Hyperspectral Image Classification With Spectral and Spatial Graph Using Inductive Representation Learning Network

Pan Yang, Lei Tong, Bin Qian, *Member, IEEE*, Zheng Gao, Jing Yu, *Member, IEEE*, and Chuangbai Xiao, *Member, IEEE*

***Abstract*—Convolutional neural networks (CNN) have achieved excellent performance for the hyperspectral image (HSI) classification problem due to better extracting spectral and spatial information. However, CNN can only perform convolution calculations on Euclidean datasets. To solve this problem, recently, the graph convolutional neural network (GCN) is proposed, which can be applied to the semisupervised HSI classification problem. However, the GCN is a direct push learning method, which requires all nodes to participate in the training process to get the node embedding. This may bring great computational cost for the hyperspectral data with a large number of pixels. Therefore, in this article, we propose an inductive learning method to solve the problem. It constructs the graph by sampling and aggregating (GraphSAGE) feature from a node's local neighborhood. This could greatly reduce the space complexity. Moreover, to enhance the classification performance, we also construct the graph using spectral and spatial information (spectra–spatial GraphSAGE). Experiments on several hyperspectral image datasets show that the proposed method can achieve better classification performance compared with state-of-the-art HSI classification methods.**

***Index Terms*—GraphSAGE, hyperspectral image classification, inductive learning method, spectral and spatial.**

I. Introduction

IN RECENT years, various applications of hyperspectral images in earth observation have aroused great interest. Hyperspectral remote sensors capture images in different electromagnetic wave range and combine both spectral and spatial information of ground objects. A hyperspectral image (HSI) normally contains tens or hundreds of bands, which can accurately identify and classify surface materials for feature extraction and image classification. Therefore, HSI has been successfully applied to agriculture [1], [2], mining [3], military [4], [5], and other earth observation fields.

In the past decades, numerous machine learning methods have been used for the hyperspectral classification problem, such as distance classifier [6], K-nearest neighbor classifier [7], support vector machine (SVM) [8], [9], Bayesian classifier [10], Gaussian maximum likelihood [11], sparse representation [12], [13], independent component discriminant (ICA) [14], kernel-based method [15], [16], and logistic regression [17], [18]. The drawback of machine learning-based methods is that they have to manually extract features [17], [19], [20], which may involve experts experience and parameter setting and affect the classification results.

Deep learning provides an ideal solution for feature extraction [21]–[26] for hyperspectral classification. Hu *et al.* [27] used convolutional neural networks (CNN) to extract spectral features for hyperspectral image classification. Mei *et al.* [28] also discussed a similar CNN strategy with spectral–spatial features. Lee and Kwon [29] proposed a scene depth CNN (CD-CNN), which optimizes the upper scene interaction by jointly using the local spectral–spatial relationship of adjacent pixel vectors in the square window. Li *et al.* [22] used joint representations based on different regions while simultaneously using spectral information, and semantic situational awareness information in each pixel. Hamida *et al.* [30] also introduced the 3-D convolution operation into hyperspectral classification. However, all these supervised deep learning methods usually need large labeled dataset to train the network.

Semisupervised methods are used to solve the large unlabeled data problem [31]. Yang *et al.* [32] proposed a semisupervised hyperspectral classification method using spectral–spatial Laplacian SVM. Tan *et al.* [33] presented a semisupervised method by the spatial neighborhood information and classifier combination. Andekah *et al.* [34] used the spectral–spatial features and superpixel-based sparse codes for classification. Recently, deep learning methods have also been used for semisupervised classification. Kipf and Welling [35] proposed a fast-approximate local convolution, which directly acts on graph signals, called graph convolution network (GCN). CN can be operated on a graph. It also can aggregate and transform features from the neighbors of each node. The convolution of GCN is

Manuscript received July 31, 2020; revised October 28, 2020; accepted November 26, 2020. Date of publication December 8, 2020; date of current version January 6, 2021. This work was supported in part by the National Natural Science Foundation of China under Grant 61 701 009 and in part by the Beijing Municipal Education Commission Science and Technology Program under Grant KM202010005016. *(Corresponding author: Lei Tong.)*

Pan Yang and Lei Tong are with the Faculty of Information Technology, Beijing University of Technology, Beijing 100124, China, and also with Engineering Research Center of Intelligent Perception and Autonomous Control, Ministry of Education, Beijing 100124, China (e-mail: yangpan@emails.bjut.edu.cn; lei_tong@bjut.edu.cn).

Bin Qian and Jing Yu are with the Traffic Management Research Institute of the Ministry of Public Security, Wuxi 214151, China (e-mail: qianbin_nust@126.com; jing.yu@bjut.edu.cn).

Zheng Gao and Chuangbai Xiao are with the Faculty of Information Technology, Beijing University of Technology, Beijing 100124, China (e-mail: gaozheng@emails.bjut.edu.cn; cbxiao@bjut.edu.cn).

Digital Object Identifier 10.1109/JSTARS.2020.3042959

controlled by the neighborhood structure of a graph. Therefore, GCN can be applied to the non-Euclidean data using the predefined graph. Wan *et al.* [36] proposed a hyperspectral image classification using multiscale dynamic GCN. It uses multiple input graphs with different neighborhood scales to enhance the performance. Qin *et al.* [37] also presented a GCN-based hyperspectral classification method. It combines the intrinsic information of labeled and unlabeled samples and uses spectral and spatial information to achieve better results. Recently, Sellars *et al.* [38] also proposed a graph-based learning method for hyperspectral image classification using superpixels. However, although the GCN can obtain the embedding of the vertices in the graph, it is a direct push learning method, which requires all nodes to participate in training to obtain node embedding. Moreover, each node learned is the only certain embedding for hyperspectral images. For the hyperspectral data, with tens of thousands of pixels, if the nodes are directly embedded, it requires large memory to compute the graph, which is difficult to apply.

Inspired by the inductive method which constructs the graph by sampling and aggregating (GraphSAGE) [39], a spectral–spatial GraphSAGE (S^2GraphSAGE) hyperspectral image classification algorithm is proposed. First, we compute the distance between the nodes and the adjacency relation of the nodes according to the spectral–spatial information of the hyperspectral image. Then, inductive learning method is adopted to construct the graph. In this way, we could fix the space and time complexity. Therefore, our proposed method can be easily applied to HSI classification systems. The main contributions of our article are as follows. 1) We introduce GraphSage into hyperspectral image classification problem. It means there is no need to establish the adjacency matrix of hyperspectral image pixels in advance. Moreover, for the proposed method the graph can be constructed during the training process, which greatly reduces the cost of the memory. 2) We use the spectral–spatial distance information to determine the adjacency relationship of the pixels in the hyperspectral image, which can enhance the performance.

The rest of this article is organized as follows. Section II introduces the related work, and Section III describes the proposed method S^2graphSage. Section IV verifies S^2graphSage by comparing it with other state-of-the-art HSI classification methods. Finally, Section V concludes this article.

II. Related Work

In this section, we review the representative work on GCN, because it is the fundamental work for our proposed method. Although the CNN has achieved great performance for classification problem, it is a challenge for the graph-structured non-Euclidean data. Therefore, the GCN [35] is proposed to solve this problem. It has the ability to extract features from graph data, which can be used to perform node classification, graph classification, and link prediction on the graph data.

For the graph data, there are N nodes, and each node has its characteristics. We set the characteristics of these nodes to form an $N \times D$ dimension matrix $\mathbf{X}$, and then the relationship between each node will also form an $N \times N$ dimension matrix $\mathbf{A}$, also known as the adjacency matrix. $\mathbf{X}$ and $\mathbf{A}$ are the inputs to our model. The GCN generally includes a two-layer network, and the propagation formula

$$\mathbf{H}^{(l+1)} = \sigma(\widetilde{\mathbf{D}}^{-\frac{1}{2}}\widetilde{\mathbf{A}}\widetilde{\mathbf{D}}^{-\frac{1}{2}}\mathbf{H}^{(l)}\mathbf{W}^{(l)}). \tag{1}$$

Among them, $\mathbf{A} = \widetilde{\mathbf{A}} + \mathbf{I}$ is the adjacency matrix of the undirected graph *G* plus self-connection (that is, each node and itself plus an edge), $\mathbf{I}$ is the identity matrix, and $\widetilde{\mathbf{D}}$ is the degree matrix; $\mathbf{H}$ is the characteristic of each layer. For the input layer, $\mathbf{H}$ is the original data. For hyperspectral images, $\mathbf{H}$ here is each pixel.

