remote sensing

MDPI

Article

Bi-Kernel Graph Neural Network with Adaptive Propagation Mechanism for Hyperspectral Image Classification

Haojie Hu, Yao Ding *, Fang He, Fenggan Zhang, Jianwei Zhao and Minli Yao

Xi'an Research Institute of High Technology, Xi'an 710025, China
* Correspondence: dingyao.88@outlook.com

Abstract: Graph neural networks (GNNs) have been widely applied for hyperspectral image (HSI) classification, due to their impressive representation ability. It is well-known that typical GNNs and their variants work under the assumption of homophily, while most existing GNN-based HSI classification methods neglect the heterophily that is widely present in the constructed graph structure. To deal with this problem, a homophily-guided Bi-Kernel Graph Neural Network (BKGNN) is developed for HSI classification. In the proposed BKGNN, we estimate the homophily between node pairs according to a learnable homophily degree matrix, which is then applied to change the propagation mechanism by adaptively selecting two different kernels to capture homophily and heterophily information. Meanwhile, the learning process of the homophily degree matrix and the bi-kernel feature propagation process are trained jointly to enhance each other in an end-to-end fashion. Extensive experiments on three public data sets demonstrate the effectiveness of the proposed method.

Keywords: graph neural networks; hyperspectral image classification; homophily degree matrix; bi-kernel feature transformation

Citation: Hu, H.; Ding, Y.; He, F.; Zhang, F.; Zhao, J.; Yao, M. Bi-Kernel Graph Neural Network with Adaptive Propagation Mechanism for Hyperspectral Image Classification. *Remote Sens.* **2022**, *14*, 6224. https://doi.org/10.3390/rs14246224

Academic Editor: Javier Marcello

Received: 24 October 2022
Accepted: 6 December 2022
Published: 8 December 2022

1. Introduction

Hyperspectral images captured by hyperspectral sensors can provide a wealth of spectral information to uniquely identify various land-covers according to their reflective spectra. Hence, they have been extensively employed in numerous remote sensing fields, including clustering [1], classification [2], unmixing [3], change detection [4], and target or anomaly detection [5–7]. Among these areas, hyperspectral image classification (HSIC) is a common task and a crucial procedure, referring to categorizing each image pixel into a certain meaningful class, according to the image contents [8].

To date, various approaches have been proposed for HSI classification. Early research primarily relied on traditional pattern recognition techniques, such as the *k*-nearest neighbor classifier [9], support vector machines (SVM) [10,11], and sparse representation [12]. However, these classifiers ignore the sensitivity to spectral fluctuation in raw HSI and solely take into account the original spectral information of pixels in the HSI. In this case, many works have focused on extracting additional discriminative spectral features or investigating spatial–spectral properties of HSI for classification. For example, principal component analysis (PCA) and linear discriminant analysis (LDA) have been applied to reduce redundancy and extract low-dimensional spectral information from HSIs [13,14]. Furthermore, spatial information is often exploited by morphological profiles (EMPs) [15], morphological attribute profiles (APs) [16], Gabor filters [17], and so on. Due to the enhanced spectral and spatial characteristics, the classification performance can be somewhat improved [18].

However, the aforementioned methods are all based on handcrafted characteristics, which heavily rely on professional experience and are quite empirical [19]. To mitigate the limitations of hand-crafted feature design, deep learning techniques have been extensively

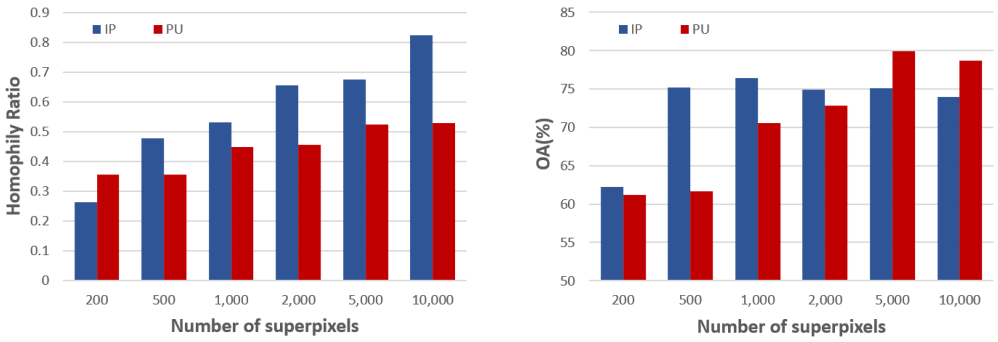
employed in the area of HSIC, through the use of various advanced deep networks. For instance, deep belief networks (DBNs) [20] and recurrent neural networks (RNNs) [21] have been adopted to extract deep features. To further exploit the original spatial structure of the HSI, convolutional neural networks (CNNs) with 2D and 3D convolutions have been extensively used for HSIC. For example, Hong et al. [22] have applied a 2-D CNN to capture spectral–spatial information from various modalities to improve the effectiveness of HSI classification. Hamida et al. [23] have designed a joint spectral and spatial information process through the use of a 3D CNN architecture. A spectral–spatial 3D–2D CNN classification model has been introduced [24], which demonstrated the excellent potential of hybrid networks mixing two- and three-dimensional convolution for the deep extraction of spectral–spatial features. Furthermore, deeper models with advanced networks have been proposed, including capsule networks [25], recursive autoencoders [26], and transformers [27].

Although current CNN-based methods have demonstrated significant effectiveness in the HSIC task, the limitations of the convolutional operation itself hinder further improvement of their performance. First, CNNs commonly possess a large number of training parameters and are prone to over-fitting due to a lack of training data as, unfortunately, there is a widespread problem with small training samples in the remote sensing field. Second, CNNs generally obfuscate the classification boundary as they use a kernel of fixed shape around the central pixel [28]. For these reasons, precise categorization of HSIs is still difficult. Finally, CNNs commonly apply patch-based neighborhoods of samples with fixed sizes as input; however, this approach cannot determine the homophily between pixels within and outside of the patch [29].

Considering the difficulties mentioned above, one possible solution is to design graph-based semi-supervised models that exploit the latent relationships between labeled and unlabeled data. Many GNN-based HSIC methods have recently been proposed for the extraction of features by considering the whole HSI as graph structure data. Normally, the success of GNN-based HSI classification algorithms relies on the propagation capability and the efficiency of the adjacency matrix. The propagation capability is usually achieved using a classical graph neural network model, such as GCN [30], GAT [31], EdgeConv [32], or GraphSage [33]. Typical works that apply these models in HSIC include [34–38]. Meanwhile, the adjacency matrix describes the similarity between two nearby pixels or superpixels. Early GNN-based approaches constructed pixel-level graphs by treating each pixel as a node in the graph [22,34], which can directly propagate information between nearby and distant regions; however, this will result in a vast amount of computation and limits its applicability, due to hundreds of thousands of pixels in an HSI. Fortunately, superpixel, which can effectively characterize the spatial semantic information of surface objects, provides a reasonable way to solve this problem [39–41]. In addition, as the number of labels is implicitly expanded in superpixels, the problem of small samples can be mitigated, to some extent [35]. Therefore, we focus on superpixel-based GNN models in our research.

Although GNNs have revealed remarkable advantages in the task of HSI classification, it has been neglected that GNN-based methods are widely believed to work well when dealing with homophily graphs, and fail to generalize to heterophily graphs when dissimilar nodes are connected [42,43]. Due to the diverse transformations in the graph construction of hyperspectral images, how can we ensure the high homophily of the graph data? As far as we know, previous GNN-based HSIC methods have not considered this problem. Noting that the homophily property can be quantitatively measured by the Homophily Ratio (HR) [44], we were inspired to determine different feature transformations through a learnable kernel, according to the homophily calculation among different local regions in a graph. However, in the HSI classification scenario, a high homophily level cannot easily determine better performance. We know that homophily is only related to the number of superpixels when the way of constructing the graph is determined. As shown on the left of Figure 1, we can see that the homophily level increases as the number of superpixels

increases in both data sets, while the classification performance does not always increase with an increasing number of superpixels. As can be seen from the right of Figure 1, the overall accuracy (OA) suffers a slight decline with a large number of superpixels for both data sets. This is because most previous methods only use the same kernel to transform the features of neighbors; in this way, a large number of superpixels may lose the power to explore local spatial information of the HSI, even if the homophily level is improved. Therefore, we cannot infinitely improve the homophily level of the graph by increasing the number of superpixels. As a result, the heterophily information that exists in the constructed graph cannot be neglected.



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

Figure 1. (**a**) Homophily Ratio and (**b**) overall accuracy (OA) with different number of superpixels on Indian Pines (IP) and Pavia University (PU) data sets.

In this paper, we propose a bi-kernel graph neural network with adaptive propagation mechanism (BKGNN) for HSI classification. In particular, a homophily degree matrix learned from the attribute and topological information is applied to model the homophily and heterophily of the graph, and is further used to adaptively change the propagation process. To avoid smoothing distinguishable features, we use bi-kernels to propagate information on the graph, with one for homophily node pairs and the other for heterophily node pairs. To make the proposed approach more easy to understand, a schematic of BKGNN is displayed in Figure 2. Compared with traditional GNN-based HSI classification algorithms, the main contributions of this paper are as follows:

1. We introduce a novel homophily degree matrix to estimate the homophily and heterophily that widely exist in the constructed graph for HSI. In the process of homophily degree matrix estimation, topological features and attribute features are learned by label propagation (LP) and Multilayer Perception (MLP) through extracting class-aware information. Thus, we can incorporate the homophily and heterophily information into the graph convolution framework.
2. We propose a homophily-guided bi-kernel propagation mechanism, through which we can automatically change the feature propagation process by utilizing both homophily and heterophily information from the graph. To the best of our knowledge, this is the first time that a homophily-guided GNN technique has been applied to the HSI classification task.
3. Extensive experiments on three real-world data sets, i.e., Indian Pines, Pavia University, and Kennedy Space Center, are conducted to validate the performance of the proposed BKGNN both qualitatively and quantitatively. The experimental results demonstrate a significant improvement over previous methods.

