此论文代码并未开源！！

DRGCN: Dual Residual Graph Convolutional Network for Hyperspectral Image Classification

Rong Chen, Guanghui Li, and Chenglong Dai

Abstract—**Recently, graph convolutional network (GCN) has drawn increasing attention in hyperspectral image (HSI) classification, as it can process arbitrary non-Euclidean data. However, dynamic GCN that refines the graph heavily relies on the graph embedding in the previous layer, which will result in performance degradation when the embedding contains noise. In this letter, we propose a novel dual residual graph convolutional network (DRGCN) for HSI classification that integrates two adjacency matrices of dual GCN. In detail, one GCN applies a soft adjacency matrix to extract spatial features, whereas the other utilizes the dynamic adjacency matrix to extract global context-aware features. Subsequently, the features extracted by dual GCN are fused to make full use of the complementary and correlated information among two graph representations. Moreover, we introduce residual learning to optimize graph convolutional layers during the training process, to alleviate the over-smoothing problem. The advantage of dual GCN is that it can extract robust and discriminative features from HSIs. Extensive experiments on four HSI datasets, including Indian Pines, Pavia University, Salinas, and Houston University, demonstrate the effectiveness and superiority of our proposed DRGCN, even with small-sized training data.**

Index Terms—**Graph convolutional network (GCN), graph representation, hyperspectral image (HSI) classification, residual learning.**

I. Introduction

HYPERSPECTRAL images (HSIs), containing abundant spectral information in a large number of spectral bands, have been widely used in environmental monitoring, geological exploration, and control. With improved satellite imaging technology, high spatial resolution HSIs are available, which makes the HSI contain rich contextual information. So far, HSI classification has attracted increasing attention in the field of remote sensing, whose goal is to classify each pixel in an HSI image based on its reflectance spectrum.

In recent years, deep-learning methods [e.g., stacked autoencoders (SAEs), recurrent neural networks (RNNs), and convolutional neural networks (CNNs)] have been proposed for HSI classification, thanks to their robust feature learning capacities. As the most extensive and effective technique for extracting spatial–spectral features from HSIs, CNN-based methods can not only extract robust features, but also alleviate the disaster of high dimensionality [1]. So far, various CNN-based methods have been developed for HSI classification. For instance, Chen *et al.* [2] explored the effectiveness of three convolution operations (e.g., 1-D CNN, 2-D CNN, and 3-D CNN) for HSI classification, and the experimental results show that 3-D CNN can extract more profound and discriminative features. Besides, Hamida *et al.* [3] proposed a 3-D CNN architecture for HSI classification with a low computing consumption cost. Furthermore, deep reinforcement learning [4] is applied to reduce the spectral dimension of HSIs. However, conventional convolution operations are conducted on the fixed structures in regular regions, and it is challenging to directly apply the CNN to graph-structured data.

Recently, graph convolutional networks (GCNs), proposed by Kipf and Welling [5], have been drawing increasing attention, thanks to their ability on non-Euclidean data based on Fourier transform. Many researchers have been devoted to combining graph-based methods and neural networks to establish more flexible methods. Qin *et al.* [6] proposed a spectral–spatial GCN method (S^2GCN), which performs at the pixel level to take full advantage of spectral–spatial neighbor information among pixels for semisupervised HSI classification. Subsequently, Sha *et al.* [7] proposed graph attention networks for HSI classification to adaptively capture the important nodes among neighborhoods. Besides, Tong *et al.* [8] proposed attention-weighted GCNs to extract the internal relationships of objects. To obtain a more reliable graph, Wan *et al.* [9] proposed a context-aware dynamic GCN (CAD-GCN) method at the superpixel level to learn long-range contextual information between graph nodes. Furthermore, in [10], a multi-scale dynamic GCN was proposed to extract multi-scale spectral–spatial information for HSI classification.