Due to high dimension of matrix X, the Huge phenomenon often occurs in the hyperspectral classification problem. Therefore, the spatial information is always used to construct the adjacency matrix A to enhance the performance [40].

III. Methodology

In this section, we introduce our proposed spectral–spatial GraphSAGE (S^2GraphSAGE) algorithm in detail. Fig. 1 shows the framework of our proposed algorithm. It mainly contains two parts: Spectral–spatial graph construction and S^2GraphSAGE. When we construct the graph, we combine the distance from spectral and spatial domains and use K-nearest neighbor (KNN) to find the nearest neighbors. Then with the graph, S^2GraphSAGE is applied for classification.

A. Spectral–Spatial Graph Construction

For the graph-based method, the first step is to construct the adjacency relationship between pixels. Instead of the adjacency matrix, in our proposed method, we should construct a list of adjacency relationships, which is different from the GCN. For the remote sensing hyperspectral images, the spatial homogeneity should be their character, which means that two adjacent pixels in the spatial domain have a high probability of belonging to the same class. Moreover, spectral domain also can provide information for classification, which means the more similar the spectral signature is the higher probability of belonging to the same class. Therefore, to enhance the performance, we determine the adjacency relationship using spectral–spatial Euclidean distance as following:

$$dist_{(i,j)} = \omega\|\mathbf{x}_i - \mathbf{x}_j\|_2 + (1-\omega)\|\mathbf{p}_i - \mathbf{p}_j\|_2 \tag{2}$$

where, the $\mathbf{x}$ and $\mathbf{p}$ represent the spectral signatures and spatial coordinates of i and j pixels, respectively. ω is a parameter, which controls the weights of the spectral and spatial domain. According to (2), k nearest neighbors of node v_i (excluding itself) are selected to form a neighbor set $(N)(i) = \{j; (j,i) \in E\}$.

B. S^2GraphSAGE

For the proposed S^2GraphSAGE, instead of learning the embedding of all nodes in a graph, it tries to generate an embedded map for each node. Instead of training a separate embedding

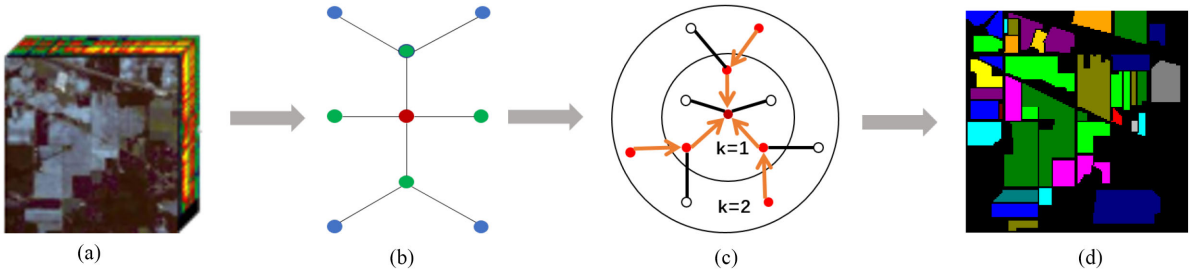

Fig. 1. Framework of S^{2}GraphSAGE. (a) Hyperspectral image. (b) Spectral–spatial graph construction. (c) GraphSAGE. (d) Classification result.

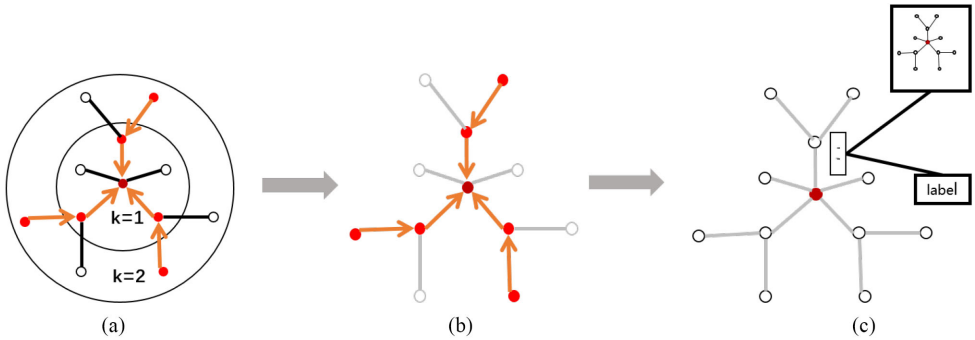

Fig. 2. GraphSAGE embedding. (a) Sample neighborhood. (b) Aggregate the information of neighbor nodes. (c) Predict the information needed in the graph.

vector for each node, our proposed method trains a set of aggregation functions that learn how to aggregate feature information from the local neighbors of a node. Each aggregation function aggregates information from a different search depth of a node. During the testing, the trained network is used to generate embedding for the vertices that have never been learned through the learned aggregation function.

GraphSAGE consists of two processes: Graph sampling and aggregate. Fig. 2 shows the process, taking the red target node for an example, which can be divided into three steps.

Step 1: Sample the neighbors of each node in the graph. For each node, for the efficiency, we set a fixed number of neighbors. Fig. 2 shows the step. k represents the search depth from the target node, $k = 1$ is the first-order neighbor node of the target node, and $k = 2$ is the two-order neighbor node of the target node.

Step 2: Aggregate the information of neighbor nodes according to the aggregation function.

Step 3: Get the vector representation of each node in the graph for the following tasks.

1) GraphSAGE Embedding: In this section, GraphSAGE's forward propagation algorithm (Algorithm 1) is as follows. Forward propagation describes how to use the aggregation function to aggregate the neighbor information of the node to generate the node embedding.

Algorithm 1 describes the process of generating embedding on the entire graph, where $\mathcal{G} = (\mathcal{V}; \mathcal{E})$ represents a graph. K is the number of layers of the network, and also represents the number of orders of adjacent nodes that each node can aggregate,

Algorithm 1: GraphSAGE Embedding Generation Algorithms.

Input: Graph $\mathcal{G}(\mathcal{V}; \mathcal{E})$;input features $\{\mathbf{x}_v, \forall_v \in \mathcal{V}\}$;depth K;weight matrices $\mathbf{W}^k, \forall_k \in \{1, ..., K\}$;non-linearity σ; differentiable aggregator functions $\text{AGGREGATE}_K, \forall_k \in \{1, ..., K\}$; neighborhood function $\mathcal{N} : v \rightarrow 2^{\mathcal{V}}$

Output: Vector representations $\mathbf{z}_v$ for all $v \in \mathcal{V}$

```
initialization:h_v^0 ← x_v, ∀v ∈ V;
for k = 1...K do
    for v ∈ V do
        h_{N(v)}^k ← AGGREGATE_k({h_u^{k-1}, ∀u ∈ N(v)});
        h_v^k ← σ(W^k · CONCAT(h_v^{k-1}, h_{N(v)}^k))
    end
    h_v^k ← h_v^k / ||h_v^k||_2, ∀v ∈ V
    z_v ← h_v^k, ∀v ∈ V
end
```

because each additional one layer, which can aggregate the information of the neighbor nodes of a further layer. $\{\mathbf{x}_v, \forall_v \in \mathcal{V}\}$ represents the feature vector of node v and serves as an input. $\{\mathbf{h}_u^{k-1}, \forall u \in \mathcal{N}(v)\}$ represents the embedding of the neighbor node u of node v in $k-1$ layer. $\mathbf{h}_{\mathcal{N}(v)}^k$ represents the feature representation of all neighbor nodes of node v in the k layer. $\{\mathbf{h}_v^k, \forall_v \in \mathcal{V}\}$ represents the feature representation of node v in

the k layer. $\mathcal{N}(v)$ is defined as uniformly taken from a fixed size in the set $\{u \in v : (u, \mathcal{V}) \in \mathcal{E}\}$, that is, the neighbors of each node in GraphSAGE are sampled from the network of the previous layer, not all neighbors participate, and the size of the sampled neighbors is fixed. At each iteration, nodes aggregate information from their local neighbor nodes, and as the process iterates, nodes will obtain information from further and further away.

2) Aggregate Function Selection: Then, we show the aggregation strategy of the algorithm. For our proposed method, we use two kinds of aggregation methods, one is the MEAN aggregator, the other is the GCN aggregator.

