\mathbf{D}^{-\frac{1}{2}}. \tag{8}$$

As noted in [49], if $\mathbf{Z}$ is non-negative, $\mathbf{L}_Z \in \mathbb{R}^{(N+M)\times(N+M)}$ exhibits the following important property.

Theorem 1: The multiplicity C of the eigenvalue 0 of the normalized Laplacian matrix $\mathbf{L}_Z$ equals the number of connected components in the bipartite graph associated with $\mathbf{Z}$.

In [49], Theorem 1 has been proved that the bipartite graph associated with $\mathbf{Z}$ has C exactly connected components if $rank(\mathbf{L}_Z) = (N + M) - C$. That is to say, N pixels and M anchors can be directly partitioned into C clusters without an additional clustering step. Inspired by Theorem 1, we introduce the rank constraint $rank(\mathbf{L}_Z) = (N + M) - C$ into (6) to learn a structured bipartite graph. Thus, structured bipartite graph learning can be formulated by

$$\begin{aligned} &\min_{\mathbf{W},\mathbf{A},\mathbf{Z}} \sum_{i=1}^{N}\sum_{j=1}^{M} \left\|\mathbf{W}^T\mathbf{x}_i - \mathbf{W}^T\mathbf{a}_j\right\|_2^2 z_{ij} + \gamma\|\mathbf{Z}\|_F^2 \\ &\text{s.t.} \quad \mathbf{z}_i^T\mathbf{1} = 1, z_{ij} \geq 0, \mathbf{W}^T\mathbf{S}_t\mathbf{W} = \mathbf{I}, \mathbf{S}_t = \mathbf{X}\mathbf{X}^T, \\ &\qquad rank(\mathbf{L}_Z) = (N + M) - C. \end{aligned} \tag{9}$$

It is noteworthy that the rank constraint on $\mathbf{L}_Z$ is still very difficult to solve. To relax the rank constraint, we denote $\sigma_i(\mathbf{L}_Z)$ as the i-th smallest eigenvalue of $\mathbf{L}_Z$. Since $\mathbf{L}_Z$ is positive semi-definite, we have $\sigma_i(\mathbf{L}_Z) \geq 0$. The rank constraint can be satisfied if $\sum_{i=1}^{C}\sigma_i(\mathbf{L}_Z)=0$. Based on Ky Fan's theorem [50], we have

$$\sum_{i=1}^{C}\sigma_i(\mathbf{L}_Z)=\min_{\mathbf{F}^T\mathbf{F}=\mathbf{I}}\operatorname{Tr}(\mathbf{F}^T\mathbf{L}_Z\mathbf{F}),\tag{10}$$

where $\mathbf{F}\in\mathbb{R}^{(N+M)\times C}$ is the embedding matrix with orthogonal constraint $\mathbf{F}^T\mathbf{F}=\mathbf{I}$. Combining (10) into (9), the formulation of the proposed BGPC is finally represented as follows:

$$\begin{aligned}\min_{\mathbf{W},\mathbf{A},\mathbf{Z},\mathbf{F}}&\sum_{i=1}^{N}\sum_{j=1}^{M}\left\|\mathbf{W}^T\mathbf{x}_i-\mathbf{W}^T\mathbf{a}_j\right\|_2^2 z_{ij}+\gamma\|\mathbf{Z}\|_F^2\\&+\lambda\operatorname{Tr}(\mathbf{F}^T\mathbf{L}_Z\mathbf{F})\\ \text{s.t.}\quad&\mathbf{z}_i^T\mathbf{1}=1,z_{ij}\geq0,\mathbf{W}^T\mathbf{S}_t\mathbf{W}=\mathbf{I},\mathbf{S}_t=\mathbf{X}\mathbf{X}^T,\mathbf{F}^T\mathbf{F}=\mathbf{I},\end{aligned}\tag{11}$$

where λ is an adaptive regularization parameter to control the number of connected components in $\mathbf{Z}$. When solving problem (11) with a large enough λ, the optimal $\mathbf{Z}$ can be achieved with $\sum_{i=1}^{C}\sigma_i(\mathbf{L}_Z)=0$. In practice, if the number of connected components is not equal to C, the value of λ will be adaptively adjusted. Thus, the optimal $\mathbf{Z}$ exactly contains C connected components. Based on this, all pixels and anchors can be divided into C clusters. The pixels within the same connected component are assigned to the same clustering label. Thus, we can obtain the clustering label for each pixel.

Algorithm 1: Optimization Strategy for BGPC.

```
Input: HSI data $\boldsymbol{\mathcal{X}}\in\mathbb{R}^{W\times H\times B}$, denoising neighbor
       number $K$, projection dimension $R$, cluster
       number $C$
Output: Clustering labels $\mathbf{Y}$
Calculate the number of superpixel by Eq. (26);
Perform PCA and ERS to generate HSI superpixels;
Perform HSI denoising by Eq. (3);
Initialize $\mathbf{A}$ by Eq. (4);
repeat
    Construct $\mathbf{F}$ by solving Eq. (19);
    repeat
        Update $\mathbf{W}$ by solving Eq. (22);
        Update $\mathbf{Z}$ by solving Eq. (15);
        Update $\mathbf{F}$ by solving Eq. (19);
        Adjust $\lambda$ according to the number of connected
          components of $\mathbf{Z}$;
    until Theorem 1 or maximum iteration;
    Update $\mathbf{A}$ by Eq. (25);
until maximum iteration;
Determine $\mathbf{Y}$ from the $C$ connected components in $\mathbf{Z}$.
```

IV. Algorithm

A. Optimization Strategy

It is difficult to jointly update all variables in problem (11) because they are coupled together. Besides, the imposed contraints are nonlinear and unsmooth. To achieve the optimal solution, an optimization strategy is designed to alternately update each variable while fixing others in the sub-problem. The update rules are shown in the following descriptions.

Update $\mathbf{Z}$*:* When $\mathbf{F}$, $\mathbf{W}$ and $\mathbf{A}$ are fixed, the optimization problem of $\mathbf{Z}$ becomes

$$\begin{aligned}\min_{\mathbf{Z}}&\sum_{i=1}^{N}\sum_{j=1}^{M}\left\|\mathbf{W}^T\mathbf{x}_i-\mathbf{W}^T\mathbf{a}_j\right\|_2^2 z_{ij}+\gamma\|\mathbf{Z}\|_F^2\\&+\lambda\operatorname{Tr}(\mathbf{F}^T\mathbf{L}_Z\mathbf{F})\\ \text{s.t.}\quad&\mathbf{z}_i^T\mathbf{1}=1,z_{ij}\geq0.\end{aligned}\tag{12}$$

Note that $\mathbf{L}_Z$, $\mathbf{D}$ and $\mathbf{P}$ rely on $\mathbf{Z}$ making problem (12) difficult to solve. Fortunately, according to the special structure of $\mathbf{P}$, we have the following relationship

$$\begin{aligned}\operatorname{Tr}\left(\mathbf{F}^T\mathbf{L}_Z\mathbf{F}\right)&=\frac{1}{2}\sum_{i=1}^{N+M}\sum_{j=1}^{N+M}p_{ij}\left\|\frac{\mathbf{f}_i}{\sqrt{\mathbf{d}_i}}-\frac{\mathbf{f}_j}{\sqrt{\mathbf{d}_j}}\right\|_2^2\\&=\sum_{i=1}^{N}\sum_{j=1}^{M}z_{ij}h_{ij},\end{aligned}\tag{13}$$

where $h_{ij}=\|\frac{\mathbf{f}_i}{\sqrt{\mathbf{d}_i}}-\frac{\mathbf{f}_{N+j}}{\sqrt{\mathbf{d}_{N+j}}}\|_2^2$, $\mathbf{f}_i$ is the i-th row vector of $\mathbf{F}$ and $\mathbf{d}_j$ is the j-th row vector of $\mathbf{D}$. With (13), problem (12) can be solved by independent updating $\mathbf{z}_i$, *i.e.*, the i-th row vector in $\mathbf{Z}$. Thus, the problem of $\mathbf{z}_i$ can be transformed into

$$\begin{aligned}\min_{\mathbf{z}_i}\sum_{j=1}^{M}&\left(\left\|\mathbf{W}^T\mathbf{x}_i-\mathbf{W}^T\mathbf{a}_j\right\|_2^2 z_{ij}+\gamma z_{ij}^2+\lambda z_{ij}h_{ij}\right)\\ \text{s.t.}\quad&\mathbf{z}_i^T\mathbf{1}=1,z_{ij}\geq0.\end{aligned}\tag{14}$$