In those works mentioned above, on the one hand, the traditional GCN that conducts convolution on a fixed graph cannot describe the potential relationship between graph nodes. On the other hand, the dynamic GCN refines the graph heavily depending on graph embedding in the previous layer during the convolution process, so it will result in performance degradation when the graph embedding contains noise and it does not consider the global context information. To solve this problem, we propose a dual residual graph convolutional network (DRGCN) that adopts dual GCN with two adjacency matrices to extract spatial–spectral features from the predefined fixed graph and dynamic graph, respectively,

Manuscript received April 8, 2022; accepted April 26, 2022. Date of publication May 2, 2022; date of current version May 10, 2022. This work was supported in part by the National Natural Science Foundation of China under Grant 62072216 and in part by the Jiangsu Agriculture Science and Technology Innovation Fund under Grant CX(19)3087. *(Corresponding author: Guanghui Li.)*

The authors are with the School of Artificial Intelligence and Computer Science, Jiangnan University, Wuxi 214122, China (e-mail: jnuc_r@163.com; ghli@jiangnan.edu.cn; chenglongdai@jiangnan.edu.cn).

Digital Object Identifier 10.1109/LGRS.2022.3171536

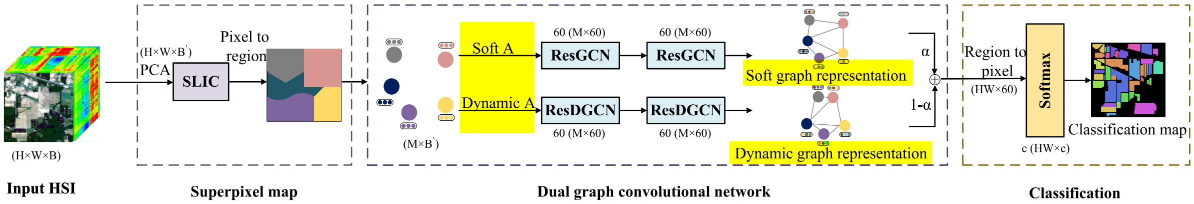

Fig. 1. Overall architecture of the proposed DRGCN. Soft A and dynamic A are the soft adjacency matrix and dynamic adjacency matrix, respectively. The superpixel map segments HSIs into homogeneous regions and obtains their graph data. The dual convolutional network obtains soft graph representation and dynamic graph representation by soft adjacency matrix and dynamic adjacency matrix, respectively, and then they are fused to get the final graph representation. The classification predicts the categories of all pixels in the HSI.

which can not only extract the contextual relations between graph nodes, but also can alleviate the problem of performance degradation due to inaccurate graph embedding. Moreover, the dual-branch structure can enhance the discriminative ability of graph nodes and increase the diversity of node features. Another advantage of dual GCN is that the different graph representations are fused to improve the classification performance further. The main contributions of our work are highlighted as follows.

1) We design a dual GCN with two adjacency matrices as inputs to extract spatial and global context information, which can make full use of the complementary and correlated information among different graph representations.
2) Different from other dynamic adjacency matrix updating strategies that only rely on the graph embedding in the previous layer, we also combine global representation to describe the context-aware information of graph nodes.
3) To alleviate the over-smoothing problem, residual learning is introduced to optimize graph convolutional layers during the training process.

The rest of this letter is organized as follows. The proposed DRGCN method is introduced in Section II. The experimental results on four HSI datasets are shown in Section III. Finally, Section IV states the conclusion of this letter.

II. Method

In this section, we introduce the proposed DRGCN, whose overall architecture is shown in Fig. 1. First, the raw HSI is converted into graph data by a superpixel map. Then, the graph data is fed into dual GCN to extract robust and discriminative features. Finally, the categories of all pixels can be predicted by the trained network.

A. Superpixel Map

Generally, HSIs contain a large number of pixels, which will limit the receptive field of graph convolution at the pixel level [9]. Superpixel maps can well conserve image structures by segmenting the HSI into many coherent regions. Specifically, given an HSI as $\mathbf{X} \in \mathbb{R}^{H\times W\times B}$, in which H, W, and B are the height, width, and the number of spectral bands, respectively. First, the principal component analysis (PCA) is utilized to reduce the spectral dimension of HSIs. In this letter, we choose the first 60 principal components as the spectral length, and then the HSI is denoted as $\mathbf{X} \in \mathbb{R}^{H\times W\times B'}$. Subsequently, the simple linear iterative clustering (SLIC) [11] is applied to produce a set of compact superpixels represented as $S = \{S_1, S_2, \ldots, S_M\}$, where $S_j = \{x_i^j\}_{i=1}^{n_j}$ represents the jth superpixel and it consists of n_j pixels ($S_i \cap S_j = \phi, H \times W = \sum_{i=1}^{M} n_i$ and $\forall i \neq j$), x_i^j is the ith pixel in S_j, and M is the number of superpixels. Each superpixel is regarded as a graph node after segmentation, significantly improving computational efficiency. The features of each superpixel are the mean spectral bands of the pixels in the corresponding superpixel, which is denoted as