Mean aggregator.

For the target node v in the k layer, the average value of each dimension of all adjacent nodes u of the node v in the $k-1$ layer is first taken, then the average value of the target node v and its neighbor node is spliced, then the weight parameter $\sigma(\mathbf{W}^k)$ is multiplied, and finally the nonlinear transformation σ is performed. It aggregates the information of the node by following equations:

$$\mathbf{h}_{\mathcal{N}(v)}^k \leftarrow Mean(\{\mathbf{h}_u^{k-1}, \forall u \in \mathcal{N}(v)\}) \tag{3}$$

$$\mathbf{h}_v^k \leftarrow \sigma(\mathbf{W}^k \cdot CONCAT(\mathbf{h}_v^{k-1}, \mathbf{h}_{\mathcal{N}(v)}^k)). \tag{4}$$

In (3) and (4), $\{\mathbf{h}_u^{k-1}, \forall u \in \mathcal{N}(v)\}$ represents the embedding of the neighbor node u of node v in the $k-1$ layer; $\mathbf{h}_{\mathcal{N}(v)}^k$ represents the feature representation of all neighbor nodes of node v in the k layer; $\mathbf{h}_v^k$ represents the feature representation of node v in the k layer.

GCN aggregator.

GCN aggregation splices the vector representation of the target node v in the $k-1$ layer and the vector representation of adjacent node u in the $k-1$ layer. Then, the average value of each dimension of the vector is calculated and multiplied by the weight parameters, and the nonlinear transformation σ is performed on the obtained results to generate the vector representation of target node v in the $k-1$ layer

$$\mathbf{h}_v^k \leftarrow \sigma(\mathbf{W} \cdot MEAN(\{\mathbf{h}_u^{k-1}\} \cup \{\mathbf{h}_u^{k-1}, \forall u \in \mathcal{N}(v)\})). \tag{5}$$

Original line 4 and 5 in algorithm 1 are as follows:

$$\mathbf{h}_{\mathcal{N}(v)}^k \leftarrow AGGREGATE_k(\{\mathbf{h}_u^{k-1}, \forall u \in \mathcal{N}(v)\}) \tag{6}$$

$$\mathbf{h}_v^k \leftarrow \sigma(\mathbf{W}^k \cdot CONCAT(\mathbf{h}_v^{k-1}, \mathbf{h}_{\mathcal{N}(v)}^k)). \tag{7}$$

In our proposed method, we use the GCN aggregator. We replace the lines 4 and 5 in algorithm 1 using (5) for GCN inductive deformation. The GCN aggregation is approximate equivalent to transductive GCN framework. The difference between this convolutional aggregator and the other aggregators is that it does not have the CONCAT operation in line 5 of the algorithm 1. The convolutional aggregator does not concatenate the representation of the previous layer of vertices $\mathbf{h}_u^{k-1}$ with the neighbor vector of the aggregation $\mathbf{h}_{\mathcal{N}(v)}^k$.

3) Definition of Neighbors: For the construction of the graph, the definition of the neighbors could affect the performance. For our proposed method, we set a fixed value, that is the same number of neighbor nodes is selected for each node. Each time a neighbor node is selected, a fixed number of neighbor nodes are sampled uniformly from the immediate neighbor nodes (first-order neighbor nodes). Although only first-order neighbor nodes information is aggregated in each aggregation, the first-order neighbor nodes of the target node also aggregates its neighbor nodes (second-order neighbor nodes) information. Therefore, as the iteration progresses, the nodes gradually extract more information, as shown in Fig. 3.

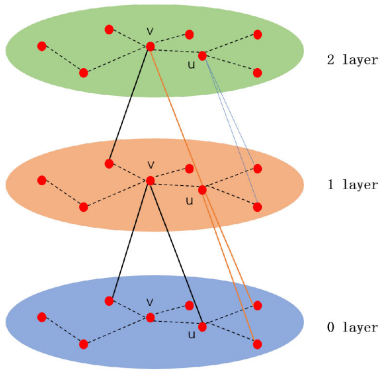

Fig. 3. Definition of neighbor.

To show the process of the construction graph, we take two neighbor nodes as an example. In Fig. 3 (for the sake of brevity of the figure, it is assumed that only two neighbor nodes are randomly aggregated), it can be seen that the information of the first-order neighbor nodes is indeed aggregated between the layers. In the "layer 1," node v aggregates the information of the two neighbor nodes of "layer 0," and the neighbor node u of v also aggregates the information of the two neighbor nodes of "layer 0". At "layer 2," you can see that node v has expanded to the second-order neighbor node of "layer 0" through node u of "layer 1." Therefore, during aggregation, k times can be extended to k-order neighbors. In the practice of GraphSAGE, when $k = 2$, it can get excellent performance. The influence of the number of neighbor nodes, we will discuss it in the experiments.

4) Learning the Parameters of S^{2}GraphSAGE: In our experiment, we use a supervised learning method for parameter learning. So the cross-entropy loss function is our objective function

$$\ell = -\sum_{s \in \mathbf{Y}_{\text{labeled}}} \sum_{c=1}^{C} \mathbf{Y}_{sc} ln \mathbf{Z}_{sc} \tag{8}$$

where $\mathbf{Y}_{sc}$ is the label of the training data and C is the number of object classes. The embedding $\mathbf{Z}_{sc}$ of node s is obtained through forward propagation, and then the gradient descent method (the Adam optimizer) is used to perform back propagation to optimize the parameters $\mathbf{W}^k$ and the parameters in the aggregation function.

IV. Experimental Results

In this section, we conducted experiments to verify the effectiveness of the proposed S^{2}GraphSage method. Three metrics

TABLE I
TRAINING, VALIDATION, AND TESTING NUMBERS IN THE INDIAN PINES DATASET

#	Class	Training	Val	Test
1	Alfalfa	30	5	11
2	Corn-notill	30	15	1383
3	Corn-mintill	30	15	785
4	Corn	30	15	192
5	Grass-pasture	30	15	438
6	Grass-trees	30	15	685
7	Grass-pasture-mowed	10	5	13
8	Hay-windrowed	30	15	433
9	Oats	10	5	5
10	Soybean-notill	30	15	927
11	Soybean-mintill	30	15	2410
12	Soybean-clean	30	15	548
13	Wheat	30	15	160
14	Woods	30	15	1220
15	Buildings-Grass-Trees-Drives	30	15	341
16	Stone-Steel-Towers	30	15	48
-	Total	440	210	9599

TABLE II
TRAINING, VALIDATION, AND TESTING NUMBERS IN THE PAVIA UNIVERSITY DATASET

#	Class	Training	Val	Test
1	Asphalt	30	15	6586
2	Meadows	30	15	18604
3	Gravel	30	15	2054
4	Trees	30	15	3019
5	Painted metal sheets	30	15	1300
6	Bare Soil	30	15	4984
7	Bitumen	30	15	1285
8	Self-Blocking Bricks	30	15	3637
9	Shadows	30	15	902
-	Total	270	135	42371

are used to measure the accuracy of the results, including overall accuracy (OA), average accuracy (AA), and kappa coefficient (Kappa).

A. Dataset

The performance of the proposed S^2GraphSAGE is evaluated on three datasets, i.e., the Indian Pines, the Pavia University, and the Kennedy Space Center.

1) Indian Pines: The Indian Pines dataset was collected by airborne visible/infrared imaging spectrometer sensor in 1992, which records north-western India. It consists of 145 × 145 pixels with a spatial resolution of 20 × 20 m and has 220 spectral channels covering the range from 0.4 to 2.5 μm. As a usual step, 20 water absorption and noisy bands are removed, and 200 bands are reserved. The original ground truth includes 16 land-cover classes. The amounts of labeled and unlabeled pixels of various classes are listed in Table I.

2) Pavia University: The Pavia University dataset captured the Pavia University in Italy with the ROSIS sensor in 2001. It consists of 610 × 340 pixels with a spatial resolution of 1.3 × 1.3 m and has 103 spectral channels in the wavelength range from 0.43 to 0.86 m after removing noisy bands. This dataset includes 9 land-cover classes. Table II lists the amounts of labeled and unlabeled pixels of each class.