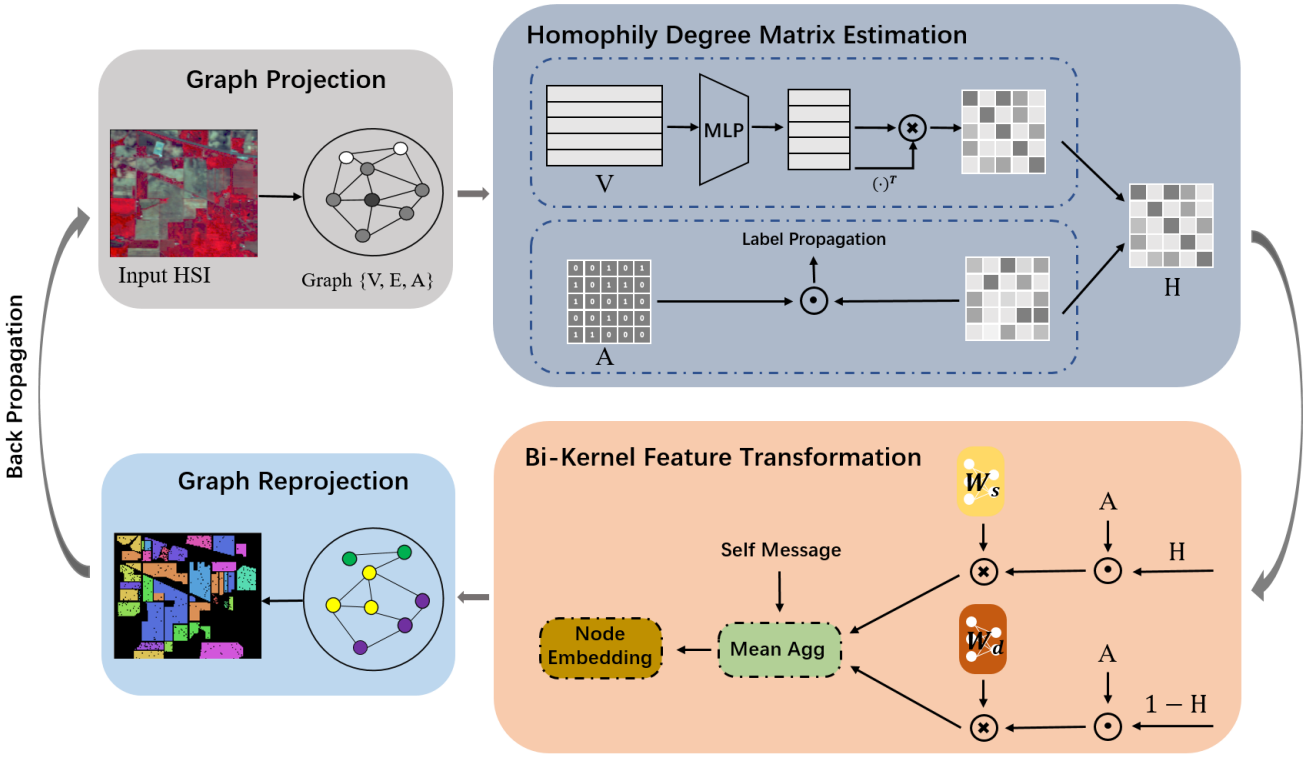


Figure 2. Overview of BKGNN. BKGNN has four main modules: Graph projection, homophily degree matrix estimation, bi-kernel feature transformation, and graph re-projection. The first module maps the original pixel-level HSI data to a superpixel-level graph structure, after which the homophily degree matrix is learned from the attribute and topological information, which is further used to conduct the bi-kernel feature transformation for capturing the similarity between nodes and the dissimilarity between nodes. After graph re-projection, the pixel-level result is obtained.

2. Methodology

In this section, we first provide some preliminaries of our method by reviewing some basic definitions and notation, including calculation of the homophily ratio, and an introduction to GNN and LP. Then the proposed BKGNN is illustrated in detail. The main notation adopted in our manuscript and relevant descriptions are provided in Table 1. All symbols in the article are explained in detail in the corresponding place.

Table 1. Main notation and descriptions.

Notation	Definition	Type	Size
$\boldsymbol{X}$	Original HSI data.	3D matrix	$H \times W \times B$
H, W and B	Height, width, and number of bands in HSI, respectively.	Scalar	1×1
C	Number of classes in HSI.	Scalar	1×1
N	Number of nodes contained in the graph (equal to the number of superpixels).	Scalar	1×1
$\mathcal{G} = (\mathcal{V}, \mathcal{E})$	Graph $\mathcal{G}$ with node set $\mathcal{V}$ and edge set $\mathcal{E}$.	-	-
$\boldsymbol{A}$	Adjacency matrix.	Matrix	$N \times N$
$\boldsymbol{V}$	Attribute matrix.	Matrix	$N \times B$
$\boldsymbol{D}$	Degree matrix of $\boldsymbol{A}$.	Matrix	$N \times N$
$\boldsymbol{Q}$, $\hat{\boldsymbol{Q}}$	Projection matrix and its normalized version.	Matrix	$HW \times B$
$\boldsymbol{H}$	Homophily degree matrix	Matrix	$N \times N$
$\boldsymbol{Z}$	Node embeddings	Matrix	$N \times F$
$\mathcal{S}_i$	The set of pixels in the ith superpixel.	-	-
$\mathbb{N}(\cdot)$	Neighborhood of $\cdot$.	-	-
$\mathcal{T}_{\mathcal{V}}$	Nodes in the training set.	-	-

2.1. Preliminaries

2.1.1. Graph Neural Network (GNN)

Graph neural networks, which operate on graph data, have demonstrated their effectiveness in various graph tasks. Let $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ represent the graph obtained from the original HSI data, where $\mathcal{V} = \{v_1, v_2, \cdots, v_n\}$ is the set of vertices and $\mathcal{E}$ is the set of edges, representing the connectivity between vertices in $\mathcal{V}$. Generally, a GNN follows a message-passing mechanism, which commonly consists of a message aggregation phase and an update phase. During the message passing phase [45], the representation $\boldsymbol{z}_v$ of node v_i is iteratively updated based on message m_v, according to

$$m_v^{l+1} = \sum_{w \in \mathbb{N}(v)} M(\boldsymbol{z}_v^l, \boldsymbol{z}_w^l, e_{vw}), \tag{1}$$

$$\boldsymbol{z}_v^{l+1} = U(\boldsymbol{z}_v^l, m_v^{l+1}), \tag{2}$$

where M is a node feature aggregation function, which is a differential permutation-invariant operation; and U is the vertex message update function; $\mathbb{N}(v)$ denotes the nodes neighboring v in $\mathcal{G}$; and $\boldsymbol{z}_v^l$ and z_v^{l+1} are the representation vectors of node v at the lth and $(l+1)$th layers, respectively.

These two functions (i.e., M and U) can take a variety of forms [46]. Concretely, the aggregation function can be a mean aggregator, a max-pooling aggregator, an attention aggregator, or an LSTM aggregator. Meanwhile, the update function is usually achieved by a multi-layer perceptron or a gated network. For example, the GCN model [30] uses a message function $M(\boldsymbol{z}_v^l, \boldsymbol{z}_w^l) = c_{vw}\boldsymbol{z}_w^l$, where $c_{vw} = (deg(v)deg(w))^{-1/2}A_{vw}$. The vertex update function is $U(\boldsymbol{z}_v^l, m_v^{l+1}) = ReLU(W^l m_v^{l+1})$. The GNN-based HSIC approach is essentially a semi-supervised node classification task. Let $\boldsymbol{Y} \in \{0,1\}^{N \times C}$ denote the labels of nodes, where C is the number of classes, while only the first m nodes ($0 < m \ll n$) have labels $\boldsymbol{Y}^L$. The objective is to learn a predictive function to infer the missing labels $\boldsymbol{Y}^U$ for the remaining $n - m$ nodes.

2.1.2. Homophily in Graphs

As we need to improve the graph convolution operation based on the homophily of the graph, we first need to determine how to measure the degree of homophily in graphs. We use the homophily ratio to measure the overall homophily level in a graph, which counts the fraction of edges connecting nodes that have the same labels. Formally, the homophily ratio is defined as:

$$h = \frac{1}{|\mathcal{E}|} \sum_{(v_i, v_j) \in \mathcal{E}} \mathbf{1}(y_i = y_j), \tag{3}$$

where $|\mathcal{E}|$ denotes the number of edges in the graph, and $\mathbf{1}$ is the indicator function. In accordance with the definition, we have $h \in [0, 1]$. A graph with high homophily ratio is considered to be highly homophilous. Correspondingly, the node-level homophily is defined as

$$h_i = \frac{1}{|\mathbb{N}(v_i)|} \sum_{v_j \in \mathbb{N}(v_i)} \mathbf{1}(y_i = y_j), \tag{4}$$

where $|\mathbb{N}(v_i)|$ is the size of the neighbor set $\mathbb{N}(v_i)$. Compared with the homophily ratio which is regarded as a global property in the whole graph, the node-level homophily focus on the local regions in a graph, and there may be different levels of homophily among different local regions in a homophily graph.

It is difficult to estimate the homophily level directly from node labels, as there are only a scarce number of nodes with labels in a semi-supervised task. We introduce a

matrix $\boldsymbol{H} \in \mathbb{R}^{n \times n}$ with its ij^{th} element defined as the possibility that the corresponding two points (i.e., v_i and v_j) belong to the same category. Note that $\boldsymbol{H}_{ij} = 1$ if v_i and v_j have the same label, and $\boldsymbol{H}_{ij} = 0$ if v_i and v_j have no edge connection. The matrix $\boldsymbol{H}$ is called the homophily degree matrix in our algorithm, which will be discussed in detail in Section 2.4.

2.1.3. Label Propagation (LP) Algorithm

In the process of calculating the homophily degree matrix, we use LP to estimate the homophily degree between node pairs from the topological space. Let $\boldsymbol{Y}^0$ represent the initial label matrix $\boldsymbol{Y}$. The rows of $\boldsymbol{Y}^0$ corresponding the labeled nodes are one-hot indicator vectors, and the unlabeled nodes are zero vectors. Label propagation is often applied to infer pseudo-labels for unlabeled nodes based on $\boldsymbol{Y}^0$. Assuming that neighboring nodes are more likely to have the same label, LP propagates labels along the edges iteratively. It is feasible to specify the formulation of the LP algorithm in iteration l as follows:

$$\boldsymbol{Y}^{(l)} = \boldsymbol{D}^{-1}\boldsymbol{A}\boldsymbol{Y}^{(l-1)}, \tag{5}$$

where $\boldsymbol{A} \in \mathbb{R}^{N \times N}$ denotes the adjacency matrix, whose elements a_{ij} denote the non-negative pairwise similarity between v_i and v_j; $\boldsymbol{D}$ is the diagonal matrix of $\boldsymbol{A}$, with entries $d_{ii} = \sum_j a_{ij}$; and $\boldsymbol{Y}^{(l)}$ is the pseudo-label matrix in iteration l. As the adjacency matrix $\boldsymbol{A}$ is a sparse matrix with non-zero elements on nearest neighbors, the true label information will propagate from each labeled example to its neighbors in each iteration.

2.2. Overall Framework

As shown in Figure 2, the proposed BKGNN consists of four main modules: graph projection, homophily degree matrix estimation, bi-kernel feature transformation, and graph re-projection. The first module maps the original pixel-level HSI data $\boldsymbol{X} \in \mathbb{R}^{H \times W \times B}$ to a superpixel-level graph structure $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, where $\mathcal{V}$ denotes the group of obtained superpixels, which can be represented by an attribute matrix $\boldsymbol{V} \in \mathbb{R}^{N \times B}$, while $\mathcal{E}$ can be represented by a spatial adjacency matrix $\boldsymbol{A} \in \mathbb{R}^{N \times N}$. We apply a multi-layer perceptron (MLP) and the label propagation (LP) technique to extract the homophily information from the attribute space and topological space, respectively, and define the whole homophily degree matrix $\boldsymbol{H} \in \mathbb{R}^{N \times N}$ based on these two types of information. After that, the bi-kernel feature transformation trains $\boldsymbol{W}_s$ and $\boldsymbol{W}_d$ to capture the similarity between nodes and the dissimilarity between nodes. The homophily degree matrix obtained from the former module is utilized to combine these two processes of message passing, producing the superpixel-level node embedding. Furthermore, self-messages are added into the procedure of computing the node embedding, in order to reduce over-smoothing. The enhanced superpixel features are then projected to pixel features by the last module, which is used to perform pixel-wise classification. In the proposed BKGNN, the cross-entropy loss function is utilized to minimize the differences between the predicted labels and the ground-truth of training samples. The details of these modules are described in the following.

2.3. Graph Projection and Re-Projection

To reduce the computational complexity while maintaining the local structure of the HSI, GNN-based HSI classification models frequently operate on superpixel-based nodes, rather than pixel-based nodes. Therefore, we establish the relationship between pixel-level HSI data and the superpixel-level graph structure. For this purpose, we pre-process the entire image into a number of spatially linked superpixels using the simple linear iterative clustering (SLIC) method. Considering each superpixel as a graph node, each superpixel's average spectral signature serves as the node feature, and edge connections are established between neighboring nodes. Consequently, the superpixel-level graph structure $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ can be obtained.