To solve problem (14), we denote a vector $\mathbf{e}_i$ with the j-th element $e_{ij}=d_{ij}^x+\lambda h_{ij}$, where $d_{ij}^x=\|\mathbf{W}^T\mathbf{x}_i-\mathbf{W}^T\mathbf{a}_j\|_2^2$. Then, problem (14) can be reformulated by

$$\min_{\mathbf{z}_i^T\mathbf{1}=1,z_{ij}\geq0}\left\|\mathbf{z}_i+\frac{1}{2\gamma}\mathbf{e}_i\right\|_2^2.\tag{15}$$

Following [51], problem (15) can be solved by a closed-form solution. Similar to [51], γ can be adaptively determined by

$$\gamma=\frac{1}{N}\sum_{i=1}^{N}\left(\frac{P}{2}e_{i,P+1}-\frac{1}{2}\sum_{j=1}^{P}e_{ij}\right).\tag{16}$$

It is easy to allocate the size of nearest neighbouring anchors P for the optimal $\mathbf{Z}$ when the rank constraint on $\mathbf{L}_Z$ is satisfied, *i.e.*, $\sum_{i=1}^{C}\sigma_i(\mathbf{L}_Z)=0$.

Update F*:* When $\mathbf{W}$, $\mathbf{A}$ and $\mathbf{Z}$ are fixed, the optimization problem of $\mathbf{F}$ becomes

$$\min_{\mathbf{F}^T\mathbf{F}=\mathbf{I}}\operatorname{Tr}(\mathbf{F}^T\mathbf{L}_Z\mathbf{F}).\tag{17}$$

With (7) and (8), problem (17) can be transformed into

$$\max_{\mathbf{F}^T\mathbf{F}=\mathbf{I}}\operatorname{Tr}\left(\mathbf{F}^T\mathbf{D}^{-\frac{1}{2}}\begin{bmatrix}&\mathbf{Z}\\\mathbf{Z}^T&\end{bmatrix}\mathbf{D}^{-\frac{1}{2}}\mathbf{F}\right).\tag{18}$$

We can rewrite $\mathbf{F}$ and $\mathbf{D}$ with the block matrices

$$\mathbf{F} = \begin{bmatrix} \mathbf{U} \\ \mathbf{V} \end{bmatrix}, \qquad \mathbf{D} = \begin{bmatrix} \mathbf{D}_U & \\ & \mathbf{D}_V \end{bmatrix},$$

where $\mathbf{U} \in \mathbb{R}^{N \times C}$, $\mathbf{V} \in \mathbb{R}^{M \times C}$, $\mathbf{D}_U \in \mathbb{R}^{N \times N}$ and $\mathbf{D}_V \in \mathbb{R}^{M \times M}$. Based on the block structures of $\mathbf{F}$ and $\mathbf{D}$, (18) is equivalent to

$$\max_{\mathbf{U}^T\mathbf{U}+\mathbf{V}^T\mathbf{V}=\mathbf{I}} \operatorname{Tr}\left(\mathbf{U}^T\mathbf{D}_U^{-\frac{1}{2}}\mathbf{Z}\mathbf{D}_V^{-\frac{1}{2}}\mathbf{V}\right), \tag{19}$$

where the constraint $\mathbf{U}^T\mathbf{U} + \mathbf{V}^T\mathbf{V} = \mathbf{I}$ is derived from $\mathbf{F}^T\mathbf{F} = \mathbf{I}$. The above problem can be solved with a closed-from solution based on Lemma 1. According to Lemma 1, $\mathbf{B} = \mathbf{D}_U^{-\frac{1}{2}}\mathbf{Z}\mathbf{D}_V^{-\frac{1}{2}}$ is formulated for simplified calculation.

Lemma 1: Given $\mathbf{B} \in \mathbb{R}^{N \times M}$, $\mathbf{U} \in \mathbb{R}^{N \times C}$, $\mathbf{V} \in \mathbb{R}^{M \times C}$, the optimal solution to the problem

$$\max_{\mathbf{U}^T\mathbf{U}+\mathbf{V}^T\mathbf{V}=\mathbf{I}} \operatorname{Tr}\left(\mathbf{U}^T\mathbf{B}\mathbf{V}\right)$$

are $\mathbf{U} = \frac{\sqrt{2}}{2}\mathbf{U}_1$, $\mathbf{V} = \frac{\sqrt{2}}{2}\mathbf{V}_1$, where $\mathbf{U}_1, \mathbf{V}_1$ are the corresponding C left and right singular vectors of $\mathbf{B}$, respectively.

Update $\mathbf{W}$*:* When $\mathbf{A}$, $\mathbf{Z}$ and $\mathbf{F}$ are fixed, the optimization problem of $\mathbf{W}$ becomes

$$\min_{\mathbf{W}^T\mathbf{S}_t\mathbf{W}=\mathbf{I}} \sum_{i=1}^{N}\sum_{j=1}^{M} \left\|\mathbf{W}^T\mathbf{x}_i - \mathbf{W}^T\mathbf{a}_j\right\|_2^2 z_{ij}. \tag{20}$$

Since $\|\mathbf{A}\|_2^2 = \operatorname{Tr}(\mathbf{A}^T\mathbf{A})$, we have

$$\begin{aligned}
&\sum_{i=1}^{n}\sum_{j=1}^{m} \left\|\mathbf{W}^T\mathbf{x}_i - \mathbf{W}^T\mathbf{a}_j\right\|_2^2 z_{ij} \\
&= \operatorname{Tr}\left(\mathbf{W}^T \sum_{i=1}^{n}\sum_{j=1}^{m} (\mathbf{x}_i - \mathbf{a}_j)(\mathbf{x}_i - \mathbf{a}_j)^T z_{ij}\mathbf{W}\right) \\
&= \operatorname{Tr}\left(\mathbf{W}^T\left(\mathbf{X}\mathbf{D}_U\mathbf{X}^T - 2\mathbf{X}\mathbf{Z}\mathbf{A}^T + \mathbf{A}\mathbf{D}_V\mathbf{A}^T\right)\mathbf{W}\right),
\end{aligned} \tag{21}$$

where the diagonal matrices $\mathbf{D}_U$ and $\mathbf{D}_V$ are with their diagonal elements $d_{ii}^U = \sum_{j=1}^{M} z_{ij}$ and $d_{jj}^V = \sum_{i=1}^{N} z_{ij}$, respectively. Letting $\mathbf{G} = (\mathbf{X}\mathbf{D}_U\mathbf{X}^T - 2\mathbf{X}\mathbf{Z}\mathbf{A}^T + \mathbf{A}\mathbf{D}_V\mathbf{A}^T)$, problem (20) can be simplified by

$$\min_{\mathbf{W}^T\mathbf{S}_t\mathbf{W}=\mathbf{I}} \operatorname{Tr}\left(\mathbf{W}^T\mathbf{G}\mathbf{W}\right). \tag{22}$$

To solve problem (22), we decompose the eigenvalue of $\mathbf{S}_t^{-1}\mathbf{G}$. The optimal $\mathbf{W}$ is composed of the eigenvectors corresponding to the R smallest eigenvalues.