TABLE III
TRAINING, VALIDATION, AND TESTING NUMBERS IN THE KENNEDY SPACE CENTER DATASET

#	Class	Training	Val	Test
1	Scrub	30	15	713
2	Willow swamp	30	15	205
3	Cabbage palm hammock	30	15	217
4	Cabbage palm/oak hammock	30	15	213
5	Slash pine	30	15	131
6	Oak/broadleaf hammock	30	15	192
7	Hardwoodd swamp	30	15	81
8	Graminoid marsh	30	15	378
9	Spartina marsh	30	15	454
10	Cattail marsh	30	15	350
11	Salt marsh	30	15	363
12	Mud flats	30	15	439
13	Water	30	15	821
-	Total	390	195	4626

3) Kennedy Space Center: The Kennedy Space Center dataset was taken by AVIRIS sensor over Florida with a spectral coverage ranging from 0.4 to 2.5 μm. This dataset contains 224 bands and 614 × 512 pixels with a spatial resolution of 18 m. After removing water absorption and noisy bands, the remaining 176 bands of the image have been preserved. The Kennedy Space Center dataset includes 13 land-cover classes. The numbers of labeled and unlabeled pixels of different classes are listed in Table III.

B. Experimental Setting

In our experiments, the proposed algorithm is implemented via Pytorch with Adam optimizer. For all the three datasets, 30 labeled pixels are randomly selected in each class for training, while 15 labeled examples are chosen for validating. Especially, for the Indian Pines dataset, only 15 labeled examples are chosen if the corresponding class has less than 30 examples and 5 or 15 labeled examples are chosen in each class for validating. Meanwhile, all the unlabeled examples are used as the test set to evaluate the classification performance. Besides, the learning rate and the number of training epochs are set to 0.001 and 5000, respectively. And the learning rate drops by 10% every 200 rounds.

C. Impact of Spectral–Spatial Euclidean Distance

For our proposed method, the parameter of ω controls the weight between the spectral and spatial domain, which may affect the results. To evaluate the ω, we change the ω from 0.5 to 0.1. The results are shown in Fig. 4. From the results, we can see that when ω is 0.4, we can get the best performance. Therefore, in all our experiments, we set $\omega = 0.4$.

D. Impact of the Number of Neighbor Nodes

When we construct the graph, the number of neighbor nodes could also affect the results. Therefore, we change the number of first-order neighbor nodes from 5 to 25 with an interval

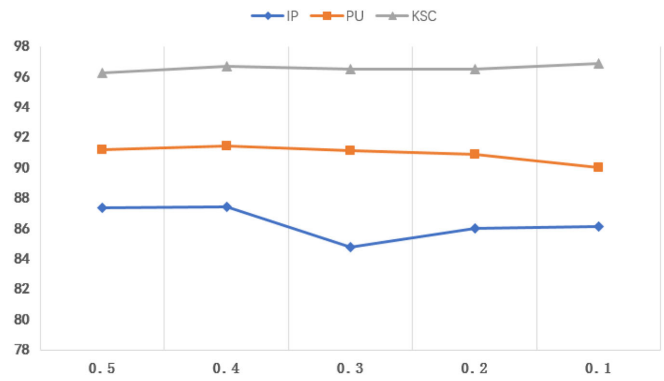

Fig. 4. OA result of different spectral–spatial Euclidean distance.

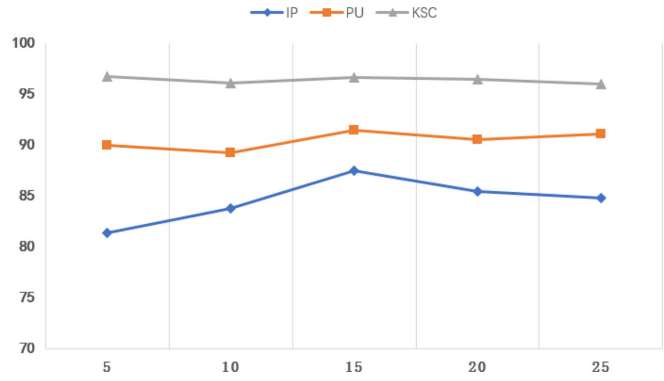

Fig. 5. OA result of different number of neighbors.

of 5. Second-order neighbor nodes have less impact on the classification effect than first-order neighbor nodes. Under the condition that the minimum value of first-order neighbor nodes is 5, set the number of second-order neighbor nodes to a fixed value of 5. The results are shown in Fig. 5. From the results, we can see when the number of neighbor nodes is small, the graph may neglect the structure information, which could lead to poor performance. On the other hand, when the number of neighbor nodes is large, the graph may contain the structure information different from the target node and get low OA. Therefore we set the number of first-order neighbor nodes as 15 and the number of second-order neighbor nodes as 5 for all the experiments.

E. Impact of the Aggregator

For our proposed method, we use two aggregation methods, i.e., mean aggregator and GCN aggregator. To evaluate the aggregation methods, as mentioned above, we set the number of first-order neighbor nodes as 15, the number of second-order neighbor nodes as 5, and the values of ω as 0.4. The results on three datasets are shown in Table IV. As can be seen from Table IV, the GCN aggregator can get better results. Therefore, we use GCN aggregator for all the experiments.

TABLE IV
OA, AA, AND KAPPA COEFFICIENT ACHIEVED BY DIFFERENT AGGREGATOR

Name	Indian Pines		Pavia University		Kennedy Space Center	
	MEAN	GCN	MEAN	GCN	MEAN	GCN
OA	81.41	**87.43**	89.75	**91.41**	95.61	**96.67**
AA	90.29	**93.40**	90.21	**91.53**	93.98	**94.51**
Kappa	78.95	**85.66**	86.09	**88.61**	95.08	**96.27**

Bold entities are the best results.

F. Classification Results

To illustrate the effectiveness of our proposed method, S^2 GraphSAGE is compared with several different hyperspectral image classification methods: Multilayer Perception (MLP), SVM based on RBF kernel [41], 3D-CNN [42], DR-CNN [22], multiscale-CNN [43], GCN [35], S^2 GCN [37], spectral-GraphSAGE and spatial-GraphSAGE. For the spectral-graphSAGE method, we only use spectral distance to construct the graph, while for the spatial-GraphSAGE method we use spatial distance. All the classification methods were tuned to the best settings. The training, validation, and testing data are divided the same as Tables I–III.

1) Results on the Indian Pines Dataset: The quantitative results of the Indian Pines dataset are shown in Table V. From the table, we can see deep learning methods perform relatively better than traditional methods due to better extracting spectral and spatial information. The results of the spectral-graphSAGE method are not as good as the spectral-graphSAGE. This indicates that the spatial information is more important than the spectral information. For our proposed method, we combine both spectral and spatial information. Therefore, it can outperform other methods. Fig. 6 exhibits a visual comparison of the classification results generated by different methods on the Indian Pines dataset.

2) Results on the Pavia University Dataset: Table VI presents the results of different methods on the Pavia University dataset. Compared with the Indian Pines dataset, the distribution of the Pavia University dataset is relatively uniform. Moreover, the number of different types of pixels is similar to each other. Therefore, the results have been slightly improved. According to [36], the GCN method needs large memory, it may not be suitable for the large dataset such as Pavia University. Therefore, we only display other HSI classification methods. From Table VI, we can see that our proposed method also gets the best performance. Fig. 7 exhibits a visual comparison of the classification results generated by different methods on the Pavia University dataset.

3) Results on the Kennedy Space Center Dataset: At last, we conduct the experiments on Kennedy Space Center dataset. The results are shown in Table VII. For the dataset, the spatial resolution is higher than the other two datasets. Therefore, the results are better than the other two datasets. The performance of the S^2GraphSAGE method we proposed is still the best. Fig. 8 exhibits a visual comparison of the classification results generated by different methods on the Kennedy Space Center dataset.