$$\mathbf{P}_i = \sum_{j=1}^{n_i} x_j^i \tag{1}$$

where $\mathbf{P}_i$ is the feature representation of the ith node. According to the superpixel matrix S, we can obtain the pixel-region assignment matrix $\mathbf{V} \in \mathbb{R}^{HW\times M}$, that is,

$$\mathbf{V}_{i,j} = \begin{cases} 1, & \text{if } \widetilde{\mathbf{X}}_i \in S_j \\ 0, & \text{otherwise} \end{cases} \tag{2}$$

where $\mathbf{V}_{i,j}$ represents the ith pixel belonging to the jth superpixel, and $\widetilde{\mathbf{X}}_i$ is the output of PCA. In summary, we project HSIs into graph nodes (i.e., superpixels) based on the superpixel map, greatly improving computational efficiency and preserving structural information of HSIs, which is conducive to HSI classification.

B. Dual Graph Convolutional Network

GCNs [5], developed for semi-supervised classification, have been widely used in computer vision [12], which can process the data with arbitrary non-Euclidean structure. Given an undirected graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, where the $\mathcal{V}$ and $\mathcal{E}$ are the sets of nodes and edges, respectively, they are encoded into a node representation matrix $\mathbf{H}$ and adjacency matrix $\mathbf{A}$. Formally, the formulations of convolution operation in GCNs can be computed by

$$\mathbf{H}^{(l+1)} = \sigma\left(\widetilde{\mathbf{D}}^{-\frac{1}{2}}\widetilde{\mathbf{A}}\widetilde{\mathbf{D}}^{-\frac{1}{2}}\mathbf{H}^{(l)}\mathbf{W}^{(l+1)}\right) \tag{3}$$

where $\mathbf{H}^{(l+1)}$ is the output of the $(l+1)$th layer, $\sigma(\cdot)$ denotes the activation function, $\mathbf{W}^{(l+1)}$ represents the trainable parameter, $\widetilde{\mathbf{A}} = \mathbf{A} + \mathbf{I}$ is an adjacency matrix with self-loops, and $\widetilde{\mathbf{D}}_{ii} = \sum_j \widetilde{\mathbf{A}}_{ij}$ is the degree matrix.

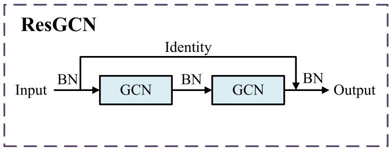

Fig. 2. Illustration of the ResGCN module. BN denotes the batch normalization (BN).

Different from the traditional dynamic GCN-based methods that heavily rely on graph embedding in the previous layer when refining the graph, we propose a dual residual GCN to extract diverse node features by using two adjacency matrices (see Fig. 1). To alleviate the over-smoothing problem, we introduce residual learning [13] to optimize GCN layers as the identity mapping without suffering from degradation of classification accuracy. Soft A (soft adjacency matrix), computed by Euclidean distance, is fed into ResGCN (residual graph convolution) to obtain the soft graph representation, which characterizes spatial information of graph nodes. Dynamic A (dynamic adjacency matrix), updated by the graph embedding in the previous layer, is fed into ResDGCN (residual dynamic graph convolution) to obtain the dynamic graph representation, which describes the correlation information and global information among graph nodes. Finally, both graph representations are fused by a linear transformation.

