A Greedy Strategy Guided Graph Self-Attention Network for Few-Shot Hyperspectral Image Classification

Fei Zhu, Cuiping Shi, *Member, IEEE*, Liguo Wang, *Member, IEEE*, and Kaijie Shi

***Abstract*— For hyperspectral image classification (HSIC), labeling samples is challenging and expensive due to high dimensionality and massive data, which limits the accuracy and stability of classification. To alleviate this problem, a greedy strategy guided graph self-attention network (GS-GraphSAT) is proposed. First, a graph self-attention (GSA) mechanism is designed by combining a multihead self-attention (MHSA) mechanism with the graph attention network (GAT), which can simultaneously consider the direct and indirect relationships between nodes and deeply analyze the intrinsic characteristics of nodes. Second, a multiattention fusion (MAF) module is developed, which utilizes multiscale convolution kernels and attention mechanisms to significantly enhance the network's ability to extract local features from images at the pixel level, thereby further enriching the hierarchy and diversity of features. Finally, a greedy training strategy (GTS) is proposed. During the training process, GTS accurately determines the optimal time to supplement samples by analyzing the changes in losses, thereby achieving a significant improvement in network classification performance with limited samples. Extensive experiments were conducted on four challenging datasets. The results demonstrate that the proposed method significantly outperforms other state-of-the-art methods in terms of classification accuracy and robustness. The performance improvement of overall accuracy (OA) can reach up to 1.70% in Houston 2013 (HT). The codes of this work will be available at https://github.com/Iseemax/IEEE_TGRS_GS-GraphSAT for reproduction.**

***Index Terms*— Convolutional neural network (CNN), few-shot learning, graph attention network (GAT), greedy training strategy (GTS), hyperspectral image classification (HSIC), self-attention.**

I. Introduction

HYPERSPECTRAL images (HSIs) integrate imaging technology and spectral detection technology so that each sample not only carries the spatial characteristics of the target but also contains dozens to hundreds of continuous and subdivided spectral information, thereby achieving precise characterization of land cover [1]. Due to its characteristic of "spectral and spatial integration," HSI is widely used in medical diagnosis [2], [3], mineral exploration [4], environmental monitoring [5], military reconnaissance [6], and other fields. Research on HSI classification (HSIC) methods has, therefore, attracted much attention in the field of remote sensing [7].

Early HSIC methods primarily focused on extracting feature information from spectral data. Notable methods include logistic regression [8], random forest [9], support vector machines [10], and sparse representation classification [11], [12]. However, these methods exhibit certain limitations, primarily due to an inadequate exploration of spatial features and an excessive reliance on the prior knowledge of experts.

In recent years, deep learning, with its powerful feature learning capabilities, has made significant breakthroughs in multiple fields, which has also attracted the attention of researchers in the remote sensing field [13], [14], [15], [16]. To address the semantic segmentation task in cross-city scenarios, Hong et al. [17] proposed a high-resolution domain adaptation network (HighDAN). HighDAN can preserve spatial topological structures and reduce the domain gap between remote sensing images from different cities through adversarial learning. In HSIC, research achievements based on convolutional neural networks (CNNs) [18], Transformers [19], [20], and graph neural networks (GNNs) [21], [22], [23] have been the most remarkable. HSIs contain rich spectral information and continuous spatial distribution of land cover. Three-dimensional convolution kernels can simultaneously consider both spatial and spectral information of the image, thereby more comprehensively extracting features of HSIs. Zhong et al. [24] proposed an end-to-end spectral–spatial residual network (SSRN). SSRN introduces 3-D convolution kernels on the basis of ResNet [25], thereby realizing the joint extraction of spectral and spatial features. However, 3-D convolutional kernels are computationally expensive and inefficient. Hence, Roy et al. [26] proposed a hybrid spectral convolutional network (HybridSN). HybridSN leverages the strengths of both 3-D and 2-D convolutional kernels, enabling efficient and effective feature extraction in both spatial and spectral dimensions. To alleviate the problem of insufficient labeled data in HSI, Yao et al. [27] proposed a semiactive CNN (SA-CNN). By combining active learning and superpixel segmentation, SA-CNNs can effectively select informative

Received 16 September 2024; revised 16 October 2024, 26 October 2024, and 9 November 2024; accepted 21 November 2024. Date of publication 25 November 2024; date of current version 5 December 2024. This work was supported in part by the National Natural Science Foundation of China under Grant 42271409 and in part by the Fundamental Research Funds in Heilongjiang Provincial Universities under Grant 145109145. *(Corresponding author: Cuiping Shi.)*

Fei Zhu and Kaijie Shi are with the Department of Communication Engineering, Qiqihar University, Qiqihar 161000, China (e-mail: 2022935750@qqhru.edu.cn; 2022910313@qqhru.edu.cn).

Cuiping Shi is with the College of Information Engineering, Huzhou University, Huzhou 313000, China (e-mail: shicuiping@zjhu.edu.cn).

Liguo Wang is with the College of Information and Communication Engineering, Dalian Nationalities University, Dalian 116000, China (e-mail: wangliguo@hrbeu.edu.cn).

Digital Object Identifier 10.1109/TGRS.2024.3505539

samples and generate pseudolabels for unlabeled data, thus improving the model's performance.

Attention mechanism is a computational mechanism that mimics human attention by selectively focusing on informative regions of the input [28], [29], [30], [31], [32], [33]. Cui et al. [34] proposed a dual-triple attention network (DTAN) for HSIC with limited training samples. DTAN utilizes a dual attention mechanism to capture the interactions between spatial and spectral information. Liang et al. [35] proposed a multiscale spectral–spatial attention network (MOCNN). MOCNN employs multiscale 2-D octave convolution and 3-D DenseNet [36] to extract spatial and spectral features, respectively, and enhances network performance through an attention mechanism. Shi et al. [37] proposed an expansion convolution network (ECNet). ECNet utilizes a similar feedback block to enhance feature representation and leverages an attention mechanism to distinguish the importance of different features. Bai et al. [38] proposed a method based on adaptive subspaces classifier and feature transformation (SSFT). SSFT alleviates the problem of limited samples through local channel attention (CA) and feature transformation modules. Wu et al. [39] proposed a new framework called cross-channel reconstruction (CCR) Net for multimodal remote sensing data classification. This framework is based on CNNs and introduces an advanced CCR module to achieve effective fusion of data from different sources. Chen et al. [40] proposed an end-to-end grid network (GNet) that effectively identifies discriminative features by balancing the extraction of spectral and spatial features.