TABLE V
CLASSIFICATION RESULTS OF DIFFERENT METHODS FOR THE INDIAN PINES DATASET

Class	MLP	SVM-RBF	3D-CNN	DR-CNN	Multi-Scale-CNN	GCN	S^2GCN	Spectral-GraphSAGE	Spatial-GraphSAGE	S^2GraphSAGE
1	91.66±5.89	87.50±5.10	**100.00±0.00**	**100.00±0.00**	**100.00±0.00**	96.97±5.25	**100.00±0.00**	87.88±5.25	96.97±5.25	**100.00±0.00**
2	55.07±6.52	55.74±3.99	73.61±7.36	**81.73±6.17**	66.42±8.05	43.89±4.36	67.99±5.76	35.91±3.15	68.88±5.05	74.21±0.92
3	53.83±4.38	58.66±4.32	72.44±8.37	86.66±3.65	69.77±1.02	50.45±8.66	86.11±7.47	49.39±3.03	81.99±9.11	**88.28±4.15**
4	76.32±9.56	70.53±8.15	94.79±4.59	98.95±1.12	93.92±7.09	**99.31±0.60**	96.88±3.17	64.93±8.75	96.88±3.65	96.18±3.05
5	74.91±6.30	84.32±4.67	92.23±0.74	**94.97±2.64**	93.76±3.11	69.18±10.66	93.23±3.03	80.29±2.52	94.52±3.09	94.45±3.31
6	85.42±1.51	90.61±2.13	**99.36±0.36**	99.12±0.43	98.78±0.47	90.03±2.07	98.93±0.81	77.18±4.94	98.83±0.91	98.39±2.40
7	94.44±4.53	90.74±6.92	**100.00±0.00**	**100.00±0.00**	97.44±4.44	97.44±4.44	97.42±4.44	87.18±11.75	**100.00±0.00**	**100.00±0.00**
8	88.91±2.55	89.58±3.01	**100.00±0.00**	99.92±0.11	**100.00±0.00**	93.07±0.80	99.46±0.13	82.84±1.48	98.62±0.80	99.69±0.13
9	76.66±4.71	96.66±4.71	**100.00±0.00**	**100.00±0.00**	**100.00±0.00**	**100.00±0.00**	**100.00±0.00**	66.67±11.55	93.33±11.55	**100.00±0.00**
10	65.18±5.03	70.24±2.86	76.34±13.53	89.28±2.88	81.41±1.39	78.68±1.62	86.41±7.89	65.55±9.84	83.79±8.07	**90.18±5.37**
11	48.66±5.45	50.59±3.72	65.67±9.34	64.45±8.74	67.89±2.09	62.33±4.39	76.31±3.35	41.51±4.63	73.43±6.48	**80.32±4.71**
12	55.00±5.86	62.87±6.52	82.11±5.02	82.72±4.93	75.31±8.34	49.69±5.59	88.20±3.51	47.08±5.66	86.38±4.94	**93.49±1.87**
13	96.95±0.54	96.57±0.93	**100.00±0.00**	**100.00±0.00**	99.79±0.36	98.75±0.00	97.29±0.72	89.58±2.19	99.38±0.63	98.12±0.63
14	78.97±3.39	82.37±3.97	93.19±3.28	91.74±2.25	91.17±3.61	74.86±6.23	94.04±0.70	83.61±3.78	91.75±1.19	**94.35±0.94**
15	64.23±4.83	64.04±2.98	92.86±6.24	**98.53±1.67**	84.55±4.27	83.09±7.21	92.04±5.23	41.02±4.36	89.05±5.63	88.07±9.37
16	87.83±8.33	89.41±8.33	**100.00±0.00**	**100.00±0.00**	**100.00±0.00**	97.92±3.61	99.21±1.37	93.75±9.08	99.31±1.20	98.61±2.41
OA(%)	63.88±2.60	66.93±1.50	80.39±0.76	83.91±1.87	79.00±0.76	67.20±1.37	84.76±0.69	57.17±1.94	83.52±0.89	**87.43±1.48**
AA(%)	74.63±1.71	77.53±1.46	90.16±1.55	93.01±0.52	88.71±0.75	80.35±1.60	92.11±1.09	68.42±1.01	90.82±0.40	**93.40±0.92**
Kappa(%)	59.35±2.89	62.81±1.66	77.69±0.93	81.77±2.06	76.18±0.89	62.72±1.48	82.58±0.7	52.05±1.99	80.80±0.47	**85.66±1.66**

Bold entities are the best results.

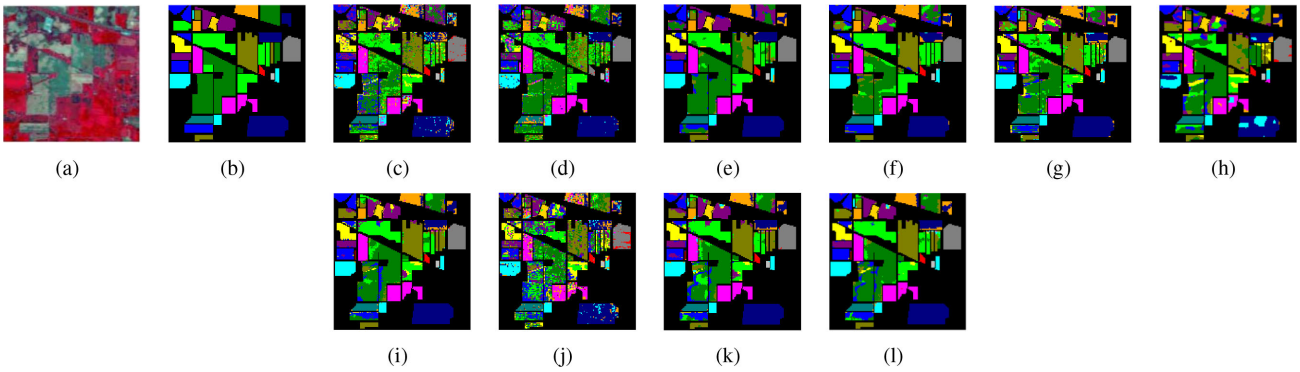

Fig. 6. Classification maps obtained by different methods on INDIAN PINES datasets. (a) False color image. (b) Ground truth. (c) MLP. (d) SVM. (e) 3D-CNN. (f) DR-CNN. (g) Multi-Scale-CNN. (h) GCN. (i) S^2 GCN. (j) Spectral-GraphSAGE. (k) Spatial-GraphSAGE. (l)S^2GraphSAGE.

TABLE VI
CLASSIFICATION RESULTS OF DIFFERENT METHODS FOR THE PAVIA UNIVERSITY DATASET

Class	MLP	SVM-RBF	3D-CNN	DR-CNN	Multi-Scale-CNN	GCN	S^2GCN	Spectral-GraphSAGE	Spatial-GraphSAGE	S^2GraphSAGE
1	64.37±3.05	65.14±5.45	77.86±3.92	**90.24±2.06**	79.44±4.98	-	-	70.99±4.83	86.33±6.33	84.87±4.29
2	54.66±1.17	59.19±4.45	83.40±5.31	78.81±5.92	**97.28±3.28**	-	-	68.31±4.43	91.49±0.4	95.29±1.06
3	47.55±1.71	27.69±2.46	74.56±5.17	84.56±4.65	85.82±0.86	-	-	68.97±7.07	**86.51±1.00**	85.96±1.83
4	92.19±2.25	95.25±2.14	92.01±1.15	**97.01±0.56**	96.73±1.25	-	-	89.80±4.08	84.71±4.93	83.63±0.98
5	98.83±0.03	99.18±0.14	**100.00±0.00**	99.97±0.03	**100.00±0.00**	-	-	98.03±0.25	**100.00±0.00**	99.52±0.40
6	62.66±4.82	70.42±9.49	80.02±3.01	92.11±6.18	71.90±5.54	-	-	70.04±5.52	**87.69±4.67**	85.85±6.24
7	90.02±2.21	90.10±1.45	95.27±3.06	90.37±1.77	83.40±0.60	-	-	85.84±4.51	94.11±1.51	**96.78±1.40**
8	66.38±0.51	87.03±2.81	91.48±3.54	89.68±3.86	82.46±7.66	-	-	81.06±8.39	92.81±3.18	**94.16±0.49**
9	99.74±0.10	99.92±0.05	**100.00±0.00**	99.77±0.18	99.85±0.25	-	-	**100.00±0.00**	98.34±0.45	97.67±0.19
OA(%)	63.87±0.17	67.93±0.55	84.24±2.65	86.10±1.85	90.35±0.58	-	-	73.90±3.01	90.29±0.64	**91.41±0.37**
AA(%)	75.16±0.01	77.11±0.16	88.29±0.49	91.39±0.28	89.43±1.52	-	-	81.63±0.77	91.16±0.51	**91.53±0.72**
Kappa(%)	55.56±0.42	60.27±0.25	79.63±3.19	82.21±2.15	87.14±0.81	-	-	66.99±3.53	87.24±0.83	**88.61±0.53**

Bold entities are the best results.