1) Residual Graph Convolution: The structure of ResGCN module is shown in Fig. 2 which contains two GCN layers. The BN [14] is utilized to accelerate the convergence of the network. The residual learning tries to optimize the two GCN layers as an identity mapping by employing short connections. In the ResGCN module, Soft A is used as the input of GCN, calculated by the spatial distance between nodes and fixed during the training process

$$\mathbf{A}_{\text{gcn}}^{ij} = \begin{cases} e^{-\gamma \|\mathbf{P}_i - \mathbf{P}_j\|^2}, & \text{if } \mathbf{P}_i \in \mathrm{N}(\mathbf{P}_j) \text{ or } \mathbf{P}_j \in \mathrm{N}(\mathbf{P}_i) \\ 0, & \text{otherwise} \end{cases} \qquad (4)$$

where $\mathbf{P}_i$ is a node, and $\mathrm{N}(\mathbf{P}_j)$ is the set of neighbors of $\mathbf{P}_j$. Following [9], parameter γ is set to 0.2. Finally, the graph convolution is denoted as

$$\mathbf{H}_{\text{gcn}}^{(l+1)} = \sigma\left(\widetilde{\mathbf{D}}^{-\frac{1}{2}}\widetilde{\mathbf{A}}_{\text{gcn}}\widetilde{\mathbf{D}}^{-\frac{1}{2}}\mathbf{H}^{(l)}\mathbf{W}^{(l+1)}\right) \qquad (5)$$

where $\widetilde{\mathbf{A}}_{\text{gcn}} = \mathbf{A}_{\text{gcn}} + \mathbf{I}$, and $\widetilde{\mathbf{D}}_{ii} = \sum_j \widetilde{\mathbf{A}}_{\text{gcn}}^{ij}$. As a result, we obtain the soft graph representation $\mathbf{H}_{\text{gcn}}$.

2) Residual Dynamic Graph Convolution: The structure of the ResDGCN module is shown in Fig. 3. Unlike ResGCN, the adjacency matrix in ResDGCN, generated by initial graph representation, is dynamic and updated during the convolution process, which combines with global representation to describe the context-aware information of graph nodes. Fig. 4 illustrates the dynamic adjacency matrix module. We can observe that the dynamic adjacency matrix $\mathbf{A}_{\text{dgcn}}$ is updated by the graph embedding in the previous layer, and it is defined as

$$\mathbf{A}_{\text{dgcn}} = \sigma\left((\mathbf{W}_c\hat{\mathbf{H}})(\mathbf{W}_c\hat{\mathbf{H}})^T\right) \qquad (6)$$

where $\mathbf{W}_c$ is the weights of the convolutional layer, σ denotes the sigmoid activation function, $\hat{\mathbf{H}}$ is obtained by concatenating $\mathbf{H}$, and its global representations $\mathbf{h}_g$, which is acquired by summation and a convolutional layer. Specifically, the global representations $h_g = \text{conv}(\sum_{i=1}^{M} h_i)$ with conv and M being the 1-D convolutional layer and the number of nodes. $\hat{\mathbf{H}}$ is denoted as

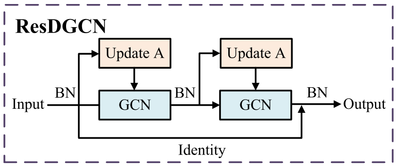

Fig. 3. Illustration of the ResDGCN module. BN denotes the BN.

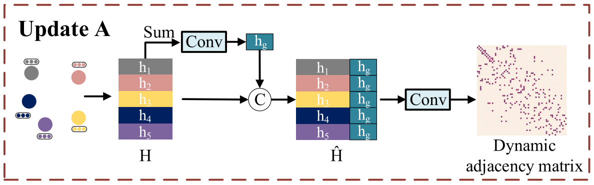

Fig. 4. Illustration of the dynamic adjacency matrix module.

$$\hat{\mathbf{H}} = [(\mathbf{h}_1; \mathbf{h}_g), (\mathbf{h}_2; \mathbf{h}_g), \ldots, (\mathbf{h}_M; \mathbf{h}_g)]. \qquad (7)$$

Based on this, the $\mathbf{A}_{\text{dgcn}}$ can be dynamically updated by graph embedding containing context-aware information. The dynamic GCN can be expressed as

$$\mathbf{H}_{\text{dgcn}}^{(l+1)} = \sigma\left(\widehat{\mathbf{D}}^{-\frac{1}{2}}\widetilde{\mathbf{A}}_{\text{dgcn}}\widehat{\mathbf{D}}^{-\frac{1}{2}}\mathbf{H}^{(l)}\mathbf{W}^{(l+1)}\right) \qquad (8)$$