While CNNs have been widely used in HSI, their performance is limited by the receptive field of convolutional kernels. Transformers, on the other hand, have shown great potential in capturing long-range dependencies in sequential data, as demonstrated by their success in natural language processing (NLP). Researchers have designed a series of Transformer variants for visual tasks, such as vision transformer (ViT) and Swin Transformer [41], [42], [43], to address the characteristics of image data. These models effectively capture global features in images by dividing images into multiple patches and feeding these patches as sequences to the Transformer. Hong et al. [44] proposed a SpectralFormer (SF) for HSIC. SF captures local details by learning the relationships between adjacent bands within a group and employs cross-layer skip connections to propagate memory information. Sun et al. [45] proposed a spectral–spatial feature tokenization Transformer (SSFTT). SSFTT utilizes convolutional layers to extract low-level features, which are then transformed into semantic tokens and modeled by a Transformer for high-level semantic features. Shi et al. [46] proposed a dual-branch multiscale transformer network (DBMST). DBMST integrates multiscale feature extraction and Transformer-based global feature extraction, achieving better classification performance.

Recently, GNNs have proven to be a promising framework in the research of non-Euclidean dependencies in HSIs [47], [48], [49]. In these methods, each pixel and its local neighborhood are represented as a node in a graph, and the edges between nodes are determined by spatial and spectral relationships. Liu et al. [50] proposed a CNN-enhanced graph convolutional network (CEGCN). CEGCN leverages CNNs to extract local pixel-level features, which are then fused with the features of large-scale irregular regions modeled by a graph convolutional network (GCN). A fast dynamic GCN and CNN parallel network (FDGC) has been proposed [51]. FDGC combines dynamic GCNs and CNNs, using multiple branches to extract different types of features in parallel. Ding et al. [52] proposed a semisupervised locality-preserving dense GNN with autoregressive moving average (ARMA) filters and context-aware learning (DARMA-CAL). This method employs ARMA filters for graph convolution operations, enabling it to better capture global structural information. Zhou et al. [53] proposed an attention multihop graph and multiscale convolutional fusion network (AMGCFN). In addition to local information extracted by CNNs, AMGCFN aggregates multihop contextual information by applying multihop graphs at different levels. However, GCNs have limitations in handling directed graphs. To address this issue, Dong et al. [54] proposed a weighted feature fusion of CNN and graph attention network (WFCG). WFCG combines the strengths of CNNs and graph attention networks (GATs), making the model completely independent of the graph structure. Shi et al. [55] proposed a CNN and enhanced-GAT fusion network (CEGAT). CEGAT captures important node features and suppresses redundant features by calculating attention coefficients between node vectors. Furthermore, the low representation capability of the original HSI limits the accuracy of superpixel segmentation. To address this issue, Chen et al. [56] proposed a local aggregation and global attention network (LAGAN). LAGAN designs a spectral-induced alignment superpixel segmentation strategy that can simultaneously utilize both original and deep abstract spectral features for superpixel segmentation, resulting in more accurate pixel-to-region assignments.

Although the above methods have achieved effective classification performance, there are still some problems.

1) Sample limitation is a long-standing issue in HSIC. The high dimensionality and large volume of HSI data make it extremely costly to label, while deep learning models require a massive amount of labeled data for training, leading to a shortage of training samples.

2) The attention mechanism employed in GAT is inherently localized, focusing solely on direct neighbors. In HSIC tasks, this limitation limits the model's ability to utilize the rich and potential long-term dependencies in the data, thereby hindering the improvement of classification performance.

To alleviate the above problems, a greedy strategy guided graph self-attention network (GS-GraphSAT) is proposed. To enhance information propagation among nodes, a graph self-attention (GSA) mechanism is designed. GSA effectively combines the advantages of multihead self-attention (MHSA) and GAT, not only fully capturing the relationships between directly connected nodes but also conducting an in-depth exploration of the connections between non-directly connected nodes with potential correlations. To fully capture pixel-level local features, a multiattention fusion (MAF) module is constructed. MAF applies convolutional kernels of different sizes

at multiple scales to extract local features of the image and fuses the weighted features of multiple attention mechanisms, thereby more comprehensively capturing and enhancing the local feature information of the image at the pixel level. To alleviate the problem of limited samples in HSI, a greedy training strategy (GTS) is proposed. GTS monitors the changes in training loss to determine when to add new samples, thereby effectively improving the classification accuracy of the model even when the number of samples is limited. Extensive experiments on four public datasets indicate that the proposed GS-GraphSAT outperforms some state-of-the-art methods.

The main contributions of this article are given as follows.

1) A novel GSA mechanism is introduced to facilitate information propagation within GNNs. GSA leverages the strengths of MHSA and GAT, allowing for the modeling of both local and nonlocal dependencies among nodes.

2) A novel MAF module is introduced to enhance the extraction of pixel-level local features. MAF leverages a multiscale convolutional architecture combined with multiple attention mechanisms to effectively capture and fuse local features from different receptive fields.

3) To mitigate the issue of limited samples in HSI, a novel GTS is proposed. GTS employs a dynamic sample selection approach based on training loss, enabling the model to learn more effectively from a small dataset.

The remaining part of this article is organized as follows. Section II introduces the proposed method in detail. Section III first describes the dataset and the parameter settings of the experiment, followed by experimental verification of the proposed method. Section IV presents the conclusion and future work.

II. Methodology

We denote HSIs as $\mathbf{X} \in \mathbb{R}^{H\times W\times B}$, where H, W, and B, respectively, represent the height, width, and spectral band number of HSI. Partition the original dataset $\mathbf{X}$ into three subsets: training set $\mathbf{X}_{\text{tr}} \in \mathbb{R}^{H\times W\times B}$, validation set $\mathbf{X}_{\text{va}} \in \mathbb{R}^{H\times W\times B}$, and test set $\mathbf{X}_{\text{te}} \in \mathbb{R}^{H\times W\times B}$.