TABLE VII
CLASSIFICATION RESULTS OF DIFFERENT METHODS FOR THE KENNEDY SPACE CENTER DATASET

Class	MLP	SVM-RBF	3D-CNN	DR-CNN	Multi-Scale-CNN	GCN	S^2GCN	Spectral-GraphSAGE	Spatial-GraphSAGE	S^2GraphSAGE
1	86.27±1.58	89.78±1.89	97.57±1.14	98.46±0.30	98.67±0.10	91.90±1.06	**100.00±0.00**	86.5±2.47	**100.00±0.00**	99.7±0.35
2	83.72±2.77	84.66±0.58	90.23±4.54	92.42±0.82	86.11±2.50	80.69±1.15	98.65±1.54	82.83±4.29	96.47±2.14	**98.99±1.75**
3	76.40±16.54	56.78±22.21	88.94±7.36	97.47±0.81	95.74±2.68	48.42±5.45	99.53±0.48	86.73±2.01	**99.77±0.33**	99.37±1.10
4	22.52±2.54	26.42±19.23	73.11±8.89	73.27±1.82	64.49±9.22	56.45±6.14	71.66±7.70	59.91±13.67	69.57±8.88	**78.26±5.57**
5	63.35±2.24	38.42±2.00	60.63±10.45	**91.95±2.26**	75.43±0.61	48.67±6.24	84.77±10.53	58.02±3.29	79.31±1.22	83.05±8.19
6	55.61±2.06	41.37±2.47	**97.10±4.09**	93.84±2.23	89.95±4.22	54.24±9.11	71.92±4.93	46.20±2.31	67.67±1.15	80.80±5.76
7	85.77±2.26	89.33±2.17	95.55±3.42	**100.00±0.00**	**100.00±0.00**	40.88±11.42	98.89±1.92	83.33±7.07	**100.00±0.00**	99.44±0.96
8	61.09±0.93	44.80±8.03	97.15±3.85	94.12±0.95	93.40±1.28	49.39±5.90	97.75±3.44	82.39±6.97	**98.58±1.28**	98.53±2.54
9	78.50±3.00	75.98±6.67	96.91±1.62	99.93±0.01	**100.00±0.00**	79.16±4.72	100.00±0.00	93.26±0.30	**100.00±0.00**	**100.00±0.00**
10	78.96±1.77	65.95±1.81	**98.51±1.91**	77.25±3.03	81.76±0.19	85.93±3.89	96.66±2.93	89.84±0.59	93.60±1.58	96.56±5.95
11	92.11±2.18	89.88±1.15	**100.00±0.00**	99.73±1.11	95.86±5.86	97.83±3.76	96.35±3.55	90.64±0.38	96.53±1.89	99.64±0.62
12	82.73±1.29	83.72±2.10	95.19±2.94	89.66±0.54	69.22±4.63	83.28±0.95	**96.14±1.95**	78.82±0.31	90.50±3.24	94.83±1.40
13	99.40±0.62	99.85±0.14	**100.00±0.00**	92.82±1.63	89.21±14.79	97.02±2.73	**100.00±0.00**	97.73±0.00	**100.00±0.00**	99.43±0.30
OA(%)	79.98±0.85	75.98±0.31	95.19±0.34	93.00±0.19	90.90±1.04	79.02±0.70	96.02±0.87	85.15±1.29	94.84±1.04	**96.67±0.51**
AA(%)	74.34±1.63	68.23±0.15	91.61±0.32	92.38±0.17	88.45±1.32	79.89±1.22	93.25±1.65	79.76±1.77	91.69±1.43	**94.51±0.62**
Kappa(%)	77.64±0.96	73.14±0.34	94.61±0.39	92.15±0.22	89.79±1.17	76.47±0.79	95.53±0.98	83.38±1.42	94.21±1.17	**96.27±0.57**

Bold entities are the best results.

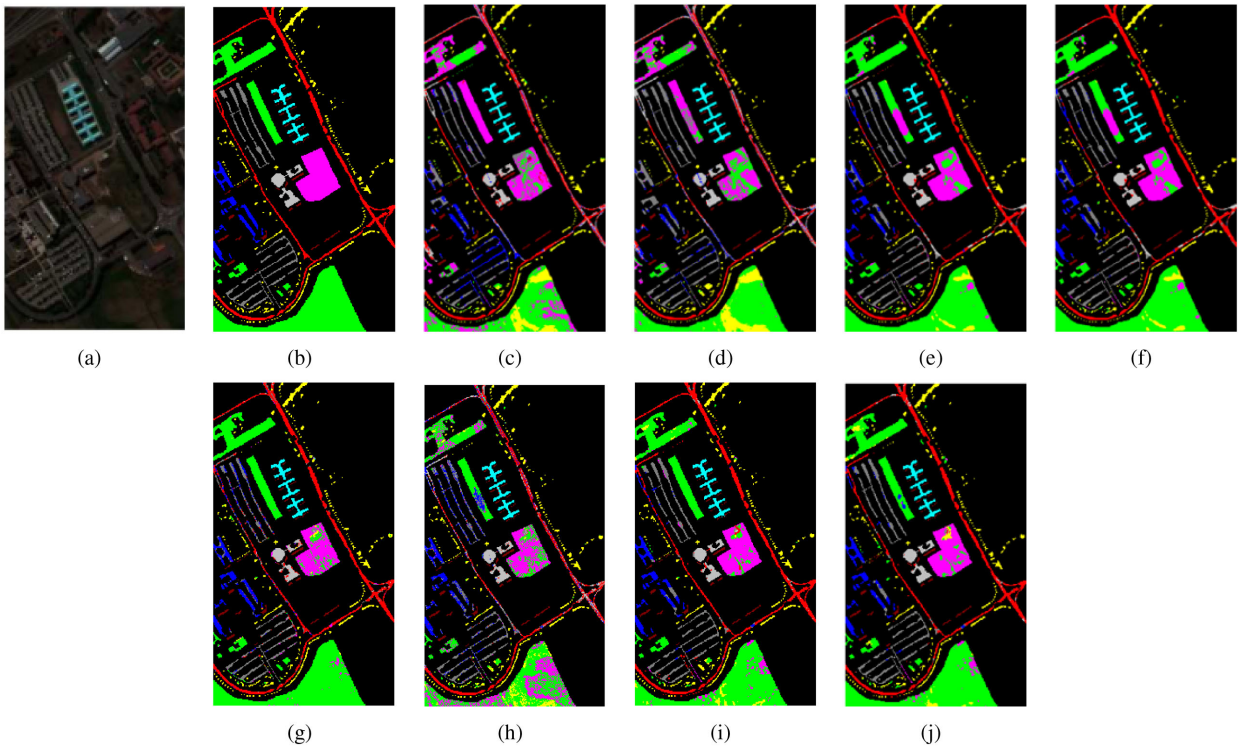

Fig. 7. Classification maps obtained by different methods on Pavia University dataset. (a) False color image. (b) Ground truth. (c) MLP. (d) SVM. (e) 3D-CNN. (g) DR-CNN. (f) Multi-Scale-CNN. (h) Spectral-GraphSAGE. (i) Spatial-GraphSAGE. (j) S^2GraphSAGE.

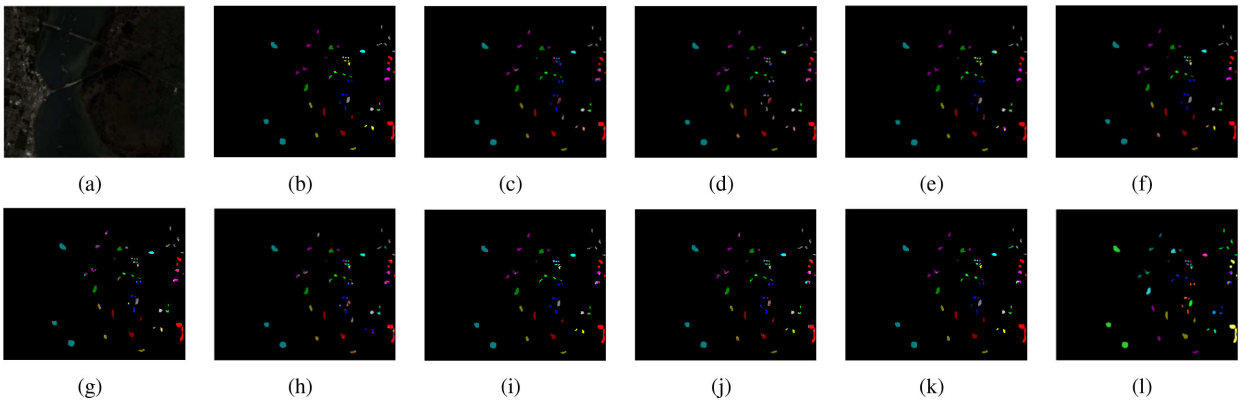

Fig. 8. Classification maps obtained by different methods on Kennedy Space Center dataset. (a) False color image. (b) Ground truth. (c) MLP. (d) SVM. (e) 3D-CNN. (f) DR-CNN. (g) Multi-Scale-CNN. (h) GCN. (i) S^2 CNN. (j) Spectral-GraphSAGE. (k) Spatial-GraphSAGE. (l) S^2GraphSAGE.