Specifically, the graph projection assigns the original HSI data $\boldsymbol{I}$ to a set of nodes $\mathcal{V}$ and determines the corresponding feature matrix $\boldsymbol{V}$ through matrix multiplication, as follows:

$$\boldsymbol{V} = \hat{\boldsymbol{Q}}^T \text{Flatten}(\boldsymbol{X}), \tag{6}$$

where $\hat{\boldsymbol{Q}}$ is the normalized $\boldsymbol{Q}$ by column (i.e., $\hat{\boldsymbol{Q}}_{i,j} = \boldsymbol{Q}_{i,j} / \sum_m \boldsymbol{Q}_{m,j}$), and Flatten$(\cdot)$ represents flattening the HSI data by the spatial dimension. The association matrix $\boldsymbol{Q} \in \mathbb{R}^{HW \times N}$ is introduced by SLIC, which is defined as

$$\boldsymbol{Q}_{i,j} = \begin{cases} 1, & if \quad \boldsymbol{x}_i \in \mathcal{S}_j, \\ 0, & \text{otherwise}, \end{cases} \tag{7}$$

where $\mathcal{S}_i$ is the ith superpixel that consists of several homogeneous pixels.

As for the edge set $\mathcal{E}$, two superpixels that share a common boundary are considered to have an edge connection. Thus, the adjacency matrix $\boldsymbol{A}$ related to $\mathcal{E}$ can be defined as

$$\boldsymbol{A}_{i,j} = \begin{cases} 1, & if \quad \text{if } \mathcal{S}_i \text{ and } \mathcal{S}_j \text{ are adjacent} \\ 0, & \text{otherwise}, \end{cases} \tag{8}$$

where $\mathcal{S}_i$ and $\mathcal{S}_j$ denote the ith and jth superpixels, respectively.

We use graph re-projection to convert the generated features back to the original coordinate space for pixel-wise classification. The transformed node features $\widetilde{\boldsymbol{V}} \in \mathbb{R}^{N \times C}$ and the assignment matrix $\boldsymbol{Q}$ are used as inputs for the graph re-projection process, which outputs the appropriate 3D feature map. This operation is defined as

$$\widetilde{\boldsymbol{X}} = \text{Reshape}(\boldsymbol{Q}\widetilde{\boldsymbol{V}}), \tag{9}$$

where Reshape$(\cdot)$ denotes restoring the spatial dimension of the flattened data.

2.4. Homophily Degree Matrix Calculation

Conventional GNN models have fundamental homophily assumptions and, as such, are not suited for heterophily graphs [42]. As shown in Figure 1, even if we can reduce the degree of heterogeneity of the graph to some extent by setting an appropriate superpixel size, the heterogeneity of the graph cannot be ignored. To solve this problem, we introduce a homophily degree matrix $\boldsymbol{H} \in \mathbb{R}^{N \times N}$, with its (i, j)th element defining the extent to which the ith and jth nodes belong to the same class. However, it is difficult to calculate the homophily degree directly from node labels, as only a small number of labels are known in the context of the semi-supervised task. In order to fill this gap, we estimate the soft labels for unlabeled nodes from the attribute space and topological space.

To utilize the attribute space of the graph, we apply a graph-agnostic multi-layer perceptron (MLP) to generate soft labels from the original node attributes. The l^{th} layer of the MLP is defined as:

$$\boldsymbol{Z}^{(l)} = \sigma(\boldsymbol{Z}^{(l-1)}\boldsymbol{W}^{(l)}), \tag{10}$$

where $\boldsymbol{W}^{(l)}$ is the learnable weight matrix, and $\sigma(\cdot)$ is an activation function. Denoting by $\boldsymbol{Z}$ the output of MLP for several iterations, the soft assignment matrix $\boldsymbol{B} \in \mathbb{R}^{N \times C}$ can be obtained, using a softmax operation, as follows:

$$\boldsymbol{B} = \text{softmax}(\boldsymbol{Z}). \tag{11}$$

Let Θ_m denote all the parameters of the MLP. Then, the optimal Θ_m^* can be obtained by minimizing the loss function:

$$\Theta_m^* = \underset{\Theta}{\operatorname{argmin}} \mathcal{L}_{mlp} = \underset{\Theta_m}{\operatorname{argmin}} \sum_{v_i \in \mathcal{T}_\mathcal{V}} J(\boldsymbol{b}_i, \boldsymbol{y}_i), \tag{12}$$

where $\boldsymbol{b}_i$ is the predicted soft label of node v_i, $\boldsymbol{y}_i$ is the ground-truth label of v_i, $\mathcal{T}_\mathcal{V}$ denotes the nodes in the training set, and $J(\cdot)$ is the cross-entropy.

In terms of the topological space, we further apply the label propagation (LP) technique to estimate the soft labels. We generalize classic LP with a learnable weight matrix $\boldsymbol{T} \in \mathbb{R}^{N \times N}$. Similar to Equation (5), the resulting generalized LP in the lth iteration is defined as:

$$\boldsymbol{Y}^{(l)} = \hat{\boldsymbol{D}}^{-1}(\boldsymbol{A} \odot \boldsymbol{T})\boldsymbol{Y}^{(l-1)}, \tag{13}$$

where $\hat{\boldsymbol{D}}$ is the diagonal matrix for matrix $\boldsymbol{A} \odot \boldsymbol{T}$. Similar to MLP optimization, the optimal weight matrix $\boldsymbol{T}$ is learned by

$$\boldsymbol{T}^* = \underset{\boldsymbol{T}}{\operatorname{argmin}} \mathcal{L}_{lp} = \underset{\boldsymbol{T}}{\operatorname{argmin}} \sum_{v_i \in \mathcal{T}_\mathcal{V}} J(\boldsymbol{y}_i^{lp}, \boldsymbol{y}_i). \tag{14}$$

Finally, the homophily degree matrix $\boldsymbol{H}$ is estimated from the attribute space and topological space with learnable parameters, as follows:

$$\boldsymbol{H} = \alpha \boldsymbol{S} + \beta \boldsymbol{T}, \tag{15}$$

where α and β are hyper-parameters, and $\boldsymbol{S}$ is defined as $\boldsymbol{B}\boldsymbol{B}^T$. Note that the obtained $\boldsymbol{H}$ is a dense matrix, as $\boldsymbol{S}$ calculates the homophily degree between any node pair. We filter the homophily degree that is not involved in the propagation process.

2.5. Bi-Kernel Feature Transformation

In this subsection, we first present the motivation for bi-kernel feature transformation, in terms of generalization ability. Specifically, we chose the complexity measure of Consistency of Representations (champion of the NIPS 2020 Competition on generalization measure) to estimate the generalization ability, defined as

$$\Gamma = \frac{1}{|\mathcal{C}|} \sum_{i=1}^{|\mathcal{C}|} \max_{i \neq j} \frac{\mathcal{O}_i + \mathcal{O}_j}{\mathcal{M}_{i,j}}, \tag{16}$$

where $\mathcal{C}$ is the set of classes, $\mathcal{C}_i, \mathcal{C}_j \in \mathcal{C}$ are two different classes, $\mathcal{O}_i = \left(\mathbb{E}_{v_k \sim \mathcal{C}_i}\left(\left|\boldsymbol{z}_k - \mu_{\mathcal{C}_i}\right|^p\right)\right)^{\frac{1}{p}}$ is the intra-class variance of class $\mathcal{C}_i$, and $\mathcal{M}_{ij} = \|\mu_{\mathcal{C}_i} - \mu_{\mathcal{C}_j}\|_p$ is the inter-class variance between $\mathcal{C}_i$ and $\mathcal{C}_j$. A higher value of Γ indicates lower generalization ability. For simplicity, we ignore non-linear activation in the GNN and only consider the binary classification problem. Let P_i denote, for a center node belonging to the ith class, the probability of its neighbors belonging to the same category. Then, the Consistency of Representations has an important property, as follows [47]:

$$\Gamma \geq \frac{c}{|(P_0 + P_1 - 1)| \left\|\mathbf{W}\left(\mu_{\mathcal{C}_0} - \mu_{\mathcal{C}_1}\right)\right\|}, \tag{17}$$

where $c \in \mathbb{R}^+$ is a constant. If $|P_0 + P_1 - 1| \to 0$, the lower bound of $\Gamma \to \infty$ and, hence, the model will lose its generalization ability. This indicates that, if there are a similar number of homophily neighbors and heterophily neighbors for graph nodes, then GNN will smooth the outputs from different classes and lose discrimination ability.

In reality, the heterophily information is widely distributed in the constructed graph structure for HSI, and we cannot extract homophily and heterophily information using

only one kernel of GNN; this is because using only a single kernel in the GNN will result in smoothing of the distinguishable features of different labels. To tackle this problem, we apply two kernels in our model; in particular, we use one kernel for homophily node pairs and another for heterophily pairs. Thus, the lower bound of Γ will be changed to $\frac{c}{\left\|(P_1\mathbf{W}_s+(P_0-1)\mathbf{W}_d)\left(\mu_{\mathcal{C}_0}-\mu_{\mathcal{C}_1}\right)+(P_0-P_1)(\mathbf{W}_s-\mathbf{W}_d)\mu_{\mathcal{C}_0}\right\|}$, where $\boldsymbol{W}_s$ is the kernel for homophily nodes and $\boldsymbol{W}_d$ is the kernel for heterophily nodes. It can be seen that $\boldsymbol{W}_s$ and $\boldsymbol{W}_d$ can adjust the relation between P_1 and $P_0 - 1$, thus avoiding the $|P_1 + P_0 - 1|$ term. Meanwhile, extra distinguishability is provided, even if the original features lack discrimination (i.e., $\|\mu_{\mathcal{C}_0} - \mu_{\mathcal{C}_1}\| \to 0$). Inspired by the work [47], we apply a bi-kernel feature transformation to tackle this problem. Specifically, we use one kernel for homophily node pairs and the other for heterophily pairs. During the propagation process, we adaptively adjust the weights between the kernels, according to the homophily degree matrix. The formal form of the feature propagation process in iteration l is given by

$$\boldsymbol{Z}^{(l)} = \sigma\Big(\boldsymbol{Z}^{(l-1)}\boldsymbol{W}_e + \boldsymbol{D}^{-1}\boldsymbol{A} \odot \boldsymbol{H}\boldsymbol{Z}^{(l-1)}\boldsymbol{W}_s + \boldsymbol{D}^{-1}\boldsymbol{A} \odot (1 - \boldsymbol{H})\boldsymbol{Z}^{(l-1)}\boldsymbol{W}_d\Big), \tag{18}$$

where $\boldsymbol{W}_e$, $\boldsymbol{W}_s$, and $\boldsymbol{W}_d$ are learnable parameters for exploiting information from the ego-representation, homophily node pairs, and heterophily node pairs, respectively; $\boldsymbol{Z}^0 = \boldsymbol{V}$ denotes the original node attributes; and σ is the activation function.

2.6. Optimization Objective

The cross-entropy loss function is a frequently used optimization objective for the HSI classification problem, in order to minimize the discrepancy between the predicted labels and the actual labels of the training samples. After graph re-projection, we map the superpixel-level graph features into pixel-level feature space. The final output of our network is defined as

$$\hat{\boldsymbol{Y}} = \text{softmax}(\text{Reprojection}(\boldsymbol{Z})). \tag{19}$$

Then, the cross-entropy loss function can be written as

$$\mathcal{L}_{ce} = -\sum_{s \in \boldsymbol{Y}_{labeled}} \sum_{f=1}^{C} \boldsymbol{Y}_{sf} ln \hat{\boldsymbol{Y}}_{sf}, \tag{20}$$

where the label matrix is represented by $\boldsymbol{Y}$, the number of object classes is C, and the probability that the sth pixel belongs to the fth class is indicated by $\hat{\boldsymbol{Y}}_{sf}$.