Update $\mathbf{A}$*:* When $\mathbf{Z}$, $\mathbf{F}$ and $\mathbf{W}$ are fixed, the optimization problem of $\mathbf{A}$ can be solved by M decomposed problems

$$\min_{\mathbf{a}_j} \sum_{i=1}^{N} \left\|\mathbf{W}^T\mathbf{x}_i - \mathbf{W}^T\mathbf{a}_j\right\|_2^2 z_{ij}. \tag{23}$$

To solve problem (23), we first let $\mathcal{J} = \sum_{i=1}^{N} \|\mathbf{W}^T\mathbf{x}_i - \mathbf{W}^T\mathbf{a}_j\|_2^2 z_{ij}$, and then calculate the derivative w.r.t $\mathbf{a}_i$ and set it to zero as follows

$$\frac{\partial \mathcal{J}}{\partial \mathbf{a}_j} = \sum_{i=1}^{N} 2\mathbf{W}^T(\mathbf{x}_i - \mathbf{a}_j)z_{ij} = 0. \tag{24}$$

TABLE I
INFORMATION OF IMPORTANT PARAMETERS FOR OUR BGPC

Paramters	Notations	Settings
Regularization Parameter	γ	Eq. (16)
Regularization Parameter	λ	$\lambda = \lambda * 2$ or $\lambda = \lambda/2$
Projected Dimensionality	R	$[0.25B,\ 0.5B]$
Neighbour size for denoising	K	13
Neighbour size for update $\mathbf{Z}$	P	5
Parameter for superpixel	S	2000

According to (24), the update rule of $\mathbf{a}_i$ can be given by

$$\mathbf{a}_j = \frac{\sum_{i=1}^{N} z_{ij}\mathbf{x}_i}{\sum_{i=1}^{N} z_{ij}}. \tag{25}$$

In (11), $\mathbf{L}_Z$ with the larger size of $(N + M) \times (N + M)$ is related to the update rules of $\mathbf{Z}$ and $\mathbf{F}$. Based on the special block structure of $\mathbf{L}_Z$, the optimization of $\mathbf{Z}$ and $\mathbf{F}$ can be transformed into the problems with smaller size as (15) and (19). Thus, the proposed model still shows efficient computation to handle large-scale HSI data. This can be also observed from the near-linear time complexity of the update rules for $\mathbf{Z}$ and $\mathbf{F}$ shown as in Section IV-B. To better illustrate the optimization strategy, we summarize the update rules for solving problem (11) in Algorithm 1.

B. Computational Complexity Analysis

The computational complexity of solving problem (11) with Algorithm 1 is mainly attributed to the iterative learning process to alternatively update each variable. The construction or updating for $\mathbf{F}$ costs $\mathcal{O}(M^3 + M^2N)$ by solving (19). When $\mathbf{W}$ is updated by solving (22), the time complexity is $\mathcal{O}(B^3)$ because of matrix inversion and eigenvalue decomposition. The time complexity of solving $\mathbf{Z}$ through a closed-form solution to (15) is $\mathcal{O}(NMC + NMlog(M))$. To update $\mathbf{A}$ with (25), it needs the computational complexity of $\mathcal{O}(NMB)$. Since there is a sub-alternating system in Algorithm 1, we denote the number of iterations for the inner loop by T_1 and the outer loop by T_2. In summary, the computational complexity of BGPC is $\mathcal{O}(((B^3 + NMB + M^2N)T_1 + NMB + M^2N)T_2)$ because $log(M)$ is small and $M^3 \ll M^2N$. Denoting $T = T_1T_2$, it can be further simplified as $\mathcal{O}(NMBT)$ because M is approximate to B and $B \ll N$. It is worth noting that the computational complexity of BGPC is linear to N.

V. EXPERIMENTS

A. Datasets

The experiments are conducted on three public HSI datasets, including Salinas, Pavia University and Pavia Center. Their false color images and superpixel segmentation images are shown in Fig. 3.

1) Salinas: This dataset was acquired by the Airborne Visible/Infrared Imaging Spectrometer (AVIRIS) sensor to record the scene over the Salinas valley, Southern California, in 1998. There are 512 × 217 pixels to archive the distribution of 16 land-cover categories with a spatial resolution of 3.7 m. After

TABLE II
CLUSTERING PERFORMANCE COMPARISON OF DIFFERENT METHODS ON HSI DATASETS

Datasets	Metrics	K-means	FCM	FSCAG	SGCNR	HESSC	NCSC	SGLSC	S^3AGC	DSCRLE	BGPC
Salinas	ACC	0.6456	0.5717	0.6590	0.6224	0.5293	0.7497	0.7664	0.7465	0.7575	**0.7880**
	Kappa	0.6044	0.5333	0.6212	0.5790	0.4787	0.7206	0.7407	0.7181	0.7314	**0.7632**
	NMI	0.7275	0.6920	0.7412	0.7124	0.6238	0.8176	**0.8388**	0.7836	0.8059	0.8134
	Purity	0.6640	0.6697	0.7070	0.6688	0.5881	0.7638	0.7679	0.7521	0.7916	**0.7931**
	ARI	0.5392	0.4480	0.5444	0.4978	0.4150	0.6650	0.6553	0.6095	0.5853	**0.6887**
	F-score	0.5549	0.5794	0.5818	0.5462	0.4375	0.5821	0.6072	0.6668	**0.7986**	0.6974
Pavia University	ACC	0.5370	0.5190	0.5288	0.5019	0.4790	0.4392	0.6245	0.5375	0.6790	**0.7405**
	Kappa	0.4357	0.4241	0.4291	0.4039	0.3663	0.2952	0.4885	0.4245	0.6038	**0.6405**
	NMI	0.5321	0.5273	0.5588	0.5068	0.4982	0.4355	0.5410	0.4902	0.6256	**0.6900**
	Purity	0.6957	0.6940	0.7055	0.6628	0.6397	0.6037	0.6769	0.6395	0.7244	**0.7406**
	ARI	0.3255	0.3065	0.3306	0.2982	0.3020	0.2905	0.5063	0.3384	0.4383	**0.6154**
	F-score	0.5079	0.5293	0.5309	0.4767	0.4130	0.3656	0.4201	0.4957	**0.7077**	0.5646
Pavia Center	ACC	0.7236	0.6630	0.7275	0.7453	0.7848	0.6584	0.8204	0.8548	0.8913	**0.9470**
	Kappa	0.6199	0.5419	0.6337	0.6519	0.6971	0.5320	0.7498	0.7905	0.8507	**0.9249**
	NMI	0.6995	0.6689	0.7339	0.6902	0.6648	0.5959	0.7651	0.7937	0.8707	**0.9263**
	Purity	0.8378	0.8313	0.8481	0.8244	0.8250	0.7829	0.8597	0.8718	0.9529	**0.9538**
	ARI	0.7669	0.7415	0.8057	0.7594	0.7381	0.7255	0.8707	0.8804	0.8723	**0.9837**
	F-score	0.4383	0.3828	0.4968	0.5475	0.4762	0.3273	0.5094	0.5605	0.7833	**0.8022**

The best results are shown in bold.