where $\widetilde{\mathbf{A}}_{\text{dgcn}} = \mathbf{A}_{\text{dgcn}} + \mathbf{I}$, and $\widehat{\mathbf{D}}_{ii} = \sum_j \widetilde{\mathbf{A}}_{\text{dgcn}}^{ij}$. As a result, we obtain the dynamic graph representation $\mathbf{H}_{\text{dgcn}}$. To explore the complementary and correlated information among two graph representations, we use weighted summation to fuse two representations. The final graph representation $\mathbf{H}_{\text{final}}$ can be defined as

$$\mathbf{H}_{\text{final}} = \alpha\mathbf{H}_{\text{gcn}} + (1-\alpha)\mathbf{H}_{\text{dgcn}} \qquad (9)$$

where α is a tradeoff parameter between $\mathbf{H}_{\text{gcn}}$ and $\mathbf{H}_{\text{dgcn}}$.

C. Classification

After conducting the dual GCN on superpixel level, the final graph representation (region-level features) $\mathbf{H}_{\text{final}}$ needs to be converted into pixel level. To be specific, the assignment of regions to pixels is achieved by linearly interpolating pixel features by the assignment matrix $\mathbf{V}$. Consequently, the output of DRGCN can be denoted as

$$\mathbf{O} = \mathbf{V}\mathbf{H}_{\text{final}}. \qquad (10)$$

In DRGCN, the classic cross-entropy error is used as classification loss, that is,

$$\mathcal{L}_C = -\sum_{i=1}^{N}\sum_{j=1}^{c}\mathbf{Y}_{ij}\log\left(\mathbf{O}_{ij}\right) \qquad (11)$$

where N and c are the number of pixels and classes, respectively. $\mathbf{Y}$ denotes the label matrix.

TABLE I

CLASSIFICATION ACCURACY (%) ON THE INDIAN PINES DATASET

Class No.	SVM	2DCNN	3DCNN	GCN	S^2GCN	CAD-GCN	DRGCN
1	94.37	**99.37**	93.75	76.25	96.25	95.89	97.51
2	61.53	58.23	65.68	59.08	75.16	88.16	**88.26**
3	67.50	23.92	63.37	51.35	87.63	90.02	**90.55**
4	81.49	34.49	85.94	**97.72**	94.68	95.74	95.45
5	86.24	40.72	85.56	80.44	88.23	**93.55**	93.37
6	80.44	58.26	86.14	76.34	**97.57**	96.82	95.42
7	87.69	**100.00**	96.15	48.46	99.98	97.88	**100.00**
8	75.31	37.25	90.51	98.51	99.03	**99.91**	99.73
9	97.05	98.00	**100.00**	34.12	**100.00**	**100.00**	**100.00**
10	47.47	48.99	63.98	72.52	**89.17**	87.89	88.95
11	30.07	33.57	46.67	72.44	80.11	86.53	**88.46**
12	32.31	25.96	55.08	47.01	92.73	92.14	**93.64**
13	93.73	98.15	95.57	97.71	97.77	99.21	**100.00**
14	70.33	55.48	89.73	72.51	96.98	97.73	**97.82**
15	41.37	10.97	52.74	53.42	97.19	98.48	**98.87**
16	93.59	84.61	95.76	94.12	**100.00**	**100.00**	**100.00**
OA	56.11	43.37	66.20	67.67	89.43	92.06	**92.26**
AA	71.34	56.75	78.54	69.49	93.77	95.27	**95.55**
Kappa	51.23	38.02	62.29	63.82	87.96	90.94	**91.11**

TABLE II

CLASSIFICATION ACCURACY (%) ON THE PAVIAU DATASET

Class No.	SVM	2DCNN	3DCNN	GCN	S^2GCN	CAD-GCN	DRGCN
1	70.63	57.01	79.42	62.18	**91.21**	82.85	84.21
2	69.06	51.67	69.36	77.95	84.88	91.82	**93.16**
3	70.95	48.41	84.73	83.71	**97.96**	93.21	97.14
4	91.77	77.52	**92.81**	85.55	75.34	83.73	79.21
5	98.98	96.56	99.25	95.77	98.40	97.23	**99.61**
6	72.54	46.85	78.12	92.14	97.61	98.77	**99.34**
7	89.88	75.24	89.48	98.24	97.58	**99.71**	99.21
8	80.36	83.97	85.76	67.08	**96.90**	96.79	93.35
9	99.32	86.22	**99.45**	90.15	83.12	80.11	85.34
OA	74.62	59.26	77.99	78.49	89.12	91.33	**91.93**
AA	82.61	69.27	86.53	83.64	91.45	91.61	**92.28**
Kappa	67.93	51.23	72.55	72.67	86.04	88.69	**89.46**