G. Running Time and Memory Consumption Analysis

Table VIII reports the running time of deep models including 3D-CNN [42], DR-CNN [22], Multi-Scale-CNN [43], GCN [35], S^2 GCN [37], and the proposed S^2GraphSAGE on three datasets. The codes for all methods are written in Python, and run on a desktop computer with a 2.90-GHz Intel Xeon W-2102 CPU with 16 GB of RAM and a GTX 1660 GPU. From Table VIII, we can see that the CNN-based methods consume massive computation time since they need to train a large number of parameters. In contrast, GCN employs a fixed graph for convolution with the first-order neighbor nodes, and thus the number of parameters is greatly reduced, which leads to fast running. However, the GCN constructs the graph only on the first-order neighbor nodes, it may not extract the information well. For our proposed method, S^2GraphSAGE can aggregate multi-order neighbor nodes, which takes more time, but improves accuracy. As a result, they need less computational time than CNN-based methods.

The drawback of the GCN method is that it needs to construct the whole graph, which needs large memory. Take Pavia University Dataset for an example, the GCN method directly constructs the adjacency matrix using the nonbackground samples,

TABLE VIII
RUNNING TIME COMPARISON (IN SECONDS) OF DIFFERENT METHODS

	3D-CNN	DR-CNN	Multi-Scale-CNN	GCN	S^2GCN	S^2GraphSAGE
IP	7277	15389	2326	71	148	2485
PAVIAU	4963	9550	1669	-	-	1900
KSC	6820	15023	2260	64	132	1709

IP denotes Indian Pines Dataset, PAVIAU denotes University of Pavia Dataset, and KSC denotes Kennedy Space Center Dataset.

and needs the double space for storage. It needs about 13.6-G memory to save the graph and more memory for convolution. Therefore, for the Pavia University Dataset, it always gets the error of memory stack overflow. Our proposed method solves this problem by constructing the graph using the inductive method. The memory consumption for Pavia University Dataset is about 334 M, which is greatly reduced compared with the GCN method.

V. CONCLUSION

In this article, we propose a novel hyperspectral image classification method with the spectral-spatial graph using the inductive representation learning network (S^2GraphSAGE). The proposed classification method mainly includes two parts: Constructing a spectral–spatial graph and introducing inductive learning. Constructing a spectral–spatial graph can aggregate more appropriate neighbor nodes information. Therefore, the spectral and spatial information can be extracted to enhance performance. Introducing inductive learning overcomes the disadvantage of using large memory. Moreover, S^2GraphSAGE can aggregate multi-order neighbor nodes information, which also leads to better performance. The classification results show the effectiveness of the proposed methods.

REFERENCES

[1] D. Haboudane, J. R. Miller, E. Pattey, P. J. Zarco-Tejada, and I. B. Strachan, "Hyperspectral vegetation indices and novel algorithms for predicting green LAI of crop canopies: Modeling and validation in the context of precision agriculture," *Remote Sens. Environ.*, vol. 90, no. 3, pp. 337–352, 2004.

[2] H. Grahn and P. Geladi, *Techniques and Applications of Hyperspectral Image Analysis*. Chichester, U.K.: Wiley, 2007.

[3] X.-Z. Shi, M. Aspandiar, and D. Oldmeadow, "Spectral reflectance variations and mineral transformations during the formation of acid sulphate soil in an incubation experiment," *Int. J. Remote Sens.*, vol. 35, no. 23, pp. 7959–7977, 2014.

[4] H. Kwon and N. M. Nasrabadi, "Kernel RX-algorithm: A nonlinear anomaly detector for hyperspectral imagery," *IEEE Trans. Geosci. Remote Sens.*, vol. 43, no. 2, pp. 388–397, Feb. 2005.

[5] H. Xu and X.-j. Wang, "Applications of multispectral/hyperspectral imaging technologies in military," *Infrared Laser Eng.*, vol. 36, no. 1, p. 13, 2007.

[6] S. Kumar, J. Ghosh, and M. Crawford, "Best-bases feature extraction algorithms for classification of hyperspectral data," *IEEE Trans. Geosci. Remote Sens.*, vol. 39, no. 7, pp. 1368–1379, Jul. 2001.

[7] E. Blanzieri and F. Melgani, "Nearest neighbor classification of remote sensing images with the maximal margin principle," *IEEE Trans. Geosci. Remote Sens.*, vol. 46, no. 6, pp. 1804–1811, Jun. 2008.

[8] Y. Tarabalka, M. Fauvel, J. Chanussot, and J. A. Benediktsson, "SVM-and MRF-based method for accurate classification of hyperspectral images," *IEEE Geosci. Remote Sens. Lett.*, vol. 7, no. 4, pp. 736–740, Oct. 2010.

[9] M. Fauvel, J. A. Benediktsson, J. Chanussot, and J. R. Sveinsson, "Spectral and spatial classification of hyperspectral data using SVMs and morphological profiles," *IEEE Trans. Geosci. Remote Sens.*, vol. 46, no. 11, pp. 3804–3814, Nov. 2008.

[10] P. K. Murphy and M. A. Kolodner, "Implementation of a multiscale Bayesian classification approach for hyperspectral terrain categorization," *Proc. SPIE*, vol. 4816, 2002, pp. 278–287.

[11] J. Chen and R.-S. Wang, "Applied research of gaussian maximum likelihood classification in hyperspectral classification," *J. Comput. Appl.*, no. 8, p. 37, 2006.

[12] X. Sun, Q. Qu, N. M. Nasrabadi, and T. D. Tran, "Structured priors for sparse-representation-based hyperspectral image classification," *IEEE Geosci. Remote Sens. Lett.*, vol. 11, no. 7, pp. 1235–1239, Jul. 2014.

[13] H. Yuan, "Robust patch-based sparse representation for hyperspectral image classification," *Int. J. Wavelets, Multiresolution Inf. Process.*, vol. 15, no. 03, 2017, Art. no. 1750028.

[14] A. Villa, J. A. Benediktsson, J. Chanussot, and C. Jutten, "Hyperspectral image classification with independent component discriminant analysis," *IEEE Trans. Geosci. Remote Sens.*, vol. 49, no. 12, pp. 4865–4876, Dec. 2011.

[15] L. Sun, C. Ma, Y. Chen, Y. Zheng, and B. Jeon, "Low rank component induced spatial-spectral kernel method for hyperspectral image classification," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 30, no. 10, pp. 3829–3842, Oct. 2020.

[16] L. Sun, C. Ma, Y. Chen, H. J. Shim, Z. Wu, and B. Jeon, "Adjacent superpixel-based multiscale spatial-spectral kernel for hyperspectral classification," *IEEE J. Sel. Top. Appl. Earth Observ. Remote Sens.*, vol. 12, no. 6, pp. 1905–1919, Jun. 2019.

[17] Y. Qian, M. Ye, and J. Zhou., "Hyperspectral image classification based on structured sparse logistic regression and three-dimensional wavelet texture features," *IEEE Trans. Geosci. Remote Sens.*, vol. 51, no. 4, pp. 2276–2291, Apr. 2013.

[18] J. Li, J. M. Bioucas-Dias, and A. Plaza, "Semisupervised hyperspectral image classification using soft sparse multinomial logistic regression," *IEEE Geosci. Remote Sens. Lett.*, vol. 10, no. 2, pp. 318–322, Mar. 2013.