A. Overall Structure

The overall structure of the proposed GS-GraphSAT is shown in Fig. 1. The network includes five parts: data preprocessing, superpixel-based GSA branch, pixel-based MAF branch, fusion classification, and GTS.

The initial step involves data preprocessing. Graph nodes are constructed from the input HSIs data $\mathbf{X}$. To reduce computational complexity, principal component analysis (PCA) [57] is employed to extract the most informative bands of the image. Subsequently, the dimensionality-reduced data are segmented into superpixels using the simple linear iterative clustering (SLIC) algorithm [58]. Similar and adjacent pixels are clustered into superpixels, forming a relationship matrix $\mathbf{G}$ and an adjacency matrix $\mathbf{H}$, where N denotes the number of superpixels. The construction of these matrices is detailed in (1) and (2)

$$\mathbf{G}_{i,j} = \begin{cases} 1, & \text{if } \bar{X}_i \in S_j \\ 0, & \text{otherwise} \end{cases} \qquad \bar{X} = \text{Flatten}(X) \tag{1}$$

where $\mathbf{G}_{i,j}$ denotes the value at (i, j) in the relationship matrix, reflecting the mapping between pixels and superpixels. The matrix's rows represent pixels, and its columns represent superpixels. Flatten$(\cdot)$ denotes the flattening operation. $\bar{X}$ denotes the 1-D vector resulting from flattening the spatial dimension, and S denotes superpixels, which are used to construct a relationship matrix in conjunction with $\bar{X}$

$$\mathbf{H}_{i,j} = \begin{cases} 1, & \text{if } S_i \text{ and } S_j \text{ are adjancent} \\ 0, & \text{otherwise} \end{cases} \tag{2}$$

where $\mathbf{H}_{i,j}$ denotes the value of the adjacency matrix at (i, j), reflecting the connectivity between superpixel nodes. Both the rows and columns of the matrix denote superpixels, with the elements indicating whether there is a connection between the respective nodes. The adjacency matrix facilitates the quick search of connection relationships between nodes.

The pixel inputs to the MAF branch undergo a two-step processing. First, a pointwise convolution is employed to reduce the dimensionality of the spectral data. Second, another pointwise convolution is utilized to refine the data, introducing nonlinearity to the reduced-dimensional features. Finally, the output $\mathbf{X}_{\text{cnn}} \in \mathbb{R}^{H\times W\times L}$ is obtained, where L denotes the length of the processed spectral sequence. This process can be expressed as

$$\mathbf{X}_{\text{cnn}} = \text{PW}(\text{PW}(\mathbf{X})) \tag{3}$$

where PW$(\cdot)$ represents a pointwise convolution block, including batch regularization, convolution, and rectified linear unit (ReLU) activation function.

The process of pointwise convolution can be expressed as

$$\mathbf{X}_{\text{out}} = F_\tau(F_{\text{BN}}(\mathbf{X}) * \mathbf{w} + \mathbf{b})) \tag{4}$$

where $F_{\text{BN}}(\cdot)$ represents a batch normalization layer, $F_\tau(\cdot)$ represents the ReLU activation function, $*$ represents the convolution operator, $\mathbf{w}$ represents the weights of the convolutional kernel, and $\mathbf{b}$ represents the bias.

The second part is a GSA branch based on superpixels. At the beginning of this stage, pixel data need to be transformed into graph node data. This process can be represented as

$$\mathbf{V} = \text{Encode}(\mathbf{X}; \mathbf{G}) = \hat{\mathbf{G}}^{\text{T}} \cdot \text{Flatten}(\mathbf{X}) \tag{5}$$

where $\mathbf{V}$ denotes the set of graph nodes composed of superpixels. $\hat{\mathbf{G}}^{\text{T}}$ denotes the relationship matrix after column regularization and transposition; the Encode$(\cdot)$ converts pixel data $\mathbf{X}$ into graph node data, resulting in the transformed graph representation $\mathcal{G} \in \mathbb{R}^{N\times L}$. Subsequently, the graph nodes undergo multiple processes through the GSA layer to yield a weighted graph. Finally, this weighted graph is transformed into pixel data. This process can be represented as

$$\mathbf{X}' = \text{Decode}(\mathbf{V}; \mathbf{G}) = \text{Reshape}(\mathbf{G} \cdot \mathbf{V}) \tag{6}$$

where Decode$(\cdot)$ is used to decode the node data into pixel data, while Reshape$(\cdot)$ is utilized to restore the spatial dimensions of flattened data.

The third part is a pixel-based MAF branch. Through the proposed MAF, the fine-grained features lacking in the