Noting that the homophily degree matrix is learned from MLP and LP, we combine the estimates of the homophily degree matrix in an end-to-end fashion. Let Θ_g denote all the parameters of the bi-kernel feature transformation. The final optimization objective can be given by

$$\Theta_g^*, \Theta_m^*, T^* = \underset{\Theta_g, \Theta_m, T}{\text{argmin}} \mathcal{L}_{ce} + \lambda \mathcal{L}_{mlp} + \gamma \mathcal{L}_{lp}, \tag{21}$$

where λ and γ are regularization parameters. It is also worth noting that the homophily degree matrix is learned from attribute and topological information by minimizing both $\mathcal{L}_{mlp}$ and $\mathcal{L}_{LP}$, which is further used to conduct bi-kernel feature transformation by minimizing $\mathcal{L}_{ce}$. In turn, the feature transformation process can help to learn a better homophily degree matrix. Therefore, these two processes are trained jointly to enhance each other. The implementation details of BKGNN are summarized in Algorithm 1.

Algorithm 1: BKGNN.

Input: Original HSI data $\boldsymbol{X}$; training labels $\boldsymbol{Y}$; number of superpixels N; number of iterations T; learning rate η; hyper-parameters α, β, λ, and γ.

1: Superpixel segmentation via SLIC.

2: Calculate the attribute matrix $\boldsymbol{V}$ and the adjacency matrix $\boldsymbol{A}$ according to Equations (6) and (8), respectively.

for $t = 1$ to T **do**

3: Perform MLP and LP according to Equations (10) and (13).

4: Calculate the homophily degree matrix $\boldsymbol{H}$ according to Equation (15).

5: Update the outputs after two layers of bi-kernel feature transformation according to Equation (16).

6: Graph reprojection according to Equation (9).

7: Calculate the overall error over all labeled instances according to Equation (21), and update the weight matrices using Adam gradient decent.

end for

8: Conduct label prediction based on the trained network.

Output: Predicted label for each pixel.

2.7. Computational Complexity

In this subsection, we discuss the computational complexity of our method. Suppose the embedding of each node is an F-dimensional feature vector, and $\|\boldsymbol{A}\|_0$ denotes the number of non-zero entries of the adjacency matrix $\boldsymbol{A}$. For layer-wise MLP and LP, the computational complexity is $O(NF^2)$ and $O(\|\boldsymbol{A}\|_0 C)$, respectively. As for the module of bi-kernel feature transformation, the time cost is $O(NF^2 + \|\boldsymbol{A}\|_0 F)$ for feature propagation, and $O(HWNF)$ for graph re-projection. Therefore, the overall time complexity of BKGNN is $O((NF^2 + \|\boldsymbol{A}\|_0 C + \|\boldsymbol{A}\|_0 F + HWNF)T)$, where T represents the number of iterations. Note that, as the number of superpixels N is much smaller that the number of pixels in HSI (i.e., HW), the time cost of our method can be greatly reduced through the use of superpixel segmentation.

3. Experiments

3.1. Data Set Description

Three widely used HSI data sets were adopted to evaluate the performance of our proposed algorithm. The details of each data set are provided in the following.

3.1.1. Indian Pines (IP)

This data set was gathered by the Airborne Visible/Infrared Imaging Spectrometer (AVIRIS) sensor at a test site in northwest Indiana. It consists of 224 spectral reflectance bands in the wavelength range of 400–2500 nm. A total of 200 bands were reserved, after removing 24 invalid and corrupted bands. The image has a spatial size of 145 × 145 pixels and includes 16 mutually exclusive vegetation classes. The spatial resolution is 20 m per pixel. A false color composite, as well as detailed category descriptions and ground-truth map, are shown in Figure 3.

3.1.2. Pavia University (PU)

This data set was acquired by the Reflection Optical System Imaging Spectrometer (ROSIS) sensor during a flight campaign over the University of Pavia campus in northern Italy. Pavia University is a 610 × 340 pixels image with 103 spectral bands in the wavelength range of 430–860 nm. The geometric resolution is 1.3 m. This data set includes nine urban land-cover categories. A false color composite, as well as detailed category descriptions and ground truth map are shown in Figure 4.

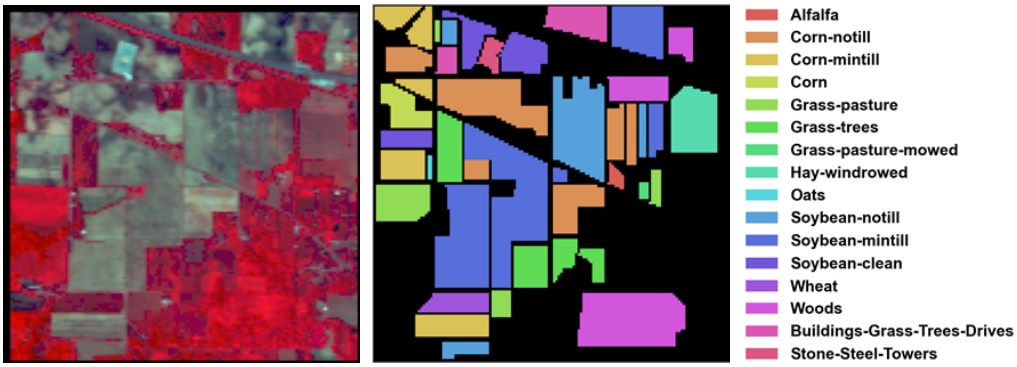


Figure 3. IP data set: (**a**) False color image; (**b**) Ground truth map; and (**c**) Color bar.

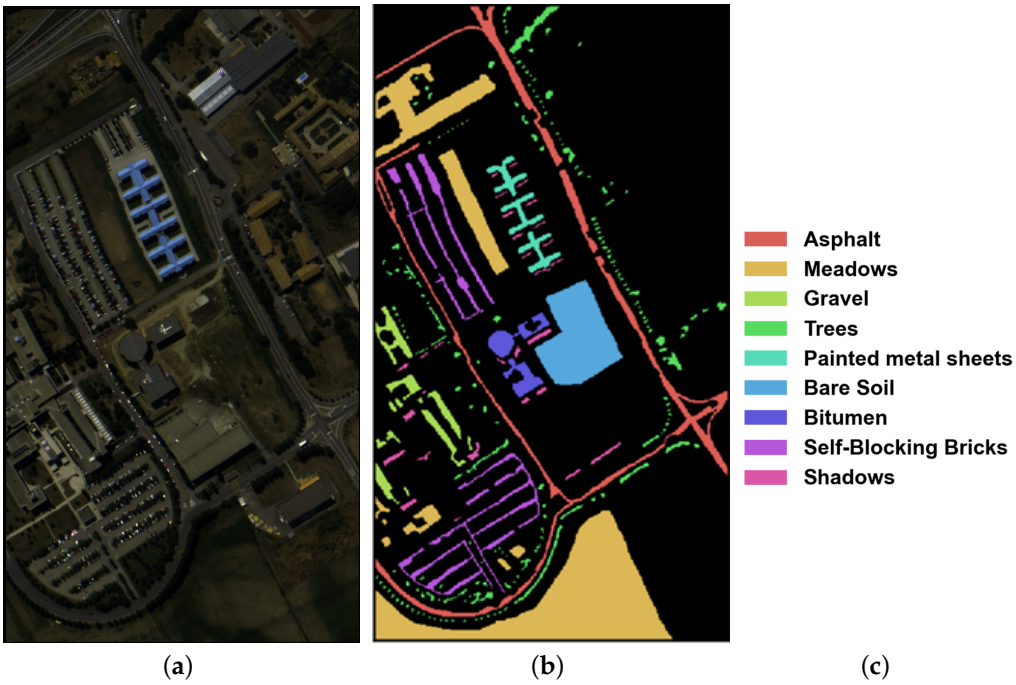


Figure 4. PU data set: (**a**) False color image; (**b**) Ground truth map; and (**c**) Color bar.

3.1.3. Kennedy Space Center

This data set was also acquired by AVIRIS in 1996, and has a wavelength range of 400–2500 nm. The image has size of 512 × 614 pixels, and 176 bands remained after removing some low signal-to-noise ratio bands. The KSC data set includes 5202 labeled samples, with 13 upland and wetland categories. The spatial resolution is 18 m per pixel. The false color composite, with detailed category descriptions and ground truth map are shown in Figure 5.

3.2. Experimental Settings

3.2.1. Implementation Details

In order to quantify the performance of different HSIC methods, four widely used metrics were calculated, including overall accuracy (OA), Per-Class Accuracy (PA), average accuracy (AA), and Kappa coefficient. Specifically, OA is computed as the fraction of samples that are differentiated correctly, PA is the accuracy for each class, AA is calculated as the average of all per-class accuracies, and kappa coefficient is a robustness measurement considering the degree of agreement.

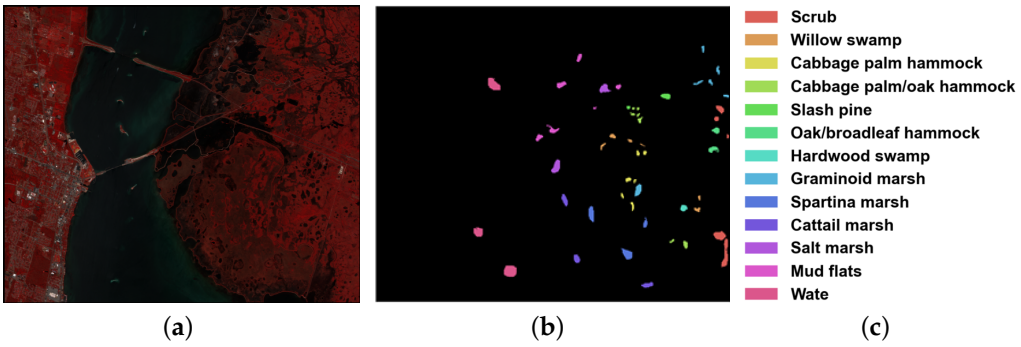


Figure 5. KSC data set: (**a**) False color image; (**b**) Ground truth map; and (**c**) Color bar.

Regarding training details, all data sets were trained with 30 labeled pixels in each class, or 15 labeled pixels if there were less than 30 samples in the corresponding class. Network optimization was carried out using the Nesterov Adam algorithm. In addition, the learning rate and the number of training epochs were set to 0.001 and 1000, respectively. The number of superpixels N was set to 500 for the IP data set, and 1000 for the other two data sets. As for the hardware environment, our experiments were implemented in PyTorch and run on a Windows 10 machine equipped with a 3.80 GHz i7-10700K CPU, 32 GB of main memory, and an RTX 3090 GPU.

3.2.2. Compared Methods

For comparison, a number of state-of-the-art baseline methods were selected, including two conventional methods (SVM-RBF [48] and MBCTU [49]), three CNN-based methods (1D CNN [50], 2D CNN [51], and 3D CNN [52]), and three GNN-based methods (NLGCN [53], GSAGE [19], and DARMA [18]). The parameter settings for these competitors are given in the following.

1. SVM-RBF: The value of γ (the spread of the RBF kernel) and C (controlling the magnitude of penalization) is searched in the range of $\gamma = 2^{-3}, 2^{-3}, \cdots, 2^{4}$ and $C = 2^{-2}, 2^{-1}, \cdots, 2^{4}$.
2. MBCTU: MBCTU is actually a random forest classifier that performs color-texture feature extraction based on the selected spectral bands. The bands are selected according to their feature importance computed by another random forest classifier.
3. 1D CNN: This architecture is constructed by one convolutional layer with 20 kernels, one max pooling layer, a ReLU activation layer, and two full connection layers.
4. 2D CNN: A semi-supervised classification model, consisting of one convolutional layer with a 3×3 filter, one max pooling layer, and followed by three decoding layers. Each decoding layer is made up of one full connection layer and one normalization layer.
5. 3D CNN: The 3D CNN model contains two convolution layers and a fully connected layer. Each convolutional layer is followed by ReLU activation layer and their kernel sizes are $3 \times 3 \times 7$ and $3 \times 3 \times 3$.
6. NLGCN: This network applies two graph convolutional layers by incorporating a graph learning procedure.
7. GSAGE: Graph convolution is achieved by graph sampling and aggregation, and the second-order nearest neighbor of the target node is taken into account.
8. DARMA: A superpixel-level GNN model which is composed of three convolutional blocks. Each block consists of an ARMA graph convolutional layer, a ReLU activation layer, and a normalization layer.