TABLE III

CLASSIFICATION ACCURACY (%) ON THE SALINAS DATASET

Class No.	SVM	2DCNN	3DCNN	GCN	S^2GCN	CAD-GCN	DRGCN
1	98.26	56.76	93.15	99.95	99.98	**100.00**	**100.00**
2	97.33	71.62	97.98	99.56	99.49	**100.00**	**100.00**
3	94.78	68.83	97.78	99.99	**100.00**	99.94	**100.00**
4	99.11	91.71	99.06	99.51	99.45	94.33	**99.58**
5	96.48	74.36	96.51	85.01	97.57	97.31	**98.02**
6	98.74	82.46	98.44	98.61	98.58	98.12	**99.97**
7	99.14	78.38	98.43	99.86	**99.91**	98.65	99.89
8	64.42	62.01	70.67	63.37	51.51	96.77	**99.46**
9	98.16	81.09	98.68	99.75	99.99	**100.00**	**100.00**
10	87.72	80.31	86.69	91.24	**97.88**	94.98	97.37
11	95.16	86.31	94.57	96.88	97.66	99.51	**100.00**
12	98.94	81.36	98.06	99.93	**99.98**	98.48	99.83
13	97.07	87.68	97.85	97.10	97.21	**98.53**	98.38
14	93.02	86.08	94.33	97.14	96.50	97.08	**98.01**
15	65.13	49.14	70.01	71.81	96.84	97.63	**99.73**
16	94.34	18.39	84.57	93.90	98.67	**99.72**	99.51
OA	85.61	67.24	82.53	86.65	88.78	98.11	**99.48**
AA	92.36	71.09	90.92	93.35	95.70	98.32	**99.36**
Kappa	84.02	64.06	80.71	85.16	87.61	97.89	**99.43**

TABLE IV

CLASSIFICATION ACCURACY (%) ON THE HOUSTON DATASET

Class No.	SVM	2DCNN	3DCNN	GCN	S^2GCN	CAD-GCN	DRGCN
1	94.88	56.25	93.62	84.05	**95.24**	93.88	82.80
2	92.74	41.69	93.72	69.95	87.09	**96.02**	93.38
3	98.63	90.72	98.33	**100.00**	98.35	94.43	98.95
4	93.43	43.74	93.35	50.12	86.11	**94.15**	90.03
5	95.78	60.75	97.99	97.90	97.02	**98.67**	97.02
6	94.47	81.22	87.86	94.23	97.96	95.63	**98.30**
7	76.68	56.31	72.21	66.07	82.22	**92.13**	88.77
8	68.79	47.29	59.32	62.24	76.02	75.42	**80.06**
9	75.26	49.55	76.81	68.35	72.35	86.48	**94.18**
10	75.51	59.47	53.79	98.43	88.66	93.82	**99.66**
11	71.08	61.74	60.52	85.80	95.35	93.61	**97.42**
12	62.76	37.87	66.89	84.25	87.36	88.23	**91.93**
13	38.58	78.38	65.53	80.67	87.47	**95.11**	84.51
14	97.83	92.56	97.33	**100.00**	99.11	**100.00**	**100.00**
15	95.25	59.76	98.26	**98.83**	98.68	97.22	95.07
OA	81.66	56.15	79.25	79.78	87.58	91.08	**92.15**
AA	82.11	61.15	81.03	82.73	89.41	92.07	**92.80**
Kappa	80.18	52.89	77.56	78.15	86.57	90.66	**91.51**

III. EXPERIMENTS

A. Datasets

In this experiment, we evaluate the proposed method with four widely used HSI datasets including Indian Pines (Indian Pines, captured over Northwest Indian, contains 16 classes and has 145 × 145 pixels with 200 spectral bands), Pavia University (PaviaU, captured over University of Pavia in Italy, contains nine classes and has 610 × 340 pixels with 103 spectral bands), Salinas (Salinas, captured over Salinas Valley in the USA, contains 16 classes and has 512 × 217 pixels with 204 spectral bands), and Houston University (Houston, captured over University of Houston campus and its neighboring areas in the USA, contains 15 classes and has 349 × 1905 pixels with 144 spectral bands). For each dataset, we randomly select 30 examples (i.e., pixels) per class for training or select 15 examples if there are less than 30 examples in the corresponding category. The remaining examples per class are used as the testing data.