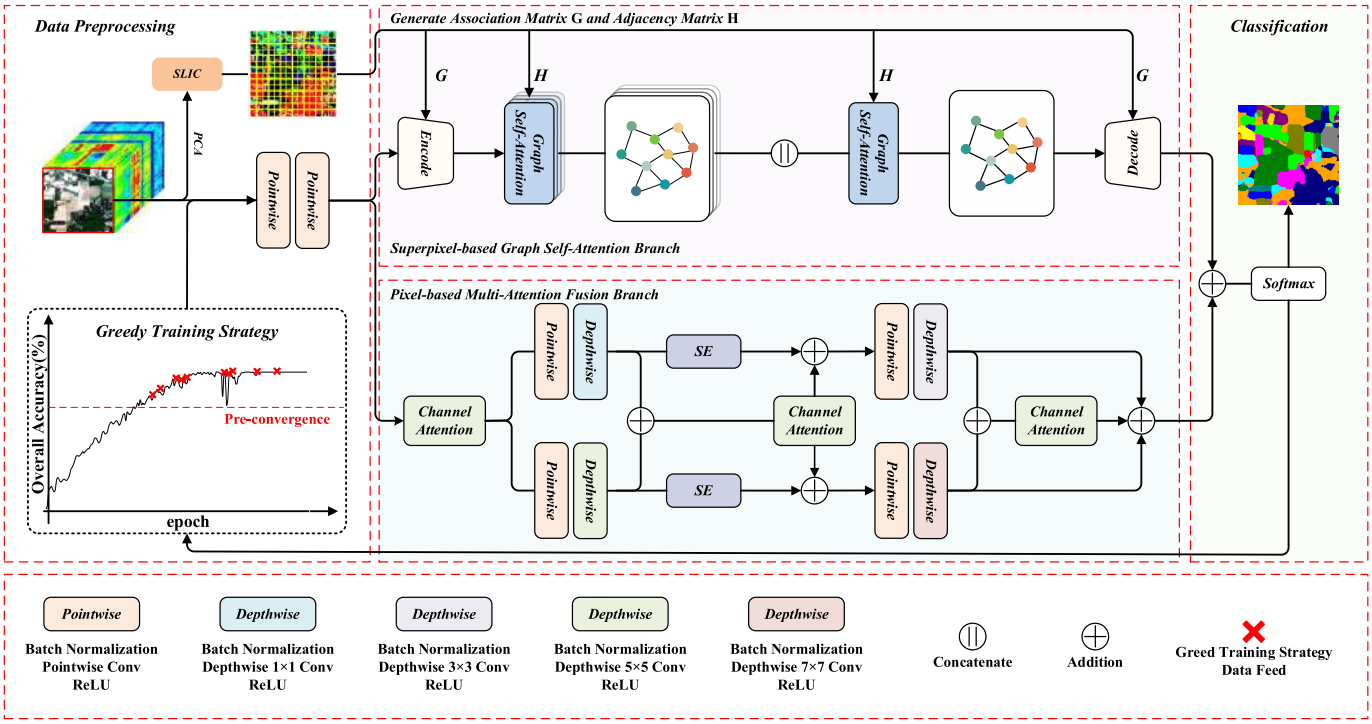


Fig. 1. Overall structure of the proposed GS-GraphSAT. The GSA branch is employed to capture coarse-grained local features, while the MAF branch is designed to extract fine-grained local features. The GTS alleviates the sample limitation problem by improving sample utilization. The detailed structures of GSA and MAF will be unfolded in Figs. 2 and 3.

graph branch are supplemented by processing with multiscale convolution kernels and attention mechanisms. The fourth part is the fusion classification. In this stage, the features processed by the graph branch and the convolution branch are weighted and fused, and the result is input into a softmax classifier to complete the classification task. This process can be represented as

$$\mathbf{X}_{\text{out}} = \alpha \cdot \mathbf{X}' + (1-\alpha) \cdot \mathbf{X}'' \tag{7}$$

$$\text{Soft max}((\mathbf{X}_{\text{out}})_i) = \frac{\exp((\mathbf{X}_{\text{out}})_i)}{\sum_j \exp\left((\mathbf{X}_{\text{out}})_j\right)} \tag{8}$$

where $\mathbf{X}'$ and $\mathbf{X}''$ represent the processed features by the graph branch and the convolution branch, respectively. $\mathbf{X}_{\text{out}}$ represents the result of the fusion of the two branches. α represents the weight coefficient of the adaptive change.

The final part is the GTS. This stage runs throughout the entire training process, enabling the network to adaptively determine the optimal time to supplement samples to the training set, thereby more effectively utilizing sample resources and optimizing training effectiveness.

B. GSA Mechanism

GAT can handle graphs with arbitrary structures, but they primarily focus on neighboring nodes, capturing local dependencies. This means that for non-directly connected nodes with potential correlations, the propagation of information between them is limited. To alleviate this problem, a novel GSA mechanism is proposed, and its detailed structure is shown in Fig. 2.

Specifically, the self-attention part initiates by applying a mapping operation to the graph $\mathcal{G}$, generating three multihead graph representations denoted as $\mathcal{G}_q, \mathcal{G}_k, \mathcal{G}_v \in \mathbb{R}^{h\times N\times d}$. Here, $L = h \times d$, h represents the number of heads, and d represents the length of the sequence after multihead segmentation.

Subsequently, a dot product is performed between $\mathcal{G}_q$ and $\mathcal{G}_k^{\text{T}}$. The results are scaled by $1/(d_k)^{1/2}$ and normalized using a softmax function to produce the attention scores. The resulting attention map is denoted as $\mathbf{M} \in \mathbb{R}^{h\times N\times N}$.

Following this, the attention map $\mathbf{M}$ is multiplied elementwise with $\mathcal{G}_v$ to produce the weighted node features for a single attention head, denoted as $\mathcal{G}_{\text{SA}} \in \mathbb{R}^{h\times N\times d}$. This process is replicated for multiple attention heads in parallel, and the resulting features are concatenated to form the final representation $\mathcal{G}' \in \mathbb{R}^{N\times L}$.

This process can be represented as

$$\mathcal{G}_{\text{SA}} = \text{Attention}\left(\mathcal{G}_q, \mathcal{G}_k, \mathcal{G}_v\right) = \text{Soft max}\left(\frac{\mathcal{G}_q \cdot \mathcal{G}_k^{\text{T}}}{\sqrt{d_k}}\right) \cdot \mathcal{G}_v \tag{9}$$

$$\mathcal{G}' = \text{Multihead}\left(\mathcal{G}_q, \mathcal{G}_k, \mathcal{G}_v\right) = ||_i^h \mathcal{G}_{\text{SA}_i} \cdot \mathbf{W} \tag{10}$$

where d_k represents the dimension of $\mathcal{G}_k$, $||(\cdot)$ represents concatenate operations, and $\mathbf{W}$ represents the parameter matrix. Moreover, it is necessary to perform the accumulation operation on the attention map $\mathbf{M}$ to obtain the attention map $\mathbf{M}' \in \mathbb{R}^{N\times N}$ for the utilization of the graph attention part. This process can be represented as