3.3. Experimental Results

Tables 2–4 provide information about per-class accuracies, OAs, AAs, and kappa coefficients obtained by various classification methods on the three data sets. All of the reported results were based on an average of 10 training sessions, in order to avoid bias caused by random sampling, and the top results are bolded. As is shown in the results, the two traditional methods (SVM-RBF and MBCTU) obtained similar classification results on IP and PU data sets. Due to the powerful learning ability of DL techniques, the 1D CNN, 2D CNN, and 3D CNN models outperformed the traditional classifiers. Unlike 1D CNN and 2D CNN, the 3D CNN was able to extract both spatial and spectral information at the same time, thus achieving higher classification accuracies. The Superpixel-level GNN approaches (DARMA and BKGNN), outperformed the classical machine learning, deep learning, and pixel-level GNN models. One probable explanation is that superpixel-level GNN methods exploit the latent relationship between different areas by constructing graph structures in HSI to boost classification performance, and meanwhile preserving the local spectral–spatial information through superpixel segmentation. Moreover, our method exceeded DARMA by a substantial margin on the first two data sets. This observation revealed that, compared to previous superpixel-level GNN methods, the proposed model can achieve better performance by performing adaptive feature propagation under the guidance of the homophily degree matrix. This analysis indicates the superiority of BKGNN.

Table 2. Accuracy comparison for the IP data set. Bold numbers indicate the best performance.

Class No.	Conventional Classifiers		CNN-Based Methods			GNN-Based Methods			
	SVM-RBF	MBCTU	1D CNN	2D CNN	3D CNN	NLGCN	GSAGE	DARMA	BKGNN
1	87.50 ± 5.10	95.83 ± 2.95	96.25 ± 4.15	97.50 ± 4.15	99.38 ± 1.87	98.75 ± 2.50	**100.00 ± 0.00**	**100.00 ± 0.00**	**100.00 ± 0.00**
2	55.74 ± 3.99	47.76 ± 0.65	63.21 ± 6.78	58.06 ± 9.13	66.82 ± 7.58	74.93 ± 8.22	76.02 ± 6.53	80.07 ± 6.15	**88.54 ± 2.34**
3	58.66 ± 4.32	68.17 ± 4.25	69.05 ± 7.37	49.59 ± 6.16	58.53 ± 5.04	78.56 ± 3.59	70.30 ± 9.86	94.07 ± 3.32	**94.68 ± 3.18**
4	70.53 ± 8.15	98.87 ± 0.91	86.04 ± 6.76	83.14 ± 5.02	84.69 ± 4.50	92.08 ± 3.73	97.49 ± 1.61	99.61 ± 0.36	**100.00 ± 0.00**
5	84.32 ± 4.67	71.52 ± 7.19	73.95 ± 16.58	77.15 ± 6.78	87.24 ± 3.67	93.11 ± 2.17	91.30 ± 9.10	95.01 ± 2.12	**96.25 ± 2.60**
6	90.61 ± 2.13	86.68 ± 0.53	91.27 ± 3.62	95.61 ± 2.48	91.87 ± 3.19	96.96 ± 2.33	97.71 ± 2.60	96.51 ± 0.71	**98.83 ± 1.21**
7	90.74 ± 6.92	**100.00 ± 0.00**	86.92 ± 7.73	96.15 ± 7.09	98.46 ± 3.08	97.69 ± 4.93	98.46 ± 4.62	96.92 ± 3.77	**100.00 ± 0.00**
8	89.58 ± 3.01	86.53 ± 5.55	94.58 ± 1.78	98.57 ± 0.80	97.05 ± 1.35	99.67 ± 0.30	97.90 ± 4.63	99.51 ± 0.78	**100.00 ± 0.00**
9	96.66 ± 4.71	93.33 ± 9.43	98.00 ± 6.00	**100.00 ± 0.00**	**100.00 ± 0.00**	**100.00 ± 0.00**	82.00 ± 32.80	**100.00 ± 0.00**	**100.00 ± 0.00**
10	70.24 ± 2.86	86.13 ± 6.07	70.91 ± 7.68	75.34 ± 5.23	72.46 ± 8.02	87.58 ± 3.42	86.28 ± 8.77	89.07 ± 2.97	**93.48 ± 3.00**
11	50.59 ± 3.72	60.89 ± 9.64	63.65 ± 7.91	63.56 ± 7.25	66.19 ± 5.15	71.95 ± 4.10	67.46 ± 6.25	86.95 ± 4.36	**92.17 ± 3.74**
12	62.87 ± 6.52	52.46 ± 13.08	70.55 ± 8.75	67.41 ± 9.45	72.02 ± 6.59	89.17 ± 2.64	86.77 ± 7.55	89.66 ± 4.41	**96.45 ± 1.00**
13	96.57 ± 0.93	99.43 ± 0.25	96.63 ± 1.90	99.89 ± 0.34	99.20 ± 0.78	99.49 ± 0.40	99.83 ± 0.26	99.89 ± 0.23	**100.00 ± 0.00**
14	82.37 ± 3.97	86.23 ± 1.03	83.28 ± 11.13	90.95 ± 5.56	90.87 ± 3.51	91.07 ± 2.82	95.51 ± 3.10	97.51 ± 2.67	**98.17 ± 1.99**
15	64.04 ± 2.98	74.44 ± 4.25	58.62 ± 12.50	61.74 ± 6.22	67.42 ± 9.75	88.01 ± 5.95	93.12 ± 3.12	98.43 ± 1.61	**99.72 ± 0.36**
16	89.41 ± 8.33	99.47 ± 0.75	94.29 ± 4.03	99.68 ± 0.95	98.41 ± 2.56	**100.00 ± 0.00**	**100.00 ± 0.00**	**100.00 ± 0.00**	**100.00 ± 0.00**
OA	66.93 ± 1.50	70.62 ± 2.94	72.64 ± 2.41	72.31 ± 1.07	75.27 ± 1.11	83.62 ± 1.72	82.48 ± 2.22	90.91 ± 2.42	**94.66 ± 0.91**
AA	77.53 ± 1.46	81.75 ± 2.17	81.07 ± 1.18	82.15 ± 1.38	84.41 ± 1.14	91.19 ± 0.97	90.01 ± 2.56	95.20 ± 1.35	**97.39 ± 0.31**
Kappa	62.81 ± 1.66	66.83 ± 3.28	69.04 ± 2.64	68.76 ± 1.12	71.98 ± 1.26	81.44 ± 1.93	80.16 ± 2.48	89.62 ± 2.76	**93.90 ± 1.03**

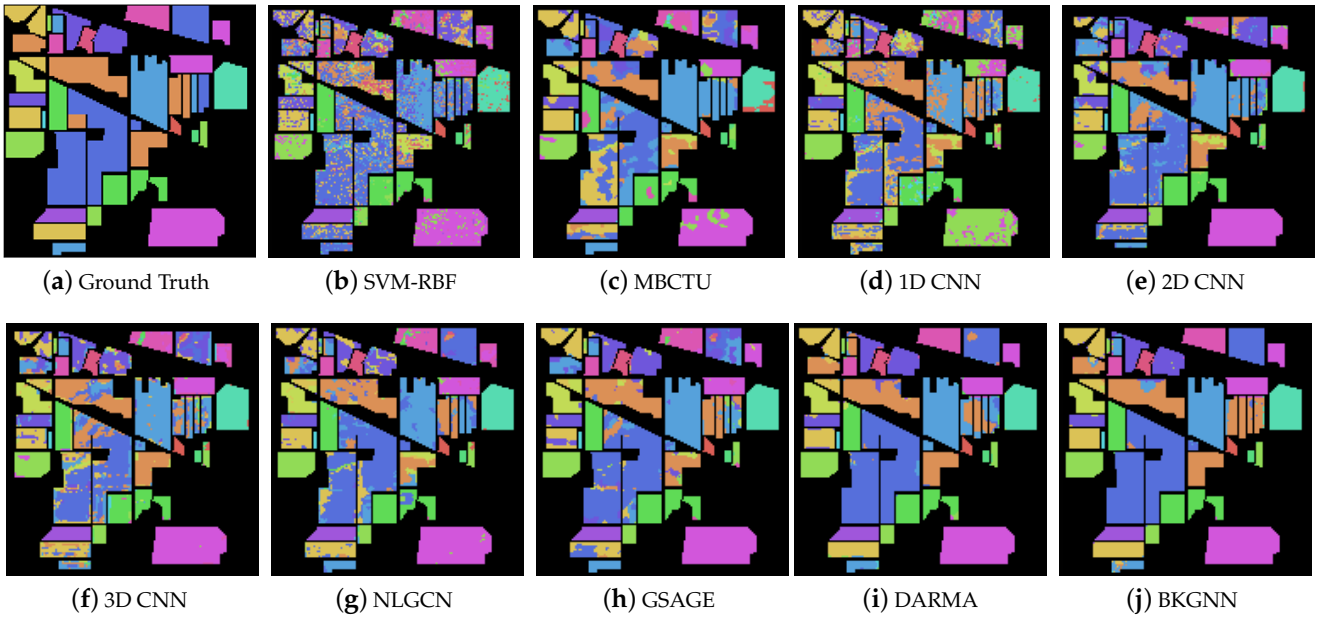
Table 3. Accuracy comparison for the PU data set. Bold numbers indicate the best performance.

Class No.	Conventional Classifiers		CNN-Based Methods			GNN-Based Methods			
	SVM-RBF	MBCTU	1D CNN	2D CNN	3D CNN	NLGCN	GSAGE	DARMA	BKGNN
1	65.14 ± 5.45	75.89 ± 4.64	69.58 ± 4.00	66.94 ± 8.50	75.56 ± 11.85	88.30 ± 2.60	90.54 ± 3.01	94.03 ± 1.35	**96.76 ± 0.72**
2	59.19 ± 4.45	55.52 ± 14.56	65.41 ± 13.70	61.68 ± 4.49	74.15 ± 11.41	75.17 ± 7.54	81.87 ± 6.95	90.75 ± 5.53	**98.62 ± 1.21**
3	27.69 ± 2.46	66.30 ± 7.35	68.27 ± 23.92	62.24 ± 15.61	81.08 ± 8.34	89.70 ± 1.82	88.09 ± 8.48	95.90 ± 5.09	**99.40 ± 0.77**
4	95.25 ± 2.14	92.23 ± 1.75	93.84 ± 4.32	91.80 ± 2.05	91.14 ± 3.96	94.18 ± 1.78	94.74 ± 2.60	88.49 ± 3.01	**96.86 ± 1.13**
5	99.18 ± 0.14	**100.00 ± 0.00**	99.41 ± 0.20	99.38 ± 0.41	98.82 ± 0.59	99.69 ± 0.22	**100.00 ± 0.00**	97.83 ± 0.84	99.68 ± 0.42
6	70.42 ± 9.49	70.01 ± 20.63	59.33 ± 17.98	82.68 ± 5.23	67.06 ± 18.27	80.53 ± 6.42	89.98 ± 5.62	94.94 ± 4.72	**99.82 ± 0.17**
7	90.10 ± 1.45	95.82 ± 1.53	91.74 ± 1.47	85.19 ± 7.88	90.48 ± 2.92	96.15 ± 0.62	96.95 ± 2.60	99.91 ± 0.06	**100.00 ± 0.00**
8	87.03 ± 2.81	77.99 ± 11.50	71.60 ± 17.12	72.50 ± 12.96	92.27 ± 4.77	92.83 ± 2.73	86.19 ± 9.48	95.35 ± 1.68	**98.96 ± 0.51**
9	99.92 ± 0.05	98.80 ± 0.27	99.95 ± 0.05	99.13 ± 0.51	98.52 ± 0.77	**99.97 ± 0.05**	99.96 ± 0.05	96.53 ± 1.93	97.71 ± 2.05
OA	67.93 ± 0.55	69.01 ± 4.20	70.65 ± 4.82	70.77 ± 2.50	78.43 ± 3.11	83.36 ± 3.22	87.17 ± 3.16	92.86 ± 2.62	**98.47 ± 0.58**
AA	77.11 ± 0.16	81.40 ± 1.17	79.90 ± 1.47	80.17 ± 1.74	85.45 ± 1.67	90.72 ± 1.01	92.03 ± 1.06	94.86 ± 1.37	**98.65 ± 0.27**
Kappa	60.27 ± 0.25	61.86 ± 3.97	63.18 ± 4.94	63.96 ± 2.74	72.59 ± 3.28	78.80 ± 3.83	83.58 ± 3.81	90.69 ± 3.31	**97.98 ± 0.76**

Table 4. Accuracy comparison for the KSC data set. Bold numbers indicate the best performance.