TABLE III
p-VALUES OF SIGNIFICANT DIFFERENCE BETWEEN BGPC AND OTHER METHODS ON HSI DATASETS

Comparisons	Salinas	Pavia University	Pavia Center
Kmeans-BGPC	1.93E-14	1.50E-18	4.00E-17
FCM-BGPC	7.88E-15	2.06E-23	4.09E-36
FSCAG-BGPC	4.91E-11	4.61E-18	3.19E-20
SGCNR-BGPC	2.67E-09	8.14E-16	1.23E-08
HESSC-BGPC	2.31E-275	7.29E-281	5.31E-269
NCSC-BGPC	9.93E-258	5.68E-282	1.66E-273
SGLSC-BGPC	2.98E-253	4.32E-269	1.76E-263
S^3AGC-BGPC	4.75E-261	1.13E-280	2.71E-267
DSCRLE-BGPC	9.23E-11	1.02E-10	1.16E-14

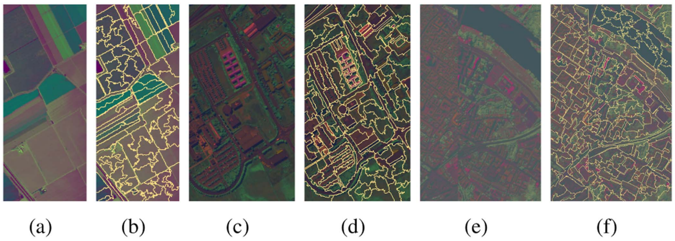


Fig. 3. False color images and their homogenous regions from superpixel segmentation of HSI datasets. (a) and (b) Salinas. (c) and (d) Pavia University. (e) and (f) Pavia Center.

discarding 20 water absorption bands, 204 spectral bands between 0.4 and 2.5 μm are left to describe this scene.

2) Pavia University: This dataset was gathered by the reflective optics system imaging spectrometer (ROSIS) sensor over the University of Pavia, Italy, in 2002. After abandoning 12 noisy and water absorption bands, there are 103 spectral bands left from 0.43 to 0.86 μm to describe the scene with 610 × 340 pixels and 9 categories of land covers at a spatial resolution of 1.3 m.

3) Pavia Center: This dataset was also collected by the ROSIS sensor over the center of Pavia, Italy, in 2003. The original image consists of 1096 × 1096 pixels at a spatial resolution of 1.3 m and 115 spectral bands between 0.43 and 0.86 μm. Since there are a part of invalid pixels and low signal-to-noise ratio (SNR) bands, we retain 1096 × 715 pixels with 9 land-cover categories to be described by 102 spectral bands for experiment. This is a large-scale HSI dataset with a total of 783640 pixels.

B. Compared Methods

To validate the effectiveness of the proposed BGPC, several clustering methods are adopted for comparison.

1) *K-means* is a hard clustering method to pursue the minimum distance between pixels and cluster centers.

2) *FCM* is a soft clustering method developed on K-means to identify the membership between pixels and cluster centers.

3) *FSCAG* [29] is a fast spectral clustering method with anchor graph by means of spatial neighbors.

4) *SGCNR* [37] is a scalable graph-based clustering method with nonnegative relaxation to obtain clustering results.

5) *HESSC* [38] is a hierarchical sparse subspace clustering method to adaptively determine the number of clusters.
6) *NCSC* [28] is a neighborhood contrastive subspace clustering network to rescale clustering at superpixel level.

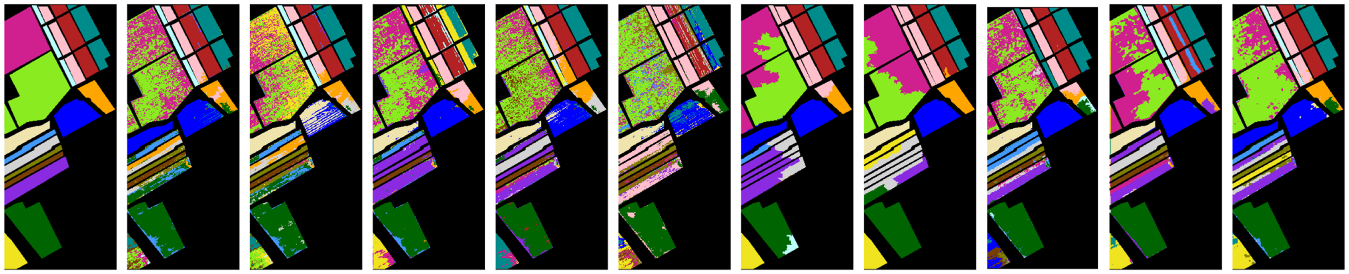
7) *SGLSC* [31] is a superpixel-level global and local similarity graph-based clustering to alleviate the computational burden for large HSI data.

8) *S^3AGC* [52] is an anchor graph-based clustering method with spatial and structural information.

9) *DSCRLE* [53] is a deep spectral clustering method by combining spectral graph theory and deep learning for HSI clustering.

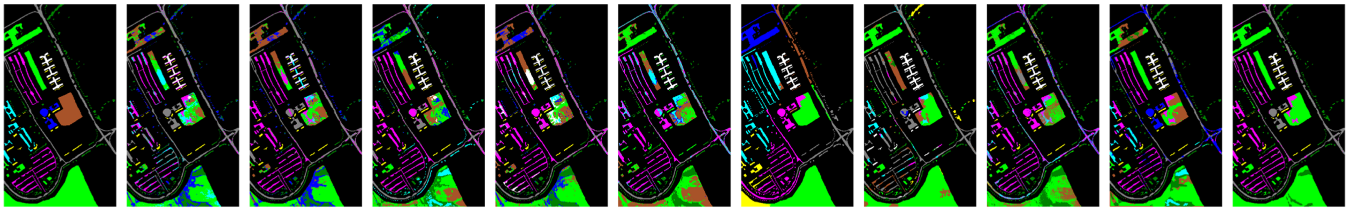
C. Experimental Setup

There are several parameters in BGPC needed to determine in advance. For the regularization parameters, γ is determined by (16), while λ is initialized as γ and adaptively updated in a heuristic way. In each iteration, the variation of λ is $\lambda = \lambda * 2$ or



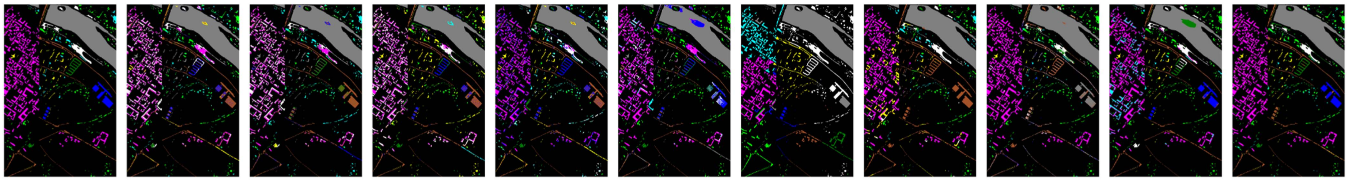
(a) GT (b) K-means (c) FCM (d) FSCAG (e) SGCNR (f) HESSC (g) NCSC (h) SGLSC (i) S^3AGC (j) DSCRLE (k) BGPC

Fig. 4. Clustering maps and ground truth (GT) of the Salinas dataset.



(a) GT (b) K-means (c) FCM (d) FSCAG (e) SGCNR (f) HESSC (g) NCSC (h) SGLSC (i) S^3AGC (j) DSCRLE (k) BGPC

Fig. 5. Clustering maps and ground truth (GT) of the Pavia University dataset.



(a) GT (b) K-means (c) FCM (d) FSCAG (e) SGCNR (f) HESSC (g) NCSC (h) SGLSC (i) S^3AGC (j) DSCRLE (k) BGPC

Fig. 6. Clustering maps and ground truth (GT) of the Pavia Center dataset.

$\lambda = \lambda/2$ if the number of the connected components in $\mathbf{Z}$ is less or more than C. To find out a superior projection dimension, R is tuned from $0.25B$ to $0.5B$. For simplicity, K is set as 13 to control the number of neighboring pixels during denoising, and P is assigned as 5 to confirm the number of neighboring anchors for (16). To exploit spatial information within HSIs, it is vital to determine an appropriate scale for segmentation. Following [54], the number of superpixels is adaptively determined based on the texture of the HSI by

$$M = S\frac{N_z}{N}, \tag{26}$$

where N_z denotes the number of non-zero elements in the detected edge image, N represents the number of pixels, and S is an adjustable parameter and default as 2000. The parameter setting for our BGPC is shown in Table I. The hyper-parameters related to other compared methods are set based on the suggestions in original papers to show their best performance.