B. Experimental Setup

We utilize four evaluation indices: per-class accuracy, overall accuracy (OA), average accuracy (AA), and Kappa coefficient (Kappa). To demonstrate the effectiveness of DRGCN, several state-of-the-art HSI classification methods are used for comparison, including the traditional method (SVM [15]), spatial–spectral CNN-based methods (2-D CNN [2], 3-D CNN [3]), and GCN-based methods (GCN [5], S^2GCN [6], CAD-GCN [9]). Our method is implemented by Pytorch with the Adam optimizer. After many trials, we obtain the optimal parameters as follows. Each GCN layer contains 60 hidden units, and the learning rate is $5e-4$. For the Houston dataset, the number of ResGCN and ResDGCN modules is set to 1, the dropout parameter is set to 0.7 to prevent overfitting, and the learning rate is 0.001. For the other datasets, the number of ResGCN and ResDGCN modules is set to 2. For Indian Pines, PaviaU, Salinas, and Houston datasets, α is 0.4, 0.6, 0.8, and 0.4, respectively. All these methods are run repeatedly ten times to obtain stable and reliable results on each HSI dataset.

C. Experimental Results

The classification results of different methods on four datasets, including Indian Pines, PaviaU, Salinas, and Houston, are shown in Tables I–IV, respectively. Our method achieves the best results in OA, AA, and Kappa, respectively, compared with other methods on four datasets. The reason is that DRGCN can take full advantage of the complementary and correlated information between different graph representations. Specifically, dual-branch GCN aims to extract spatial and global context-aware information from graph nodes, which can enhance the discriminative ability of graph nodes and increase the diversity of node features. The classification maps on the Salinas dataset are shown in Fig. 5. We can see that the classification map of DRGCN on the Salinas dataset is closer to the ground truth, which is smoother in homogeneous regions than other methods, which is consistent with the classification performance offered in Table III.

To study the impact of the number of training samples on HSI classification, we conduct experiments with different sizes of training sets per class ranging from 5 to 30 on four datasets, and the OAs of all methods are reported in Fig. 6. The OA of all methods improves as training size

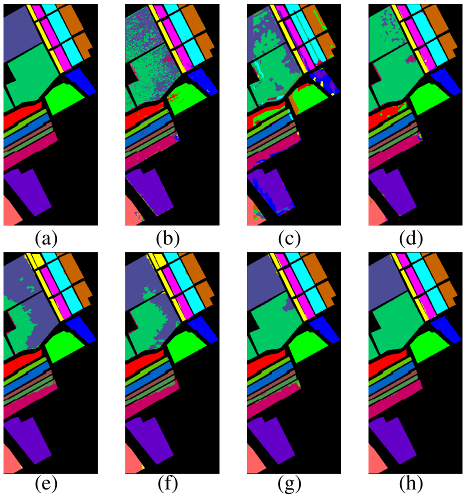
(a) (b) (c) (d)
(e) (f) (g) (h)

Fig. 5. Classification maps of different methods on Salinas dataset. (a) Ground truth. (b) SVM. (c) 2DCNN. (d) 3DCNN. (e) GCN. (f) S^{2}GCN. (g) CAD-GCN. (h) DRGCN.

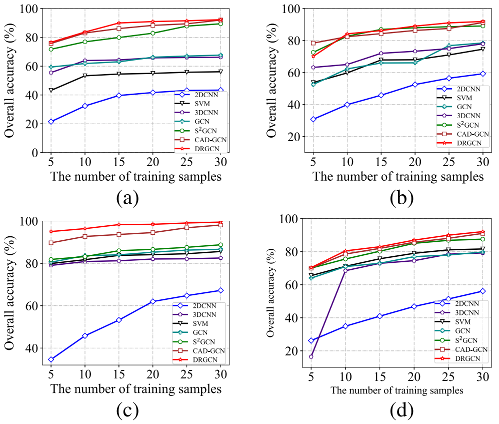

Fig. 6. OAs obtained by various methods under different numbers of training samples per class. (a) Indian Pines. (b) PaviaU. (c) Salinas. (d) Houston.