Class No.	Conventional Classifiers		CNN-Based Methods			GNN-Based Methods			
	SVM-RBF	MBCTU	1D CNN	2D CNN	3D CNN	NLGCN	GSAGE	DARMA	BKGNN
1	89.78 ± 1.89	97.84 ± 0.10	79.78 ± 17.29	79.49 ± 5.55	93.57 ± 6.16	97.13 ± 1.26	96.79 ± 2.00	99.64 ± 0.24	**99.97 ± 0.05**
2	84.66 ± 0.58	92.49 ± 2.26	83.90 ± 4.08	80.85 ± 11.14	74.65 ± 7.86	90.94 ± 2.46	91.92 ± 2.12	97.28 ± 3.15	**100.00 ± 0.00**
3	56.78 ± 22.21	94.87 ± 2.14	50.44 ± 34.24	69.03 ± 17.86	85.40 ± 12.67	94.65 ± 5.32	97.92 ± 5.66	97.26 ± 2.30	**99.82 ± 0.22**
4	26.42 ± 19.23	14.14 ± 4.63	32.93 ± 23.16	46.08 ± 14.75	22.52 ± 13.80	59.46 ± 9.54	54.23 ± 23.74	**99.01 ± 1.41**	98.65 ± 1.61
5	38.42 ± 2.00	44.43 ± 5.66	38.93 ± 13.56	66.26 ± 13.75	84.73 ± 5.67	85.65 ± 8.61	84.89 ± 18.63	87.79 ± 0.84	**96.34 ± 2.95**
6	41.37 ± 2.47	77.69 ± 1.29	38.89 ± 11.66	80.65 ± 15.11	79.90 ± 9.24	75.58 ± 8.85	87.04 ± 6.59	97.59 ± 4.11	**100.00 ± 0.00**
7	89.33 ± 2.17	95.47 ± 4.27	84.67 ± 16.44	95.33 ± 8.56	98.00 ± 1.54	95.20 ± 6.09	98.27 ± 3.16	**100.00 ± 0.00**	99.47 ± 1.07
8	44.80 ± 8.03	65.34 ± 4.14	83.44 ± 6.95	67.88 ± 5.05	69.58 ± 4.17	93.17 ± 2.93	97.33 ± 2.50	**100.00 ± 0.00**	**100.00 ± 0.00**
9	75.98 ± 6.67	83.43 ± 1.13	71.94 ± 16.71	84.02 ± 5.77	81.63 ± 8.74	96.88 ± 4.49	97.14 ± 2.66	**100.00 ± 0.00**	**100.00 ± 0.00**
10	65.95 ± 1.81	93.64 ± 2.67	78.66 ± 8.99	96.47 ± 2.61	94.92 ± 3.76	97.99 ± 1.61	96.39 ± 5.79	99.79 ± 0.11	**99.89 ± 0.13**
11	89.88 ± 1.15	**100.00 ± 0.00**	93.42 ± 1.39	99.54 ± 0.52	99.74 ± 1.65	98.92 ± 0.94	**100.00 ± 0.00**	**100.00 ± 0.00**	**100.00 ± 0.00**
12	83.72 ± 2.10	89.81 ± 0.84	80.13 ± 4.95	94.28 ± 4.87	82.47 ± 1.81	92.79 ± 3.21	95.03 ± 3.28	99.37 ± 1.27	**99.20 ± 1.50**
13	99.85 ± 0.14	**100.00 ± 0.00**	**100.00 ± 0.00**	99.99 ± 0.03	**100.00 ± 0.00**	99.79 ± 0.46	**100.00 ± 0.00**	**100.00 ± 0.00**	**100.00 ± 0.00**
OA	75.98 ± 0.31	86.58 ± 0.34	78.03 ± 4.33	84.18 ± 1.93	85.87 ± 1.64	93.70 ± 0.73	94.70 ± 1.31	99.14 ± 0.41	**99.73 ± 0.14**
AA	68.23 ± 0.15	80.70 ± 0.55	70.55 ± 4.21	79.86 ± 2.97	82.24 ± 2.80	90.63 ± 1.11	92.07 ± 1.60	98.29 ± 0.58	**99.49 ± 0.26**
Kappa	73.14 ± 0.34	84.99 ± 0.38	75.53 ± 4.74	82.38 ± 2.15	84.22 ± 1.94	92.95 ± 0.81	94.08 ± 1.47	99.04 ± 0.45	**99.70 ± 0.16**

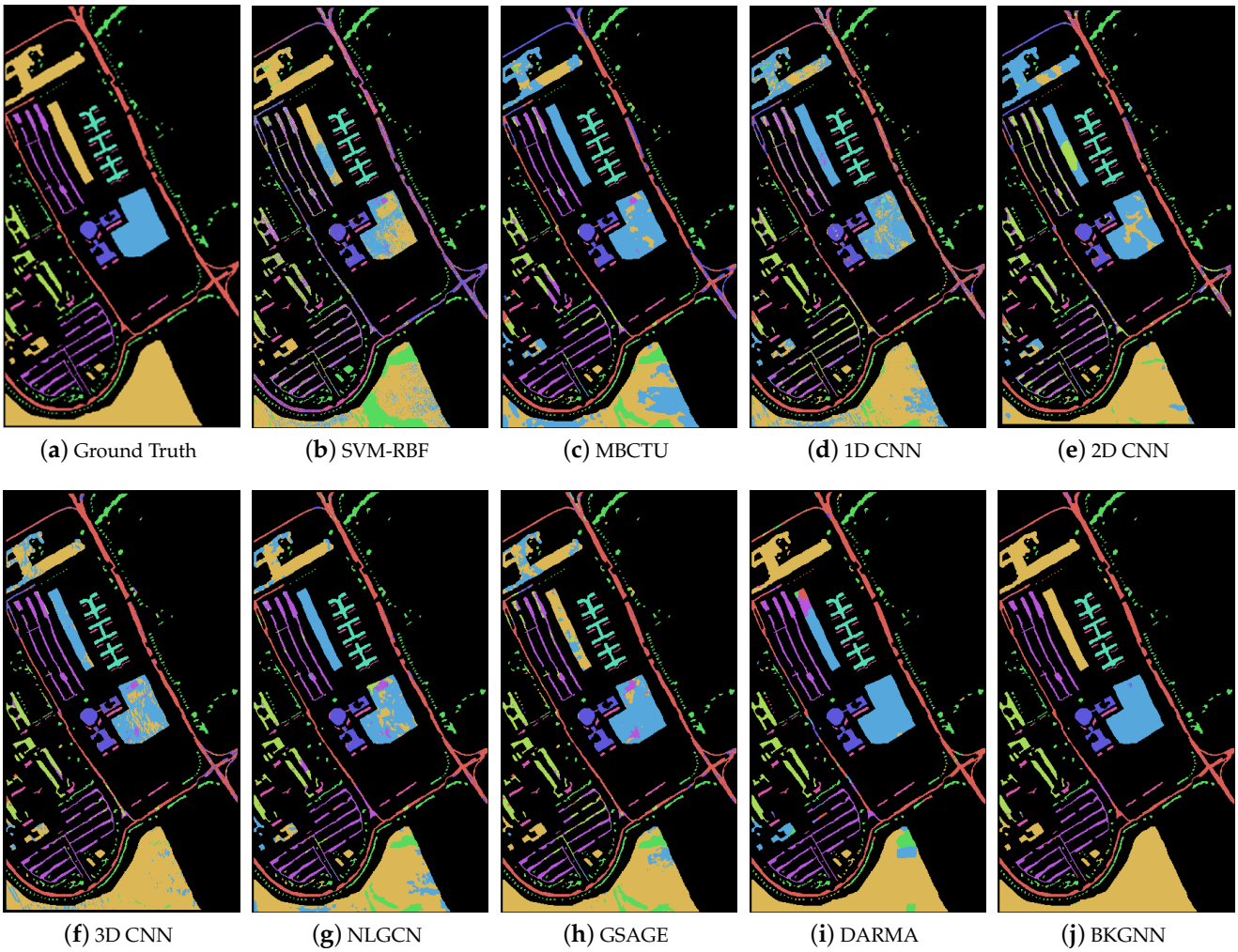
Moreover, the classification maps for the comparative methods are displayed in Figures 6–8. In general, SVM-RBF and 1D CNN suffered from serious salt and pepper noise in the classification maps, while 2D CNN and 3D CNN alleviated this problem by automatically extracting spatial features. Compared with traditional classifiers and CNN-based approaches, the GNN models were able to preserve more edge details, mainly due to their ability to learn the relationships between various land-cover classes and model their spatial topologies on graphs. Compared with other methods, the visual maps of the proposed BKGNN were significantly more similar to the ground truth. This further demonstrates that the proposed model can significantly improved the discriminative ability of features to satisfy the classification performance.



(**a**) Ground Truth (**b**) SVM-RBF (**c**) MBCTU (**d**) 1D CNN (**e**) 2D CNN

(**f**) 3D CNN (**g**) NLGCN (**h**) GSAGE (**i**) DARMA (**j**) BKGNN

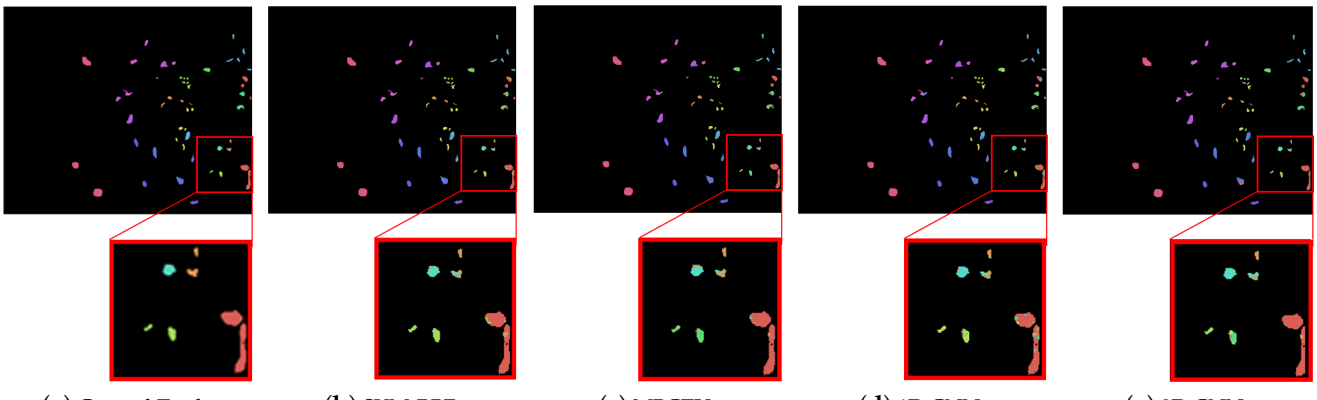
Figure 6. Ground truth and classification maps obtained by different methods on the Indian Pines data set.