To quantify the clustering performance, we adopt six commonly used metrics for evaluation. They are overall clustering accuracy (ACC), Kappa coefficient, normalized mutual information (NMI), purity, adjusted rand index (ARI) and F-score. For all these metrics, higher values signify better clustering performance. Except for NCSC and DSCRLE implemented in Python 3.7.13, all methods are implemented in MATLAB R2020a. For the implementation of our BGPC, the edge detection for HSI segmentation is based on the Image Processing Toolbox in MATLAB, while the connected component discovery for bipartite graph is based on the Bioinformatics Toolbox in MATLAB. All experiments are run on a machine with 3.00 GHz CPU and 64 G RAM. To alleviate randomness, the experiments for each compared method are repeated 10 times and their average results are reported.

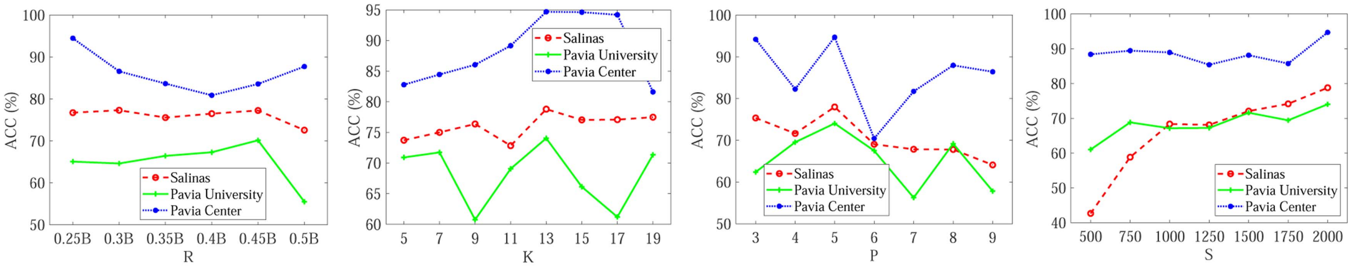
D. Experimental Results

Table II displays the clustering results of different methods in terms of six evaluation metrics. The optimal results are in bold font for emphasis. Based on Table II, the following observations can be given. First, K-means is superior to FCM with 5.72% overall improvement for traditional clustering, while SGLSC is better than FSCAG, SGCNR, HESSC and NCSC with 10.35%, 15.92%, 25.84% and 23.53% overall enhancement for advanced clustering. Besides, traditional clustering methods are inferior to advanced clustering methods in most cases. This is because advanced clustering methods simultaneously consider spectral and spatial information during learning while traditional clustering methods only utilize spectral information. Second, for pixel-level methods, FSCAG and SGCNR are superior to HESSC on the Salinas dataset and inferior to HESSC on the Pavia University and Pavia Center datasets. For superpixel-level

TABLE IV
COMPUTATIONAL EFFICIENCY COMPARISON OF DIFFERENT METHODS ON HSI DATASETS (MEASURED BY SECOND)

Computational Consumption		K-means	FCM	FSCAG	SGCNR	HESSC	NCSC	SGLSC	S^{3}AGC	DSCRLE	BGPC
Time Complexity		$\mathcal{O}(NBCT)$	$\mathcal{O}(NBCT)$	$\mathcal{O}(NMB)$	$\mathcal{O}(NMB)$	$\mathcal{O}(NB\log N)$	$\mathcal{O}(M^2BT)$	$\mathcal{O}(M^2BT)$	$\mathcal{O}(NMBC)$	$\mathcal{O}(NWT)$	$\mathcal{O}(NMBT)$
Runtime	Salinas	4.07	17.18	6.95	28.67	633.42	4204.52	29.90	13.80	11243.33	81.02
	Pavia University	3.71	13.09	9.96	26.61	946.31	8003.41	356.05	25.33	4879.72	198.63
	Pavia Center	26.56	49.15	52.55	127.48	4637.90	52612.00	1500.00	149.02	16970.24	981.89

N: pixel size. B: spectral dimension. M: anchor/superpixel size. C: cluster size. T: iteration size. W: network parameter size.



(a) Effect of R (b) Effect of K (c) Effect of P (d) Effect of S

Fig. 7. Performance variance of BGPC with respect to different settings for the parameter K, S, R and P on HSI datasets.

methods, SGLSC always provides better results compared to NCSC. To obtain better performance for NCSC and SGLSC, it is very important to generate the accurate superpixels. Third, it is clear that our BGPC significantly outperforms other competitors except for SGLSC in terms of NMI and DSCRLE in terms of F-score. Compared to the second best method DSCRLE, BGPC gains at least 1.64%, 5.64% and 6.07% improvement by averaging six metrics on the three datasets, respectively. In comparison with other superior methods, *e.g,* SGLSC and S ^{3} AGC, the enhancement of BGPC is even more evident. The superiority of BGPC is attributed to the exploitation of spatial information in denoising and anchor initialization for the joint learning of projection and optimal bipartite graph to directly obtain the clustering outcome after one-pass learning.

To demonstrate the significant difference between BGPC and other methods from a statistical perspective, we utilize a two-tailed t-test with 95% confidence for evaluation. Table III presents the statistical results with respect to ACC. There is a significant difference between two compared methods if their p-value is smaller than 0.05. Based on Table III, it is obvious that all p-values of comparisons between BGPC and other methods are much smaller than 0.05 on the three HSI datasets. Recalling Table II, we can find that BGPC always exhibits superior clustering performance with significant difference than other methods. In addition, to visualize the clustering performance, we display the clustering maps of the three HSI datasets obtained by different methods in Figs. 4–6. As can be observed, the superiority of BGPC is aligned with the aforementioned observations.

E. Computational Efficiency Analysis

To analyze the computational efficiency of all compared methods, we display their time complexities and corresponding running time in Table IV. About the notations of time complexity, N, B, M, C and T denotes the number of pixels, bands, anchors/superpixels, clusters and learning iterations, respectively.

TABLE V
CLUSTERING PERFORMANCE OF BGPC WITH DIFFERENT SEGMENTATION METHODS AND THEIR RUNNING TIME FOR SEGMENTATION ON HSI DATASETS (MEASURED BY SECOND)

Datsets	Segmentation Methods	Metrics			
		ACC	NMI	Purity	Runtime
Salinas (512×217)	SLIC	0.7451	0.8036	0.7754	0.28
	LSC	0.7365	0.8117	0.7506	0.33
	ERS	**0.7880**	**0.8134**	**0.7931**	0.24
Pavia University (610×340)	SLIC	0.6520	0.6389	0.7091	0.43
	LSC	0.6610	0.6052	0.7130	0.54
	ERS	**0.7405**	**0.6900**	**0.7406**	0.38
Pavia Center (1096×715)	SLIC	0.8244	0.7751	0.8311	1.43
	LSC	0.9023	0.9189	0.9078	1.91
	ERS	**0.9470**	**0.9263**	**0.9538**	2.34

The best results are shown in bold.

As can be observed, K-means, FCM, FSCAG, SGCNR, S^{3}AGC, DSCRLE and our BGPC show the linear time complexity for the number of pixels, while NCSC and SGLSC represent the quadratic time complexity for the number of superpixels. Besides, HESSC demonstrates the efficiency between the linear and quadratic time complexity for the number of pixels. It is clear that our BGPC demonstrates the superiority in time complexity.