TABLE V

EFFECTIVENESS OF THE COMPONENTS IN THE PROPOSED METHOD ON THE INDIAN PINES DATASET

Methods	OA	AA	Kappa
w/o-ResGCN	91.74	95.06	90.58
w/o-ResDGCN	84.75	90.51	82.68
w/o-Residual	91.69	95.15	90.52
DRGCN	**92.21**	**95.55**	**91.11**

increases, but DRGCN achieves higher classification results than other competitors, even with a small number of training samples.

To evaluate the contributions of each module in DRGCN, an ablation study is also conducted. Various variants of DRGCN are designed, called w/o-ResGCN, w/o-ResDGCN, and w/o-Residual, representing without the ResGCN, ResDGCN, and residual learning, respectively. The OA of these variants and DRGCN on the Indian Pines dataset are shown in Table V. We can observe that the lack of any component will inevitably reduce the OA value.

IV. Conclusion

In this letter, a DRGCN method is proposed for HSI classification by extracting spatial and global context-aware information from graph nodes with a dual GCN module, which considers the complementary and correlated information between different graph representations. Unlike prior works that heavily rely on graph embedding to refine the graph, the graph in DRGCN is refined by the previous layer's graph embedding and global representation, extracting robust graph representations containing abundant relation information between graph nodes. In addition, we also introduce residual learning to alleviate the over-smoothing problem. The experiments on four general HSI datasets show that DRGCN achieves better results compared with other state-of-the-art methods.

References

[1] Z. Wu and Z. Yan, "Selection of optimal bands for hyperspectral local feature descriptor," *IEEE Geosci. Remote Sens. Lett.*, vol. 19, pp. 1–5, 2022.

[2] 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, Jul. 2016.

[3] 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.

[4] L. Mou, S. Saha, Y. Hua, F. Bovolo, L. Bruzzone, and X. X. Zhu, "Deep reinforcement learning for band selection in hyperspectral image classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 60, pp. 1–14, 2022.

[5] T. N. Kipf and M. Welling, "Semi-supervised classification with graph convolutional networks," in *Proc. 5th Int. Conf. Learn. Represent.*, Apr. 2017, pp. 1–14.

[6] 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. 2018.

[7] A. Sha, B. Wang, X. Wu, and L. Zhang, "Semisupervised classification for hyperspectral images using graph attention networks," *IEEE Geosci. Remote Sens. Lett.*, vol. 18, no. 1, pp. 157–161, Jan. 2021.

[8] X. Tong, J. Yin, B. Han, and H. Qv, "Few-shot learning with attention-weighted graph convolutional networks for hyperspectral image classification," in *Proc. IEEE Int. Conf. Image Process. (ICIP)*, Oct. 2020, pp. 1686–1690.

[9] S. Wan, C. Gong, P. Zhong, S. Pan, G. Li, and J. Yang, "Hyperspectral image classification with context-aware dynamic graph convolutional network," *IEEE Trans. Geosci. Remote Sens.*, vol. 59, no. 1, pp. 597–612, Jan. 2021.

[10] 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.

[11] R. Achanta, A. Shaji, K. Smith, A. Lucchi, P. Fua, and S. Süsstrunk, "SLIC superpixels compared to state-of-the-art superpixel methods," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 34, no. 11, pp. 2274–2282, Nov. 2012.

[12] J. Chen, B. Lei, Q. Song, H. Ying, D. Z. Chen, and J. Wu, "A hierarchical graph network for 3D object detection on point clouds," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 392–401.

[13] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2016, pp. 770–778.

[14] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," in *Proc. Int. Conf. Mach. Learn.*, Jul. 2015, pp. 448–456.

[15] B. Waske, S. van der Linden, J. Benediktsson, A. Rabe, and P. Hostert, "Sensitivity of support vector machines to random feature selection in classification of hyperspectral data," *IEEE Trans. Geosci. Remote Sens.*, vol. 48, no. 7, pp. 2880–2889, Jul. 2010.