(**a**) Ground Truth (**b**) SVM-RBF (**c**) MBCTU (**d**) 1D CNN (**e**) 2D CNN

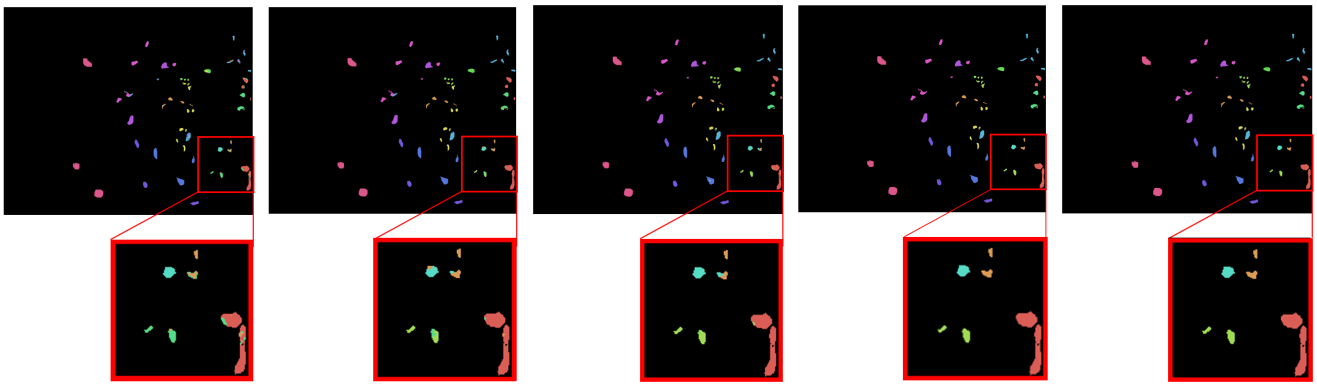
(**f**) 3D CNN (**g**) NLGCN (**h**) GSAGE (**i**) DARMA (**j**) BKGNN

Figure 7. Ground truth and classification maps obtained by different methods on the Pavia University data set.



(**a**) Ground Truth (**b**) SVM-RBF (**c**) MBCTU (**d**) 1D CNN (**e**) 2D CNN

Figure 8. *Cont.*



(**f**) 3D CNN (**g**) NLGCN (**h**) GSAGE (**i**) DARMA (**j**) BKGNN

Figure 8. Ground truth and classification maps obtained by different methods on the Kennedy Space Center data set.

4. Discussion

In this section, we analyze the influence of the number of superpixels on the effectiveness and efficiency of the proposed method, and further conduct hyper-parameter sensitivity experiments.

4.1. Analysis of the Number of Superpixels

In our proposed method, the number of superpixels N plays an important part in constructing the homophily degree matrix. We varied N from 200 to 5000, and the classification results and time costs are reported in Table 5. OM means "out of memory". It can be seen that the OAs first grew and gradually decreased as N increased. Note that, with small N (e.g., 200), the classification accuracy was greatly reduced. This is because the superpixels might incorporate pixels with many different labels when performing superpixel segmentation. Similarly, the performance decreased with a large N, proving the importance of selecting a suitable number of superpixel for our algorithm. The number of superpixels has a significant impact on the time consumption, and a large N may even lead to out-of-memory errors. Empirically, we chose $N = 500$ for the IP data set, and $N = 1000$ for the other two data sets.

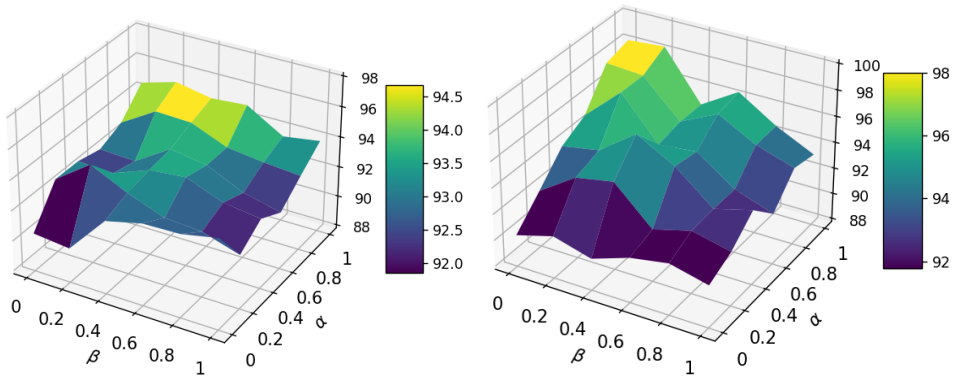
Table 5. Classification performance (OA) and time cost (s) with varying N on three data sets.

Datasets		200	500	1000	2000	3000	4000	5000
IP	OA	89.65	94.66	94.59	92.70	91.94	91.55	90.25
	time	18.7	21.2	24.8	70.4	76.71	339.1	342.8
PU	OA	90.25	97.66	98.47	95.78	97.69	96.10	94.68
	time	45.0	50.8	66.2	104.1	185.4	301.1	OM
KSC	OA	95.57	98.74	99.73	99.60	98.18	98.37	98.47
	time	69.9	74.7	89.5	125.9	216.5	285.5	OM

4.2. Analysis of Weights α and β

As the performance of our method highly depends on the quality of the homophily degree matrix (i.e., $\boldsymbol{H}$), we investigated the performance gains obtained by adjusting the two parameters α and β, which represent the weights estimated from the attribute space and topology, respectively. We show the classification performance change trend in Figure 9, obtained by varying α and β from 0 to 1. As can be seen from the figure, the performance was relatively poor when $\alpha = 0$ and $\beta = 0$, which reveals that it is necessary to estimate the homophily degree matrix by combining node attributes and network topology. Furthermore, our method performed best when $\alpha = 1$ and $\beta = 0.2$,

demonstrating that the attribute information is more important than topology information on these two data sets.

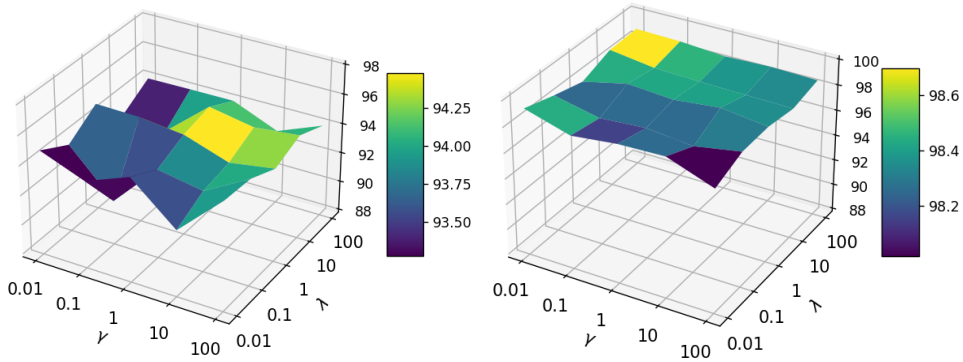


(**k**) IP (**l**) PU

Figure 9. Analysis results for varying weights α and β.

4.3. Analysis of Trade-Off Parameters λ and γ

We validate our approach's sensitivity to λ and γ which trade-off between MLP and LP loss. The value ranges of λ and γ are $\{0.01, 0.1, 1, 10, 100\}$. It can be seen from Figure 10a that the proposed method with parameters λ and γ in the range of 1 to 100 achieved suboptimal classification performance on the Indian Pines data set. It is interesting to observe from Figure 10b that the performance is fairly stable on the Pavia University data set. This phenomenon illustrates that the proposed BKGNN can achieve satisfactory results on a wide range of trade-off parameters, demonstrating the practicability of the algorithm.



(**a**) IP (**b**) PU

Figure 10. Analysis results of trade-off parameters λ and γ.

5. Conclusions

In this paper, by analyzing the homophily levels in HSI, we find that the heterophily information that exists in the constructed graph cannot be neglected. Therefore, we propose a novel bi-kernel GNN model, which learns two kernels to model homophily and heterophily, respectively, and the kernel is adaptively selected according to a learnable homophily degree matrix. In order to better model the homophily and heterophily in graph structure, the homophily degree matrix is calculated by exploiting the topological features and attribute features through LP and MLP, respectively. The estimation of the homophily degree matrix and the process of bi-kernel feature transformation are jointly trained with supervised loss, thus they can be enhanced by each other. The experimental results on three real-world data sets demonstrated the effectiveness of our method.

Author Contributions: Conceptualization, H.H. and Y.D.; methodology, H.H.; software, H.H. and Y.D.; validation, H.H., Y.D. and F.H.; formal analysis, F.Z.; writing—original draft preparation, H.H.; writing—review and editing, Y.D., F.H. and J.Z.; visualization, F.H.; supervision, M.Y.; funding acquisition, F.Z. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Data Availability Statement: Publicly available data sets were analyzed in this study, which can be found here: http://www.ehu.eus/ccwintco/index.php?title=Hyperspectral_Remote_Sensing_Scenes (accessed on 1 April 2022).

Acknowledgments: The authors would like to thank the authors of all the references used in the paper, the editors, and the anonymous reviewers for their detailed comments and suggestions.