As for running time, K-means, FCM, FSCAG, SGCNR and S^{3}AGC are more efficient among all compared methods due to the linear time complexity. Compared to these five methods, our BGPC requires more runing time for pursuing exactly connected components. It is obvious that HESSC, NCSC and DSCRLE are the most time-consuming methods because of the repeated tree node partition and complex network training. For SGLSC, it needs less running time on the Salinas dataset, but takes much more running time on the Pavia University and Pavia Center datasets. Since these datasets are with large sizes, the running time of BGPC is reasonable in practice. In summary, our BGPC

TABLE VI
ABLATION STUDIES OF OUR BGPC ON HSI DATASETS

Setting	Components				Salinas			Pavia University			Pavia Center		
	SG	SD	SAS	PL	ACC	NMI	Purity	ACC	NMI	Purity	ACC	NMI	Purity
S1	✗	✗	✗	✗	0.5745	0.6953	0.5791	0.4761	0.2369	0.4896	0.7199	0.6851	0.8230
S2	✓	✓	✗	✗	0.6896	0.7890	0.6952	0.5636	0.5736	0.7014	0.7809	0.7271	0.8024
S3	✓	✗	✓	✗	0.6653	0.7700	0.6654	0.6093	0.5479	0.6298	0.7609	0.6702	0.8246
S4	✓	✓	✓	✗	0.7009	0.7572	0.7037	0.6380	0.6298	0.5082	0.8204	0.7910	0.8583
BGPC	✓	✓	✓	✓	**0.7880**	**0.8134**	**0.7931**	**0.7405**	**0.6900**	**0.7406**	**0.9470**	**0.9263**	**0.9538**

SG: Superpixel generation. SD: Spatial denoising. SAS: Spatial anchor selection. PL: Projection learning.
The best results are shown in bold.

TABLE VII
CLUSTERING PERFORMANCE OF BGPC WITH DIFFERENT DENOISING STRATEGIES ON HSI DATASETS

Datsets	Denoising Strategies	Metrics		
		ACC	NMI	Purity
Salinas (512×217)	Entire image	0.7431	0.7136	0.7998
	Window	0.7385	0.7100	0.7759
	Superpixel	**0.7880**	**0.8134**	**0.7931**
Pavia University (610×340)	Entire image	0.6605	0.5504	0.5788
	Window	0.6974	0.5929	0.6637
	Superpixel	**0.7405**	**0.6900**	**0.7406**
Pavia Center (1096×715)	Entire image	0.8682	0.8076	0.8161
	Window	0.8260	0.7602	0.8642
	Superpixel	**0.9470**	**0.9263**	**0.9538**

The best results are shown in bold.

not only shows superior clustering performance but also represents acceptable computational efficiency.

F. Parameter Sensitivity Study

For BGPC, the dimension R is crucial for projection learning, the parameter K is related to superpixel denoising, the parameter P is relevant to update $\mathbf{Z}$, and the parameter S is vital for superpixel segmentation. To study their parameter sensitivities, we report the performance variance by assigning different values for them in Fig. 7. Specifically, R is tuned from $0.25B$ to $0.5B$ with step $0.05B$, K is varying from 5 to 19 with step 2, S is adjusted from 500 to 2000 with step 250, and P is adapted from 3 to 9 with step 1. With different values for them, BGPC shows different clustering results. In detail, BGPC show superior performance even when R is small. The performance variance with different dimensions is attributed to the information discrepancy of different projected representations. The performance of BGPC is superior when K is moderate, and the best results are obtained when $K = 13$. The suitable value for K is conducive to perform effective denoising for robust representation and further learning. Besides, BGPC demonstrates the best performance when $S = 2000$, and displays inferior performance when $S \leq 1750$. The suitable value for S is beneficial to generating superpixels with well-exploited spatial information. The performance of BGPC is fluctuant when setting different values for P, and the optimal results are achieved when $P = 5$. The proper value for P is more able to learn the sparsity and connected components of the bipartite graph. The above observations also validate our experimental setup.

G. Superpixel Segmentation Analysis

In our BGPC, ERS [45] is employed as a tool to localize the homogeneity of HSI data and generate superpixels with spatial information. Apart from ERS, many representative methods are designed for superpixel segmentation, such as simple linear iterative clustering (SLIC) [55] and linear spectral clustering (LSC) [56]. In Table. V, we report the clustering performance of our BGPC with three segmentation methods (SLIC, LSC and ERS) and record the corresponding running time for segmentation. We can find that the three segmentation methods are very efficient. With different segmentation methods, our BGPC represents different clustering results. It is clear that our BGPC with ERS provides much better clustering results than the other two settings. Thus, we utilize ERS in our setting because of its potential in both effectiveness and efficiency.

H. Ablation Study

To investigate the effect of each component, we conduct ablation study to show the degraded variants of BGPC with different settings by removing different components. For the degraded variants, we use the notations “√” and “✗” to indicate their settings containing a specific component or not. As can be observed from Table VI, the variant with setting “S1” shows the worst performance without considering any components. Under local region guidance, the variant with setting “S2” containing spatial denoising is superior to the variant with setting “S3” containing spatial anchor selection in most cases. By introducing both spatial denoising and spatial anchor selection, the variant with setting “S4” significantly outperforms the variants with setting “S2” and setting “S3”. It is clear that, compared to the second best variant with setting “S4”, BGPC shows the best performance with at least 10.76%, 22.24% and 14.48% improvement on the three datasets by simultaneous considering all valuable components for better HSI clustering.

To further validate the effect of our spatial denoising and anchor selection, we also use different denoising strategies [29] and different anchor selection strategies [37] for substitution and comparison. The results are presented in Tables VII–VIII. Compared to denoising across the entire image and denoising with a square window, our spatial denoising strategy with superpixel is able to well utilize spatial information to achieve

TABLE VIII
CLUSTERING PERFORMANCE OF BGPC WITH DIFFERENT ANCHOR INITIALIZATION STRATEGIES ON HSI DATASETS

Datsets	Anchor Strategies	Metrics		
		ACC	NMI	Purity
Salinas (512×217)	Random	0.6324	0.5855	0.7595
	K-means	0.7232	0.6897	0.8101
	Superpixel	**0.7880**	**0.8134**	**0.7931**
Pavia University (610×340)	Random	0.6190	0.5187	0.5687
	K-means	0.6517	0.5140	0.6066
	Superpixel	**0.7405**	**0.6900**	**0.7406**
Pavia Center (1096×715)	Random	0.8440	0.7851	0.8402
	K-means	0.8931	0.8483	0.8808
	Superpixel	**0.9470**	**0.9263**	**0.9538**

The best results are shown in bold.

much higher clustering results. This is because HSI denoising with spatial information is able to eliminate the noises and improve the representation capability. Compared to random strategy and K-means strategy, our spatial anchor selection strategy with superpixel also exhibit better clustering performance with the aid of spatial information. This can be demonstrated that anchor initialization with spatial information is good at selecting high-quality anchors.

VI. CONCLUSION

In this paper, we proposed a bipartite graph-based projected clustering (BGPC) method with local region guidance for HSIs. Based on texture information, the HSI is partitioned into multiple superpixels to reflect the spatial distribution of land covers. For each superpixel, denoising is performed by reconstructing pixels with a weighted summation of nearest neighboring pixels and anchor initialization is determined by averaging all denoised pixels. To facilitate efficient clustering, projection learning and structured bipartite graph learning is jointly formulated into the optimization problem of the ideal and adaptive neighbor assignment. An alternating optimization strategy is designed as solution to the formulated problem. It can directly provide clustering results with linear time complexity. Extensive experiments have demonstrated the remarkable performance of our BGPC by comparing the state-of-the-art HSI clustering methods.

REFERENCES

[1] Y. Zhang, J. Wu, Z. Cai, and S. Y. Philip, “Multi-view multi-label learning with sparse feature selection for image annotation,” *IEEE Trans. Multimedia*, vol. 22, pp. 2844–2857, 2020.