$$\mathbf{M}' = \sum_{i=1}^{h} \mathbf{M}_i. \tag{11}$$

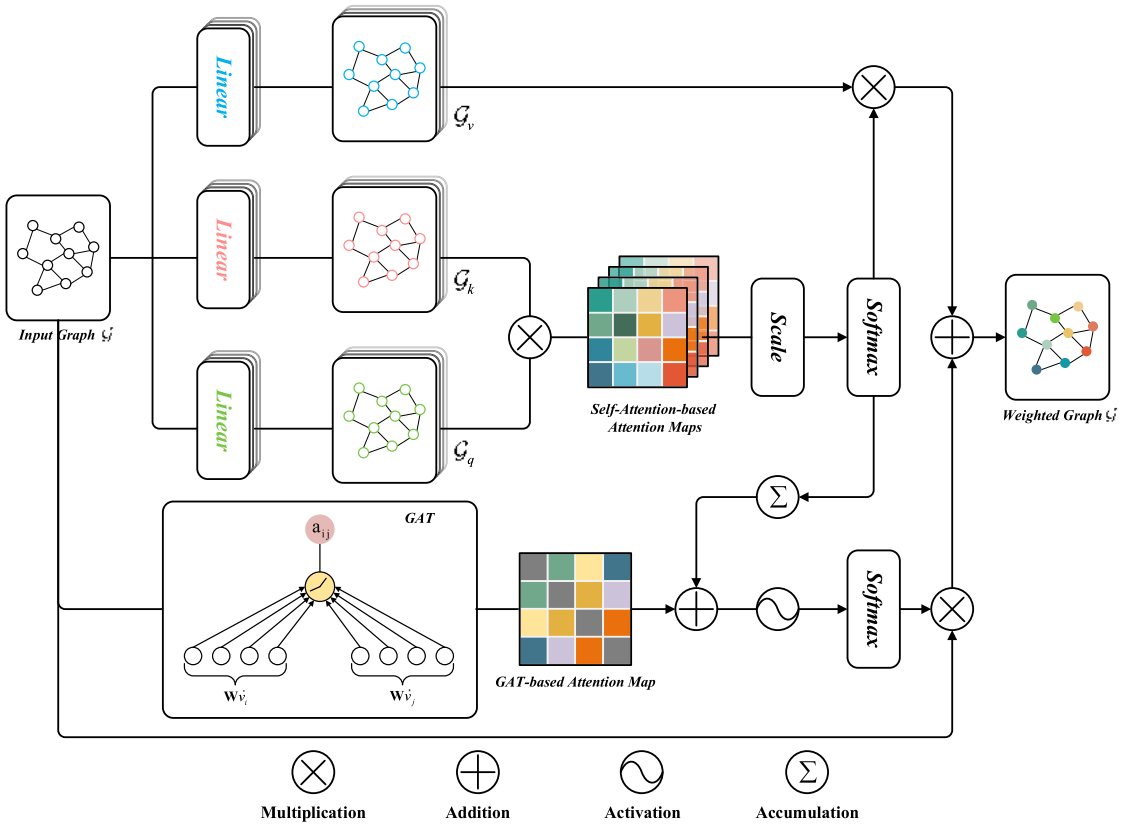


Fig. 2. Structure of GSA ("Activation" represents utilizing adjacency matrix to activate attention maps, and "Accumulation" represents accumulating attention maps from MHSA).

Moving on to the graph attention part. First, a linear transformation is applied to the nodes set $\mathbf{V} = \{\vec{v}_1, \vec{v}_2, \ldots, \vec{v}_n\}$. Subsequently, we compute the attention coefficients using a shared attention mechanism. This process can be represented as

$$e_{i,j} = A\left(\mathbf{W}\vec{v}_i, \mathbf{W}\vec{v}_j\right) \tag{12}$$

where $e_{i,j}$ denotes the attention coefficient, which can denote the correlation between node i and node j. $\mathbf{W}$ represents the parameter matrix, and $A(\cdot)$ represents the shared attention mechanism. Afterward, the softmax function is utilized to normalize the attention coefficients into scores. This process can be expressed as

$$\mathbf{M}'' = \text{Soft}\max\left(e_{i,j}\right) = \frac{\exp\left(e_{i,j}\right)}{\sum_{k\in N_i}\exp\left(e_{i,k}\right)}. \tag{13}$$

Then, the attention maps $\mathbf{M}''$ and $\mathbf{M}'$ obtained from the self-attention part are combined through elementwise addition. The fused result is then modulated by the adjacency matrix $\mathbf{H}$ to obtain the final attention map, denoted as $\mathbf{M}''' \in \mathbb{R}^{N\times N}$.

Specifically, for node positions with connections, retain the fused results; otherwise, set the value of that position to zero. This process can be represented as

$$\mathbf{M}''' = \begin{cases} \mathbf{M}'_{i,j} + \mathbf{M}''_{i,j}, & \text{If there is a connection at } \mathbf{H}_{i,j} \\ 0, & \text{otherwise.} \end{cases} \tag{14}$$

Next, $\mathbf{M}'''$ is multiplied with the original graph $\mathcal{G}$ to obtain the weighted graph of the graph attention part, denoted as $\mathcal{G}'' \in \mathbb{R}^{N\times L}$. Finally, the weighted feature graphs $\mathcal{G}'$ and $\mathcal{G}''$ of the two attention parts are fused to obtain the final graph, denoted as $\mathcal{G}_{\text{out}} \in \mathbb{R}^{N\times L}$. This process can be represented as

$$\begin{cases} \mathcal{G}'' = \mathbf{M}''' \cdot \mathcal{G} \\ \mathcal{G}_{\text{out}} = \mathcal{G}' + \mathcal{G}''. \end{cases} \tag{15}$$

To ensure a stable fusion of the weighted feature maps from MHSA and GAT, a multihead mechanism is applied to the first GSA layer. Specifically, the first GSA layer is executed independently K times, and the results are then concatenated. In the second GSA layer, a single-layer structure was adopted. The final graph was decoded by the $Decode(\cdot)$ function and input into the classifier. This process can be expressed as

$$\mathcal{G} = ||_{k=1}^{K}\sigma_r\left(\mathcal{G}_{\text{out}_k}\right). \tag{16}$$

By combining the multihead attention mechanism of MHSA and the graph attention mechanism of GAT, GSA comprehensively obtains the correlation information between nodes and strengthens the original attention graph, effectively improving the classification performance of the model.