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

References

1. Wang, R.; Nie, F.; Yu, W. Fast spectral clustering with anchor graph for large hyperspectral images. *IEEE Geosci. Remote Sens. Lett.* **2017**, *14*, 2003–2007. [CrossRef]
2. Chen, Y.; Lin, Z.; Zhao, X.; Wang, G.; Gu, Y. Deep learning-based classification of hyperspectral data. *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.* **2014**, *7*, 2094–2107. [CrossRef]
3. Bioucas-Dias, J.M.; Plaza, A.; Dobigeon, N.; Parente, M.; Du, Q.; Gader, P.; Chanussot, J. Hyperspectral unmixing overview: Geometrical, statistical, and sparse regression-based approaches. *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.* **2012**, *5*, 354–379. [CrossRef]
4. Liu, S.; Marinelli, D.; Bruzzone, L.; Bovolo, F. A review of change detection in multitemporal hyperspectral images: Current techniques, applications, and challenges. *IEEE Geosci. Remote Sens. Mag.* **2019**, *7*, 140–158. [CrossRef]
5. Nasrabadi, N.M. Hyperspectral target detection: An overview of current and future challenges. *IEEE Signal Process. Mag.* **2013**, *31*, 34–44. [CrossRef]
6. Li, W.; Du, Q. Collaborative representation for hyperspectral anomaly detection. *IEEE Trans. Geosci. Remote Sens.* **2014**, *53*, 1463–1474. [CrossRef]
7. Hu, H.; Yao, M.; He, F.; Zhang, F.; Zhao, J.; Yan, S. Nonnegative collaborative representation for hyperspectral anomaly detection. *Remote Sens. Lett.* **2022**, *13*, 352–361. [CrossRef]
8. Hu, H.; He, F.; Zhang, F.; Ding, Y.; Wu, X.; Zhao, J.; Yao, M. Unifying Label Propagation and Graph Sparsification for Hyperspectral Image Classification. *IEEE Geosci. Remote Sens. Lett.* **2022**, *19*, 1–5. [CrossRef]
9. Blanzieri, E.; Melgani, F. Nearest Neighbor Classification of Remote Sensing Images With the Maximal Margin Principle. *IEEE Trans. Geosci. Remote Sens.* **2008**, *46*, 1804–1811. [CrossRef]
10. Tarabalka, Y.; Fauvel, M.; Chanussot, J.; Benediktsson, J.A. SVM- and MRF-Based Method for Accurate Classification of Hyperspectral Images. *IEEE Geosci. Remote Sens. Lett.* **2010**, *7*, 736–740. [CrossRef]
11. Fauvel, M.; Benediktsson, J.A.; Chanussot, J.; Sveinsson, J.R. Spectral and Spatial Classification of Hyperspectral Data Using SVMs and Morphological Profiles. *IEEE Trans. Geosci. Remote Sens.* **2008**, *46*, 3804–3814. [CrossRef]
12. Sun, X.; Qu, Q.; Nasrabadi, N.M.; Tran, T.D. Structured Priors for Sparse-Representation-Based Hyperspectral Image Classification. *IEEE Geosci. Remote Sens. Lett.* **2014**, *11*, 1235–1239. [CrossRef]
13. Uddin, M.P.; Mamun, M.A.; Hossain, M.A. Effective feature extraction through segmentation-based folded-PCA for hyperspectral image classification. *Int. J. Remote Sens.* **2019**, *40*, 7190–7220. [CrossRef]
14. Bandos, T.V.; Bruzzone, L.; Camps-Valls, G. Classification of Hyperspectral Images With Regularized Linear Discriminant Analysis. *IEEE Trans. Geosci. Remote Sens.* **2009**, *47*, 862–873. [CrossRef]
15. Gu, Y.; Liu, T.; Jia, X.; Benediktsson, J.A.; Chanussot, J. Nonlinear multiple kernel learning with multiple-structure-element extended morphological profiles for hyperspectral image classification. *IEEE Trans. Geosci. Remote Sens.* **2016**, *54*, 3235–3247. [CrossRef]
16. Xia, J.; Dalla Mura, M.; Chanussot, J.; Du, P.; He, X. Random subspace ensembles for hyperspectral image classification with extended morphological attribute profiles. *IEEE Trans. Geosci. Remote Sens.* **2015**, *53*, 4768–4786. [CrossRef]
17. Jia, S.; Hu, J.; Xie, Y.; Shen, L.; Jia, X.; Li, Q. Gabor cube selection based multitask joint sparse representation for hyperspectral image classification. *IEEE Trans. Geosci. Remote Sens.* **2016**, *54*, 3174–3187. [CrossRef]
18. Ding, Y.; Zhao, X.; Zhang, Z.; Cai, W.; Yang, N.; Zhan, Y. Semi-Supervised Locality Preserving Dense Graph Neural Network With ARMA Filters and Context-Aware Learning for Hyperspectral Image Classification. *IEEE Trans. Geosci. Remote Sens.* **2021**. [CrossRef]
19. Yang, P.; Tong, L.; Qian, B.; Gao, Z.; Yu, J.; Xiao, C. Hyperspectral Image Classification With Spectral and Spatial Graph Using Inductive Representation Learning Network. *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.* **2021**, *14*, 791–800. [CrossRef]
20. Chen, Y.; Zhao, X.; Jia, X. Spectral–spatial classification of hyperspectral data based on deep belief network. *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.* **2015**, *8*, 2381–2392. [CrossRef]

21. Mou, L.; Ghamisi, P.; Zhu, X.X. Deep recurrent neural networks for hyperspectral image classification. *IEEE Trans. Geosci. Remote Sens.* **2017**, *55*, 3639–3655. [CrossRef]
22. Hong, D.; Gao, L.; Yokoya, N.; Yao, J.; Chanussot, J.; Du, Q.; Zhang, B. More Diverse Means Better: Multimodal Deep Learning Meets Remote-Sensing Imagery Classification. *IEEE Trans. Geosci. Remote Sens.* **2021**, *59*, 4340–4354. [CrossRef]
23. Hamida, A.B.; Benoit, A.; Lambert, P.; Amar, C.B. 3-D deep learning approach for remote sensing image classification. *IEEE Trans. Geosci. Remote Sens.* **2018**, *56*, 4420–4434. [CrossRef]
24. Roy, S.K.; Krishna, G.; Dubey, S.R.; Chaudhuri, B.B. HybridSN: Exploring 3-D–2-D CNN feature hierarchy for hyperspectral image classification. *IEEE Geosci. Remote Sens. Lett.* **2019**, *17*, 277–281. [CrossRef]
25. Zhu, K.; Chen, Y.; Ghamisi, P.; Jia, X.; Benediktsson, J.A. Deep convolutional capsule network for hyperspectral image spectral and spectral-spatial classification. *Remote Sens.* **2019**, *11*, 223. [CrossRef]
26. Zhang, X.; Liang, Y.; Li, C.; Huyan, N.; Jiao, L.; Zhou, H. Recursive autoencoders-based unsupervised feature learning for hyperspectral image classification. *IEEE Geosci. Remote Sens. Lett.* **2017**, *14*, 1928–1932. [CrossRef]
27. Hong, D.; Han, Z.; Yao, J.; Gao, L.; Zhang, B.; Plaza, A.; Chanussot, J. SpectralFormer: Rethinking hyperspectral image classification with transformers. *IEEE Trans. Geosci. Remote Sens.* **2021**, *60*, 1–15. [CrossRef]
28. Bai, J.; Ding, B.; Xiao, Z.; Jiao, L.; Chen, H.; Regan, A.C. Hyperspectral Image Classification Based on Deep Attention Graph Convolutional Network. *IEEE Trans. Geosci. Remote Sens.* **2021**. [CrossRef]
29. Mu, C.; Dong, Z.; Liu, Y. A Two-Branch Convolutional Neural Network Based on Multi-Spectral Entropy Rate Superpixel Segmentation for Hyperspectral Image Classification. *Remote Sens.* **2022**, *14*, 1569. [CrossRef]
30. Kipf, T.N.; Welling, M. Semi-supervised classification with graph convolutional networks. *arXiv* **2016**, arXiv:1609.02907.
31. Veličković, P.; Cucurull, G.; Casanova, A.; Romero, A.; Lio, P.; Bengio, Y. Graph attention networks. *arXiv* **2017**, arXiv:1710.10903.
32. Wang, Y.; Sun, Y.; Liu, Z.; Sarma, S.E.; Bronstein, M.M.; Solomon, J.M. Dynamic Graph CNN for Learning on Point Clouds. *ACM Trans. Graph.* **2019**, *38*, 1–12. [CrossRef]
33. Hamilton, W.; Ying, Z.; Leskovec, J. Inductive representation learning on large graphs. *Adv. Neural Inf. Process. Syst.* **2017**, *30*, 1025–1035.
34. Qin, A.; Shang, Z.; Tian, J.; Wang, Y.; Zhang, T.; Tang, Y.Y. Spectral Spatial Graph Convolutional Networks for Semisupervised Hyperspectral Image Classification. *IEEE Geosci. Remote Sens. Lett.* **2019**, *16*, 241–245. [CrossRef]
35. Dong, Y.; Liu, Q.; Du, B.; Zhang, L. Weighted feature fusion of convolutional neural network and graph attention network for hyperspectral image classification. *IEEE Trans. Image Process.* **2022**, *31*, 1559–1572. [CrossRef]
36. Hu, H.; Yao, M.; He, F.; Zhang, F. Graph neural network via edge convolution for hyperspectral image classification. *IEEE Geosci. Remote Sens. Lett.* **2021**, *19*, 1–5. [CrossRef]
37. Ding, Y.; Zhao, X.; Zhang, Z.; Cai, W.; Yang, N. Graph Sample and Aggregate-Attention Network for Hyperspectral Image Classification. *IEEE Geosci. Remote. Sens. Lett.* **2021**, *19*, 5504205. [CrossRef]
38. Ding, Y.; Zhao, X.; Zhang, Z.; Cai, W.; Yang, N. Multiscale Graph Sample and Aggregate Network With Context-Aware Learning for Hyperspectral Image Classification. *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.* **2021**, *14*, 4561–4572. [CrossRef]
39. Jia, S.; Deng, X.; Xu, M.; Zhou, J.; Jia, X. Superpixel-level weighted label propagation for hyperspectral image classification. *IEEE Trans. Geosci. Remote Sens.* **2020**, *58*, 5077–5091. [CrossRef]
40. Zhang, H.; Zou, J.; Zhang, L. EMS-GCN: An End-to-End Mixhop Superpixel-Based Graph Convolutional Network for Hyperspectral Image Classification. *IEEE Trans. Geosci. Remote Sens.* **2022**, *60*, 5526116. [CrossRef]
41. Bai, J.; Shi, W.; Xiao, Z.; Regan, A.C.; Ali, T.A.A.; Zhu, Y.; Zhang, R.; Jiao, L. Hyperspectral Image Classification Based on Superpixel Feature Subdivision and Adaptive Graph Structure. *IEEE Trans. Geosci. Remote Sens.* **2022**, *60*, 5524415. [CrossRef]
42. Ma, Y.; Liu, X.; Shah, N.; Tang, J. Is homophily a necessity for graph neural networks? *arXiv* **2021**, arXiv:2106.06134.
43. Wang, T.; Jin, D.; Wang, R.; He, D.; Huang, Y. Powerful graph convolutional networks with adaptive propagation mechanism for homophily and heterophily. In Proceedings of the AAAI Conference on Artificial Intelligence, Virtual, 22 February– 1 March 2022; Volume 36, pp. 4210–4218.
44. Zhu, J.; Yan, Y.; Zhao, L.; Heimann, M.; Akoglu, L.; Koutra, D. Beyond homophily in graph neural networks: Current limitations and effective designs. *Adv. Neural Inf. Process. Syst.* **2020**, *33*, 7793–7804.
45. Gilmer, J.; Schoenholz, S.S.; Riley, P.F.; Vinyals, O.; Dahl, G.E. Neural message passing for quantum chemistry. In Proceedings of the International Conference on Machine Learning, PMLR, Sydney, Australia, 6–11 August 2017; pp. 1263–1272.
46. Li, G.; Mueller, M.; Qian, G.; Delgadillo Perez, I.C.; Abualshour, A.; Thabet, A.K.; Ghanem, B. DeepGCNs: Making GCNs Go as Deep as CNNs. *IEEE Trans. Pattern Anal. Mach. Intell.* **2021**. [CrossRef] [PubMed]
47. Du, L.; Shi, X.; Fu, Q.; Ma, X.; Liu, H.; Han, S.; Zhang, D. GBK-GNN: Gated Bi-Kernel Graph Neural Networks for Modeling Both Homophily and Heterophily. In Proceedings of the ACM Web Conference 2022, Virtual Event, 25–29 April 2022; pp. 1550–1558.
48. Kuo, B.C.; Ho, H.H.; Li, C.H.; Hung, C.C.; Taur, J.S. A kernel-based feature selection method for SVM with RBF kernel for hyperspectral image classification. *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.* **2013**, *7*, 317–326. [CrossRef]
49. Djerriri, K.; Safia, A.; Adjoudj, R.; Karoui, M.S. Improving hyperspectral image classification by combining spectral and multiband compact texture features. In Proceedings of the IGARSS 2019—2019 IEEE International Geoscience and Remote Sensing Symposium, Yokohama, Japan, 28 July–2 August 2019; pp. 465–468.
50. Hu, W.; Huang, Y.; Wei, L.; Zhang, F.; Li, H. Deep convolutional neural networks for hyperspectral image classification. *J. Sens.* **2015**, *2015*, 258619. [CrossRef]

51. Liu, B.; Yu, X.; Zhang, P.; Tan, X.; Yu, A.; Xue, Z. A semi-supervised convolutional neural network for hyperspectral image classification. *Remote Sens. Lett.* **2017**, *8*, 839–848. [CrossRef]
52. Li, Y.; Zhang, H.; Shen, Q. Spectral–spatial classification of hyperspectral imagery with 3D convolutional neural network. *Remote Sens.* **2017**, *9*, 67. [CrossRef]
53. Mou, L.; Lu, X.; Li, X.; Zhu, X.X. Nonlocal Graph Convolutional Networks for Hyperspectral Image Classification. *IEEE Trans. Geosci. Remote Sens.* **2020**, *58*, 8246–8257. [CrossRef]