[2] L. Zhang and L. Zhang, “Artificial intelligence for remote sensing data analysis: A review of challenges and opportunities,” *IEEE Geosci. Remote Sens. Mag.*, vol. 10, no. 2, pp. 270–294, Jun. 2022.

[3] B. Du, M. Zhang, L. Zhang, R. Hu, and D. Tao, “PLTD: Patch-based low-rank tensor decomposition for hyperspectral images,” *IEEE Trans. Multimedia*, vol. 19, pp. 67–79, 2017.

[4] H. Zhu, H. Ni, S. Liu, G. Xu, and L. Deng, “TNLRS: Target-aware non-local low-rank modeling with saliency filtering regularization for infrared small target detection,” *IEEE Trans. Image Process.*, vol. 29, pp. 9546–9558, 2020.

[5] C. Shi and C.-M. Pun, “Multiscale superpixel-based hyperspectral image classification using recurrent neural networks with stacked autoencoders,” *IEEE Trans. Multimedia*, vol. 22, pp. 487–501, 2020.

[6] H. Zhu et al., “Bilateral weighted regression ranking model with spatial-temporal correlation filter for visual tracking,” *IEEE Trans. Multimedia*, vol. 24, pp. 2098–2111, 2022.

[7] Y. Xu, B. Du, and L. Zhang, “Self-attention context network: Addressing the threat of adversarial attacks for hyperspectral image classification,” *IEEE Trans. Image Process.*, vol. 30, pp. 8671–8685, 2021.

[8] S. Jia, S. Jiang, S. Zhang, M. Xu, and X. Jia, “Graph-in-graph convolutional network for hyperspectral image classification,” *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 35, no. 1, pp. 1157–1171, Jan. 2024.

[9] Y. Zhou and Y. Wei, “Learning hierarchical spectral–spatial features for hyperspectral image classification,” *IEEE Trans. Cybern.*, vol. 46, no. 7, pp. 1667–1678, Jul. 2016.

[10] L. Zhang, L. Zhang, B. Du, J. You, and D. Tao, “Hyperspectral image unsupervised classification by robust manifold matrix factorization,” *Inf. Sci.*, vol. 485, pp. 154–169, 2019.

[11] Y. Zhang, Y. Wang, X. Chen, X. Jiang, and Y. Zhou, “Spectral–spatial feature extraction with dual graph autoencoder for hyperspectral image clustering,” *IEEE Trans. Circuits Syst. Video Technol.*, vol. 32, no. 12, pp. 8500–8511, Dec. 2022.

[12] P. Ghamisi, A.-R. Ali, M. S. Couceiro, and J. A. Benediktsson, “A novel evolutionary swarm fuzzy clustering approach for hyperspectral imagery,” *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 8, no. 6, pp. 2447–2456, Jun. 2015.

[13] H. Zhai, H. Zhang, P. Li, and L. Zhang, “Hyperspectral image clustering: Current achievements and future lines,” *IEEE Geosci. Remote Sens. Mag.*, vol. 9, no. 4, pp. 35–67, Dec. 2021.

[14] B. Zhang et al., “A neighbourhood-constrained k-means approach to classify very high spatial resolution hyperspectral imagery,” *Remote Sens. Lett.*, vol. 4, no. 2, pp. 161–170, 2013.

[15] S. Ghaffarian and S. Ghaffarian, “Automatic histogram-based fuzzy c-means clustering for remote sensing imagery,” *ISPRS J. Photogramm. Remote Sens.*, vol. 97, pp. 46–57, 2014.

[16] C. Cariou and K. Chehdi, “Unsupervised nearest neighbors clustering with application to hyperspectral images,” *IEEE J. Sel. Topics Signal Process.*, vol. 9, no. 6, pp. 1105–1116, Sep. 2015.

[17] B. Tu, X. Yang, N. Li, C. Zhou, and D. He, “Hyperspectral anomaly detection via density peak clustering,” *Pattern Recognit. Lett.*, vol. 129, pp. 144–149, 2020.

[18] A. Paoli, F. Melgani, and E. Pasolli, “Clustering of hyperspectral images based on multiobjective particle swarm optimization,” *IEEE Trans. Geosci. Remote Sens.*, vol. 47, no. 12, pp. 4175–4188, Dec. 2009.

[19] Y. Zhong, L. Zhang, B. Huang, and P. Li, “An unsupervised artificial immune classifier for multi/hyperspectral remote sensing imagery,” *IEEE Trans. Geosci. Remote Sens.*, vol. 44, no. 2, pp. 420–431, Feb. 2006.

[20] H. Zhang, H. Zhai, L. Zhang, and P. Li, “Spectral–spatial sparse subspace clustering for hyperspectral remote sensing images,” *IEEE Trans. Geosci. Remote Sens.*, vol. 54, no. 6, pp. 3672–3684, Jun. 2016.

[21] L. Tian, Q. Du, I. Kopriva, and N. Younan, “Spatial-spectral based multi-view low-rank sparse sbuspace clustering for hyperspectral imagery,” in *Proc. IEEE Int. Geosci. Remote Sens. Symp.*, 2018, pp. 8488–8491.

[22] H. Zhai, H. Zhang, L. Zhang, P. Li, and A. Plaza, “A new sparse subspace clustering algorithm for hyperspectral remote sensing imagery,” *IEEE Geosci. Remote Sens. Lett.*, vol. 14, no. 1, pp. 43–47, Jan. 2017.

[23] S. Huang, H. Zhang, and A. Pižurica, “Subspace clustering for hyperspectral images via dictionary learning with adaptive regularization,” *IEEE Trans. Geosci. Remote Sens.*, vol. 60, 2022, Art. no. 5524017.

[24] Y. Cai et al., “Graph convolutional subspace clustering: A robust subspace clustering framework for hyperspectral image,” *IEEE Trans. Geosci. Remote Sens.*, vol. 59, no. 5, pp. 4191–4202, May 2021.

[25] X. Hu, T. Li, T. Zhou, and Y. Peng, “Deep spatial-spectral subspace clustering for hyperspectral images based on contrastive learning,” *Remote Sens.*, vol. 13, no. 21, 2021, Art. no. 4418.

[26] T. Han et al., “Deep low-rank graph convolutional subspace clustering for hyperspectral image,” *IEEE Trans. Geosci. Remote Sens.*, vol. 60, 2022, Art. no. 5533513.

[27] J. Lei et al., “Deep spatial-spectral subspace clustering for hyperspectral image,” *IEEE Trans. Circuits Syst. Video Technol.*, vol. 31, no. 7, pp. 2686–2697, Jul. 2021.

[28] Y. Cai et al., “Superpixel contracted neighborhood contrastive subspace clustering network for hyperspectral images,” *IEEE Trans. Geosci. Remote Sens.*, vol. 60, 2022, Art. no. 5530113.

[29] R. Wang, F. Nie, and W. Yu, "Fast spectral clustering with anchor graph for large hyperspectral images," *IEEE Geosci. Remote Sens. Lett.*, vol. 14, no. 11, pp. 2003–2007, Nov. 2017.

[30] X. Yang, Y. Xu, S. Li, Y. Liu, and Y. Liu, "Fuzzy embedded clustering based on bipartite graph for large-scale hyperspectral image," *IEEE Geosci. Remote Sens. Lett.*, vol. 19, 2022, Art. no. 5505605.

[31] H. Zhao, F. Zhou, L. Bruzzone, R. Guan, and C. Yang, "Superpixel-level global and local similarity graph-based clustering for large hyperspectral images," *IEEE Trans. Geosci. Remote Sens.*, vol. 60, 2022, Art. no. 5519316.

[32] Q. Wang, Y. Miao, M. Chen, and Y. Yuan, "Spatial-spectral clustering with anchor graph for hyperspectral image," *IEEE Trans. Geosci. Remote Sens.*, vol. 60, 2022, Art. no. 5542413.