C. MAF Module

The local features extracted from graph nodes constructed based on superpixels mainly reflect the coarse-grained information of the image. Without the supplementation of

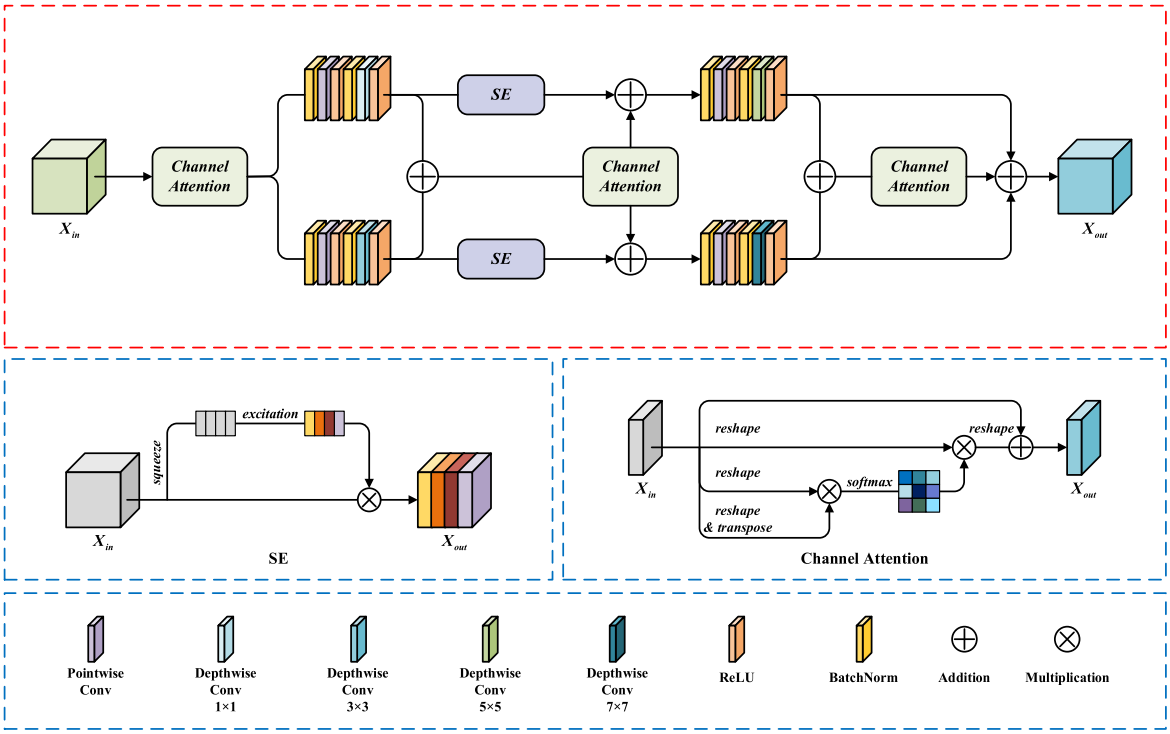


Fig. 3. Structure of MAF, with CA in the lower right and SE attention in the lower left.

fine-grained information, the classification performance of the model will be significantly affected. To address the above problem, an MAF module is proposed, as shown in Fig. 3.

First, the module utilizes CA to weight the spectral dimension of pixel inputs, enabling the network to focus on discriminative features. This process can be expressed as

$$M_{ij} = \frac{\exp\left(X_i \cdot X_j\right)}{\sum_{i=1}^{N} \exp\left(X_i \cdot X_j\right)} \tag{17}$$

$$X'_j = \eta \cdot \sum_{i=1}^{N} \left(M_{ij} \cdot X_j\right) + X_j \tag{18}$$

where X'_j represents the weighted data processed by CA and η represents a learnable coefficient preset to 0.

Following this, the weighted data undergo processing by two convolutional branches with varying receptive fields. Pointwise convolution is adopted to fuse information across channels, facilitating interchannel communication. Subsequently, depthwise convolution is utilized to capture spatial features at multiple scales, with each channel being processed independently to extract distinct and salient features. This process can be expressed as

$$\begin{cases} \mathbf{X}_1 = \mathrm{DW}_{1\times1}(\mathrm{PW}(\mathrm{CA}(\mathbf{X}_{\mathrm{cnn}}))) \\ \mathbf{X}_2 = \mathrm{DW}_{5\times5}(\mathrm{PW}(\mathrm{CA}(\mathbf{X}_{\mathrm{cnn}}))) \end{cases} \tag{19}$$

where $\mathrm{DW}_{1\times1}(\cdot)$ represents the depthwise convolution block with a kernel size set to 1. $\mathrm{DW}_{5\times5}(\cdot)$ represents the depthwise convolution block with a kernel size set to 5. $\mathrm{CA}(\cdot)$ represents the CA.

Subsequently, the outputs $\mathbf{X}_1$ and $\mathbf{X}_2$ from the two branches are further weighted using the squeeze-and-excitation network (SE) module shown in Fig. 3. Meanwhile, $\mathbf{X}_1$ and $\mathbf{X}_2$ are fused and reweighted by the CA module. Both SE and CA modules assign weights but focus on different features. The result of the second CA weighting is used as shared weights and fused with the features from both branches after SE processing. This process enables the network to more accurately capture local image features and distinguish features of varying importance. This process can be expressed as

$$\mathrm{SE}(\mathbf{X}) = \mathrm{Linear}(F_\tau(\mathrm{Linear}(\mathrm{Avg}(\mathbf{X})))) \cdot \mathbf{X} \tag{20}$$

$$\begin{cases} \mathbf{X}_3 = \mathrm{SE}(\mathbf{X}_1) + \mathrm{CA}(\mathbf{X}_1 + \mathbf{X}_2) \\ \mathbf{X}_4 = \mathrm{SE}(\mathbf{X}_2) + \mathrm{CA}(\mathbf{X}_1 + \mathbf{X}_2) \end{cases} \tag{21}$$

where $\mathrm{Avg}(\cdot)$ represents average pooling, $\mathrm{Linear}(\cdot)$ represents linear mapping, and $\mathrm{Sigmoid}(\cdot)$ represents activation function. Next, pointwise convolutions are utilized again to integrate information from different channels in $\mathbf{X}_3$ and $\mathbf{X}_4$, and large-sized depthwise convolution blocks are employed to extract local feature information, resulting in the outputs $\mathbf{X}_5$ and $\mathbf{X}_6$.