[19] L. Shen and S. Jia, "Three-dimensional Gabor wavelets for pixel-based hyperspectral imagery classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 49, no. 12, pp. 5039–5046, Dec. 2011.

[20] S. Jia, J. Hu, Y. Xie, L. Shen, X. Jia, and Q. Li, "Gabor cube selection based multitask joint sparse representation for hyperspectral image classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 54, no. 6, pp. 3174–3187, Jun. 2016.

[21] Y. Chen, Y. Wang, Y. Gu, X. He, P. Ghamisi, and X. Jia, "Deep learning ensemble for hyperspectral image classification," *IEEE J. Sel. Top. Appl. Earth Observ. Remote Sens.*, vol. 12, no. 6, pp. 1882–1897, Jun. 2019.

[22] M. Zhang, W. Li, and Q. Du, "Diverse region-based CNN for hyperspectral image classification," *IEEE Trans. Image Process.*, vol. 27, no. 6, pp. 2623–2634, Jun. 2018.

[23] Y. Chen, Z. Lin, X. Zhao, G. Wang, and Y. Gu, "Deep learning-based classification of hyperspectral data," *IEEE J. Sel. Top. Appl. Earth Observ. Remote Sens.*, vol. 7, no. 6, pp. 2094–2107, Jun. 2014.

[24] T. Chao, H. Pan, Y. Li, and Z. Zou, "Unsupervised spectral-spatial feature learning with stacked sparse autoencoder for hyprspectral imagery classification," *IEEE Geosci. Remote Sens. Lett.*, vol. 12, no. 12, pp. 2438–2442, Dec. 2015.

[25] Y. Chen, Z. Xing, and X. Jia, "Spectral–spatial classification of hyperspectral data based on deep belief network," *IEEE J. Sel. Top. Appl. Earth Observ. Remote Sens.*, vol. 8, no. 6, pp. 1–12, Jun. 2015.

[26] P. Zhong, Z. Gong, S. Li, and C.-B. Schonlieb, "Learning to diversify deep belief networks for hyperspectral image classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 55, no. 6, pp. 3516–3530, Jun. 2017.

[27] W. Hu, Y. Huang, L. Wei, F. Zhang, and H. Li, "Deep convolutional neural networks for hyperspectral image classification," *J. Sensors*, vol. 2015, 2015, Art. no. 258619.

[28] S. Mei, J. Ji, J. Hou, X. Li, and Q. Du, "Learning sensor-specific spatial-spectral features of hyperspectral images via convolutional neural networks," *IEEE Trans. Geosci. Remote Sens.*, vol. 55, no. 8, pp. 4520–4533, Aug. 2017.
[29] H. Lee and H. Kwon, "Going deeper with contextual CNN for hyperspectral image classification," *IEEE Trans. Image Process.*, vol. 26, no. 10, pp. 4843–4855, Oct. 2017.
[30] A. B. Hamida, A. Benoit, P. Lambert, and C. B. Amar, "3-D deep learning approach for remote sensing image classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 56, no. 8, pp. 4420–4434, Aug. 2018.
[31] X. J. Zhu, "Semi-supervised learning literature survey," Dept. Comput. Sci., Univ. Wisconsin–Madison, Madison, WI, USA, Tech. Rep., 2005.
[32] L. Yang, S. Yang, P. Jin, and R. Zhang, "Semi-supervised hyperspectral image classification using spatio-spectral laplacian support vector machine," *IEEE Geosci. Remote Sens. Lett.*, vol. 11, no. 3, pp. 651–655, Mar. 2014.
[33] K. Tan, J. Hu, J. Li, and P. Du, "A novel semi-supervised hyperspectral image classification approach based on spatial neighborhood information and classifier combination," *ISPRS J. Photogrammetry Remote Sens.*, vol. 105, pp. 19–29, 2015.
[34] Z. A. Andekah, M. Naderan, and G. Akbarizadeh, "Semi-supervised hyperspectral image classification using spatial-spectral features and superpixel-based sparse codes," in *Proc. Iranian Conf. Elect. Eng.*, 2017, pp. 2229–2234.
[35] T. N. Kipf and M. Welling, "Semi-supervised classification with graph convolutional networks," in *Proc. 5th Int. Conf. Learn. Representations*, Toulon, France, Apr. 2017. [Online]. Available: https://openreview.net/forum?id=SJU4ayYgl
[36] S. Wan, C. Gong, P. Zhong, B. Du, L. Zhang, and J. Yang, "Multiscale dynamic graph convolutional network for hyperspectral image classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 58, no. 5, pp. 3162–3177, May 2020.
[37] A. Qin, Z. Shang, J. Tian, Y. Wang, T. Zhang, and Y. Y. Tang, "Spectral–spatial graph convolutional networks for semisupervised hyperspectral image classification," *IEEE Geosci. Remote Sens. Lett.*, vol. 16, no. 2, pp. 241–245, Feb. 2019.
[38] P. Sellars, A. I. Aviles-Rivero, and C. Schnlieb, "Superpixel contracted graph-based learning for hyperspectral image classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 58, no. 6, pp. 4180–4193, Jun. 2020.
[39] W. Hamilton, Z. Ying, and J. Leskovec, "Inductive representation learning on large graphs," in *Proc. Adv. Neural Inf. Process. Syst.*, 2017, pp. 1024–1034.
[40] A. Qin, C. Liu, Z. Shang, and J. Tian, "Spectral-spatial graph convolutional networks for semel-supervised hyperspectral image classification," in *Proc. Int. Conf. Wavelet Anal. Pattern Recognit.*, 2018, pp. 89–94.
[41] B.-C. Kuo, H.-H. Ho, C.-H. Li, C.-C. Hung, and J.-S. Taur, "A kernel-based feature selection method for SVM with RBF kernel for hyperspectral image classification," *IEEE J. Sel. Top. Appl. Earth Observ. Remote Sens.*, vol. 7, no. 1, pp. 317–326, Jan. 2014.
[42] Y. Chen, H. Jiang, C. Li, X. Jia, and P. Ghamisi, "Deep feature extraction and classification of hyperspectral images based on convolutional neural networks," *IEEE Trans. Geosci. Remote Sens.*, vol. 54, no. 10, pp. 6232–6251, Oct. 2016.
[43] F. Zhang and X. Yang, "Deep multi-scale convolutional neural network for hyperspectral image classification," *Proc. SPIE*, vol. 10615, 2018, Art. no. 106152Z.

Pan Yang received the B.S. degree from the Harbin University of Science and Technology, Harbin, China, in 2018. He is currently working toward the M.S. degree with the Faculty of Information Technology, Beijing University of Technology, Beijing, China.

His research interests include pattern recognition and computer vision, and image processing.

Lei Tong received the B.E. degree in measurement and control technology and instrumentation, and the M.E. degree in measurement technology and automation devices from Beijing Jiaotong University, Beijing, China, in 2010 and 2012, respectively. He received the Ph.D. degree in engineering from Griffith University, Brisbane, Australia, in 2016.

Currently, he is an Associate Professor with Faculty of Information Technology, Beijing University of Technology, Beijing, China. His current research interests include signal and image processing, pattern recognition, and remote sensing.

Bin Qian (Member, IEEE) received the Ph.D degree in computer science and technology from Nanjing University of Science and Technology, Nanjing, China, in 2018.

He is currently a senior engineer with Traffic Management Research Institute of the Ministry of Public Security. His research interests include pattern recognition and computer vision, nonnegative matrix factorization, and subspace clustering

Zheng Gao received the B.S. degree from the Shijiazhuang University, Hebei, China, in 2015. He is currently working toward the M.S. degree with the Faculty of Information Technology, Beijing University of Technology, Beijing, China.

His research interests include pattern recognition and computer vision, and image processing.

Jing Yu (Member, IEEE) received the Ph.D. degree in electronic engineering from Tsinghua University, Beijing, China, in 2011.

She is currently an Associate Professor with the Faculty of Information Technology, Beijing University of Technology, Beijing. Her research interests include image processing and pattern recognition.

Chuangbai Xiao (Member, IEEE) received the Ph.D. degree from the Department of Automation, Tsinghua University, Beijing, China, in 1995.

He is currently a Professor with the Faculty of Information Technology, Beijing University of Technology, Beijing, China. His research interests include pattern recognition, digital image processing, digital signal processing, computer network security, and cloud computing.