[33] X. Yang, G. Lin, Y. Liu, F. Nie, and L. Lin, "Fast spectral embedded clustering based on structured graph learning for large-scale hyperspectral image," *IEEE Geosci. Remote Sens. Lett.*, vol. 19, 2022, Art. no. 5501705.

[34] J. M. Murphy and M. Maggioni, "Unsupervised clustering and active learning of hyperspectral images with nonlinear diffusion," *IEEE Trans. Geosci. Remote Sens.*, vol. 57, no. 3, pp. 1829–1845, Mar. 2019.

[35] T. Li, Y. Cai, Y. Zhang, Z. Cai, and X. Liu, "Deep mutual information subspace clustering network for hyperspectral images," *IEEE Geosci. Remote Sens. Lett.*, vol. 19, 2022, Art. no. 6009905.

[36] Y. Kong, Y. Cheng, C. P. Chen, and X. Wang, "Hyperspectral image clustering based on unsupervised broad learning," *IEEE Geosci. Remote Sens. Lett.*, vol. 16, no. 11, pp. 1741–1745, Nov. 2019.

[37] R. Wang, N. Feiping, W. Zhen, H. Fang, and L. Xuelong, "Scalable graph-based clustering with nonnegative relaxation for large hyperspectral image," *IEEE Trans. Geosci. Remote Sens.*, vol. 57, no. 10, pp. 7352–7364, Oct. 2019.

[38] K. R. Shahi, M. Khodadadzadeh, L. Tusa, P. Ghamisi, and R. Gloaguen, "Hierarchical sparse subspace clustering (HESSC): An automatic approach for hyperspectral image analysis," *Remote Sens.*, vol. 12, no. 15, 2020, Art. no. 2421.

[39] A. Hassanzadeh, A. Kaarna, and T. Kauranne, "Sequential spectral clustering of hyperspectral remote sensing image over bipartite graph," *Appl. Soft Comput.*, vol. 73, pp. 727–734, 2018.

[40] N. Huang, L. Xiao, and Y. Xu, "Bipartite graph partition based coclustering with joint sparsity for hyperspectral images," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 12, no. 12, pp. 4698–4711, Dec. 2019.

[41] Y. Ding et al., "Unsupervised self-correlated learning smoothy enhanced locality preserving graph convolution embedding clustering for hyperspectral images," *IEEE Trans. Geosci. Remote Sens.*, vol. 60, 2022, Art. no. 5536716.

[42] Y. Zhao, Y. Yuan, and Q. Wang, "Fast spectral clustering for unsupervised hyperspectral image classification," *Remote Sens.*, vol. 11, no. 4, 2019, Art. no. 399.

[43] Y. Zhang, X. Wang, X. Jiang, and Y. Zhou, "Robust dual graph self-representation for unsupervised hyperspectral band selection," *IEEE Trans. Geosci. Remote Sens.*, vol. 60, 2022, Art. no. 5538513.

[44] D. Zhu, B. Du, Y. Dong, and L. Zhang, "Target detection with spatial-spectral adaptive sample generation and deep metric learning for hyperspectral imagery," *IEEE Trans. Multimedia*, vol. 35, pp. 6538–6550, 2023.

[45] M.-Y. Liu, O. Tuzel, S. Ramalingam, and R. Chellappa, "Entropy rate superpixel segmentation," in *Proc. IEEE Comput. Vis. Pattern Recognit.*, 2011, pp. 2097–2104.

[46] X. Zhang, X. Jiang, J. Jiang, Y. Zhang, and Z. Cai, "Spectral–spatial and superpixelwise PCA for unsupervised feature extraction of hyperspectral imagery," *IEEE Trans. Geosci. Remote Sens.*, vol. 60, 2022, Art. no. 5502210.

[47] F. Nie, C.-L. Wang, and X. Li, "K-multiple-means: A multiple-means clustering method with specified k clusters," in *Proc. ACM SIGKDD Int. Conf. Knowl. Discov. Data Mining*, 2019, pp. 959–967.

[48] Z. Fu et al., "One-step low-rank representation for clustering," in *Proc. ACM Int. Conf. Multimedia*, 2022, pp. 2220–2228.

[49] F. R. Chung, *Spectral Graph Theory*, vol. 92. Providence, RI, USA: American Mathematical Soc., 1997.

[50] K. Fan, "On a theorem of Weyl concerning eigenvalues of linear transformations I," *Proc. Nat. Acad. Sci.*, vol. 35, no. 11, pp. 652–655, 1949.

[51] F. Nie, X. Wang, and H. Huang, "Clustering and projected clustering with adaptive neighbors," in *Proc. ACM SIGKDD Int. Conf. Knowl. Discov. Data Mining*, 2014, pp. 977–986.

[52] X. Chen, Y. Zhang, X. Feng, X. Jiang, and Z. Cai, "Spectral-spatial superpixel anchor graph-based clustering for hyperspectral imagery," *IEEE Geosci. Remote Sens. Lett.*, vol. 20, 2023, Art. no. 5507405.

[53] Y. Zhao and X. Li, "Deep spectral clustering with regularized linear embedding for hyperspectral image clustering," *IEEE Trans. Geosci. Remote Sens.*, vol. 61, 2023, Art. no. 5509311.

[54] J. Jiang, J. Ma, and X. Liu, "Multilayer spectral–spatial graphs for label noisy robust hyperspectral image classification," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 33, no. 2, pp. 839–852, Feb. 2022.

[55] R. Achanta et al., "SLIC superpixels compared to state-of-the-art superpixel methods," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 34, no. 11, pp. 2274–2282, Nov. 2012.

[56] J. Chen, Z. Li, and B. Huang, "Linear spectral clustering superpixel," *IEEE Trans. Image Process.*, vol. 26, no. 7, pp. 3317–3330, Jul. 2017.

Yongshan Zhang (Member, IEEE) received the B.Eng. and Ph.D. degrees in computer science from the China University of Geosciences, Wuhan, China, in 2014 and 2019, respectively. She has been a member of the BDSC Laboratory, University of Illinois at Chicago, Chicago, IL, USA, and the VIP Lab, University of Macau, Macau, China. She is currently an Associate Professor with the School of Computer Science, China University of Geosciences. Her research interests include machine learning and remote sensing image analysis.

Guozhu Jiang is currently working toward the M.S. degree with the School of Computer Science, China University of Geosciences, Wuhan, China. His research interests include machine learning and hyperspectral image clustering.

Zhihua Cai received the B.Sc. degree from Wuhan University, Wuhan, China, in 1986, the M.Sc. degree from the Beijing University of Technology, Beijing, China, in 1992, and the Ph.D. degree from the China University of Geosciences, Wuhan, in 2003. He is currently a Faculty Member with the School of Computer Science, China University of Geosciences. He has authored or coauthored more than 200 research papers in journals and international conferences, such as IEEE Transactions on Knowledge and Data Engineering, IEEE Transactions on Cybernetics, *Applied Soft Computing*, *Information Sciences, Knowledge-Based Systems*, and *Knowledge and Information Systems*. His research interests include data mining, machine learning, evolutionary computation, and their applications.

Yicong Zhou (Senior Member, IEEE) received the B.S. degree in electrical engineering from Hunan University, Changsha, China, and the M.S. and Ph.D. degrees in electrical engineering from Tufts University, Medford, MA, USA. He is currently a Professor with the Department of Computer and Information Science, University of Macau, Macau, China. His research interests include image processing, computer vision, machine learning, and multimedia security. Dr. Zhou is a Fellow of SPIE (the Society of Photo-Optical Instrumentation Engineers) and was recognized as one of "Highly Cited Researchers" in 2020, 2021 and 2023. He is an Associate Editor for IEEE Transactions on Cybernetics, IEEE Transactions on Neural Networks and Learning Systems, and IEEE Transactions on Geoscience and Remote Sensing.