Compared to the previous convolutional blocks, these two convolutional blocks can capture more local details, further enhancing the representational capacity of features. Subsequently, $\mathbf{X}_3$ and $\mathbf{X}_4$ are fused and fed into the CA module for a third weighting to boost the weights of key features. Finally, the weighted output is fused with $\mathbf{X}_5$ and $\mathbf{X}_6$ to obtain the final feature representation. This process can be expressed as

$$\begin{cases} \mathbf{X}_5 = \mathrm{DW}_{3\times3}(\mathrm{PW}(\mathbf{X}_3)) \\ \mathbf{X}_6 = \mathrm{DW}_{7\times7}(\mathrm{PW}(\mathbf{X}_4)) \\ \mathbf{X}_{\mathrm{out}} = \mathbf{X}_5 + \mathbf{X}_6 + \mathrm{CA}(\mathbf{X}_5 + \mathbf{X}_6) \end{cases} \tag{22}$$

Algorithm 1 Implementation Process of GTS

Input: the training set $\mathbf{X}_{\text{tr}} \in \mathbb{R}^{H\times W\times B}$, validation set $\mathbf{X}_{\text{va}} \in \mathbb{R}^{H\times W\times B}$, landcover labels $\mathbf{Y} \in \mathbb{R}^{H\times W}$
Output: the optimal network model.
1. Set the training epochs E, pre-convergence accuracy OA', minimum loss $Loss_{min}$, loss list L.
2. Set the count of consecutive increases in losses required to trigger GTS N'.
3. Set the count of current consecutive increases in losses$N$.
4. Set the total number of supplementary samples Num.
5. Set the number of sample supplements T', and the current number of supplements T.
6. Initialize the network.
7. for i to E do
8. Training the Network;
9. Perform validation inference on the current trained model;
10. If $L_i \leq Loss_{min}$
Save the model of that epoch and update the $Loss_{min}$.
11. If the current validation accuracy is greater than OA' and the $T \leq T'$
Mark that the current network is in a "pre-convergence" state.
Otherwise, set N to 0 and perform step 8.
12. If the $L_{i-1} \leq L_i$
Update the N;
Update the $L_i = L_{i-1}$;
Otherwise, perform step 8.
13. If N is equal to N'
Select the maximum loss sample for a batch from $\mathbf{X}_{\text{va}}$, with a sample size of Num/T';
Supplement the batch of samples to the $\mathbf{X}_{\text{tr}}$;
Update the T;
Otherwise, perform step 8.
14. end for

where $DW_{3\times3}(\cdot)$ represents the depthwise convolution block with a kernel size set to 3; $DW_{7\times7}(\cdot)$ represents the depthwise convolution block with a kernel size set to 7.

By integrating a multiscale convolutional architecture with multiple attention mechanisms, MAF effectively extracts local features from diverse receptive fields, thus compensating for the graph branch's limited ability to capture fine-grained information.

D. Greedy Training Strategy

Limited sample size poses a significant challenge in HSIC. Due to the scarcity of available labeled samples, the training process of classifiers often fails to obtain sufficient data support, resulting in suboptimal performance, particularly in terms of accuracy and robustness. Therefore, optimizing the utilization of limited samples has become a critical research direction in this field.

To address the above problem, a GTS is proposed in this article. The detailed workflow of GTS is shown in Algorithm 1. During the data preparation process, the dataset is generally divided into the training set $\mathbf{X}_{\text{tr}}$, the validation set $\mathbf{X}_{\text{va}}$, and the test set $\mathbf{X}_{\text{te}}$. The training set $\mathbf{X}_{\text{tr}}$ is used to train the model; the validation set $\mathbf{X}_{\text{va}}$ is used to evaluate the performance of the model and help determine whether the model is overfitting; the test set $\mathbf{X}_{\text{te}}$ is used to evaluate the final performance of the model. Unlike the previous approach of setting the number of samples in the training and validation sets to be equal, GTS further subdivides the training set. In the initial stage of training, only a part of the samples in the training set are used for network training, while the remaining number of samples is temporarily merged with the validation set. As the training progresses, these numbers of samples will be remerged into the training set in batches at appropriate times to achieve efficient utilization of sample resources and continuous improvement of model performance. The validation set plays an important role in evaluating model performance. Although not directly involved in training, it can help the model filter out the sample with the highest loss. These samples often contain feature information that the model has not yet understood. If the model can successfully understand the feature information, its ability to interpret features will be significantly improved, thereby optimizing the overall performance of the model.

During the training process of the model, when the network reaches an overall accuracy (OA) of 60% or above for the first time, it is considered to have reached a "pre-convergence" state, indicating that the model has initially acquired the ability to understand features and process information. At this moment, GTS begins to analyze the loss changes during the model training process to accurately determine when to reintegrate the remaining number of samples into the training set. Specifically, when the loss of the model does not decrease continuously for n epochs, it means that its classification performance has approached a maximum point, indicating that the model has entered a convergence state. Afterward, a predetermined number of maximum loss samples are selected through the validation process, and these samples are merged back into the training set. The newly merged

samples may lead to significant fluctuations in the loss during the following epochs. However, as the model converges again, its classification performance will be further improved.

During the training process guided by GTS, the loss of the model exhibits a periodic and fluctuating downward trend, incorporating cycles of both increases and decreases. Whenever new samples are merged into the training set, the ability of the model to understand feature information will be further improved. After all batches of samples are merged, the classification performance of the model will also reach its optimal state. This process not only expands the training data of the model but also enhances the model's ability to understand various features in the dataset, thereby improving the overall performance of the model.

The appropriate timing for sample merging can be subdivided into two situations. The first one is to supplement new samples to the training set in a timely manner when the model reaches a "pre-convergence" state for the first time, which can accelerate the convergence of the model and quickly approach the optimal performance level. Another is that when the model has already achieved an optimal state, the classification performance tends to stabilize, and it is difficult to have a significant improvement if there is no addition of new samples. Therefore, continuously inputting new samples to the network in this stage can enable it to learn more feature information, thus further improving its classification performance.

III. Experimental Results and Analysis

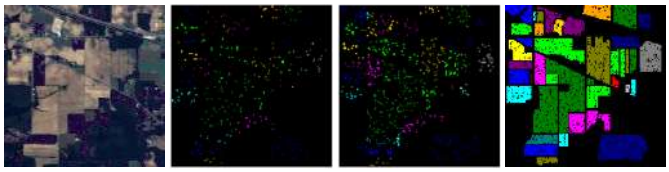
In this section, the effectiveness of the proposed method will be verified. First, the datasets, hardware configuration, parameter settings, and comparison methods used in the experiment are introduced. Then, the proposed method is comprehensively tested and analyzed from multiple perspectives such as ablation experiments, quantitative evaluation, and visual evaluation.

A. Dataset Description

To verify the performance of the proposed method, sufficient experiments were conducted on four publicly available datasets, including Indian Pines (IP), Pavia University (UP), WHU-Hi-LongKou (LK), and Houston 2013 (HT).

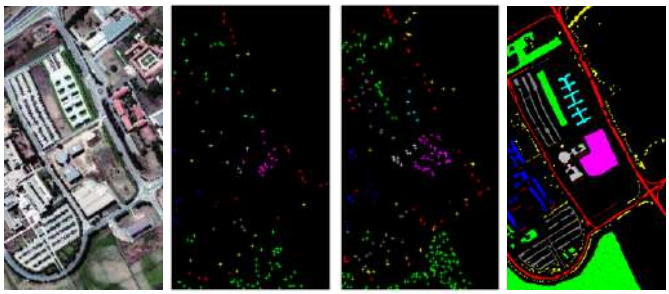
The IP dataset is an HSI captured through airborne visible/infrared imaging spectrometer sensor (AVIRIS) in Indiana, USA, in June 1992, with a size of 145 × 145 pixels, used for early image classification research. Its wavelength range covers 400–2500 nm, with a spatial resolution of 20 m, and includes 220 bands. However, due to the presence of absorbent bands, the actual number of bands used for training has been reduced to 200. This dataset covers 16 types of land cover, including corn, oats, wheat, and so on, with a total of 21 025 pixels, of which 10 249 are land cover pixels and the rest are background pixels.

The UP dataset is an HSI captured through ROSIS-03 in Pavia, Italy, in 2003. It covers 115 bands in the wavelength range of 0.43–0.86 μm, with a spatial resolution of 1.3 m. However, due to the influence of noise, 103 bands are usually used. This dataset contains 610 × 340 pixels, with a total of 2 247 400 pixels, but only 42 776 pixels represent features, covering nine types of features such as trees and asphalt roads.



(a) (b) (c) (d)

Fig. 4. Sample distribution on the IP dataset. (a)–(d) Pseudocolor map, training set, validation set, and test set.



(a) (b) (c) (d)

Fig. 5. Sample distribution on the UP dataset. (a)–(d) Pseudocolor map, training set, validation set, and test set.

The LK dataset is HSI collected on July 17, 2018, in Longkou Town, Hubei Province, China, using hyperspectral imaging sensors carried by DJI Matrice 600 Pro drones. The research area includes nine crops, such as corn, cotton, narrow-leaf soybeans, and rice. The image size is 550 × 400 pixels, with a spatial resolution of approximately 0.463 m, covering wavelengths ranging from 400 to 1000 nm and containing 270 bands.

The HT dataset is an HSI collected in Houston, Texas, and surrounding rural areas in the United States, using the CASI-1500 sensor. The research area includes 15 ground object categories, such as trees, soil, commercial, and highway. The image size is 349 × 1905 pixels, with a spatial resolution of 2.5 m, covering a wavelength range from 364 to 1046 nm and containing 144 bands.

Tables I and II list the main land cover categories involved in the four research scenarios mentioned above, as well as the number of training, validation, and testing samples used for classification tasks. Correspondingly, Figs. 4–7 show the spatial distribution of pseudocolor map, training, validation, and testing samples for four research scenarios. To facilitate the observation of the spatial distribution of the sample, we perform scaling on the pixels.

B. Experimental Configuration

1) Hardware Configuration: The proposed method is implemented in PyTorch 1.10.1 and Python 3.7.0 environments. The hardware configuration consists of an Intel Core i9-9900K CPU with 128-GB RAM and an NVIDIA RTX 3090 GPU. To avoid the randomness of the results, the average of ten independent experiments was taken for all experimental results.

TABLE I

LAND COVER CLASSES OF THE IP AND UP DATASETS, ALONG WITH THE NUMBER OF TRAINING, VALIDATION, AND TESTING SAMPLES FOR EACH CLASS

No	Indian Pines				Pavia University			
	Class Name	Training Origin/Final	Validation Origin/Final	Test	Class Name	Training Origin/Final	Validation Origin/Final	Test
1	Alfafa	2/-	2/-	42	Asphalt	20/-	41/-	6570
2	Corn-notill	43/-	105/-	1280	Meadows	56/-	135/-	18458
3	Corn-mintill	25/-	64/-	741	Gravel	7/-	14/-	2078
4	Corn	8/-	12/-	217	Trees	10/-	22/-	3032
5	Grass-passture	15/-	28/-	440	Painted metal sheets	5/-	5/-	1335
6	Grass-trees	22/-	51/-	657	Bare soil	16/-	36/-	4977
7	Grass-passture-mowed	1/-	4/-	23	Bitumen	4/-	11/-	1315
8	Hay-windrowed	15/-	28/-	435	Self-Blocking Bricks	12/-	30/-	3640
9	Oats	1/-	2/-	17	Shadows	3/-	5/-	939
10	Soybean-notill	30/-	70/-	872				
11	Soybean-mintill	74/-	181/-	2200				
12	Soybean-clean	18/-	41/-	534				
13	Wheat	7/-	15/-	183				
14	Woods	38/-	87/-	1140				
15	Buildings-Grass-Trees-Drivers	12/-	20/-	354				
16	Stone-Steel-Towers	3/-	7/-	83				
-	Total	314/514	717/517	9218		133/213	299/219	42344

TABLE II

LAND COVER CLASSES OF THE LONGKOU AND HT DATASETS, ALONG WITH THE NUMBER OF TRAINING, VALIDATION, AND TESTING SAMPLES FOR EACH CLASS

No	Longkou				Houston 2013			
	Class Name	Training Origin/Final	Validation Origin/Final	Test	Class Name	Training Origin/Final	Validation Origin/Final	Test
1	Corn	8/-	54/-	34439	Healthy Grass	26/-	50/-	1175
2	Cotton	5/-	17/-	8352	Stressed Grass	26/-	42/-	1186
3	Sesame	2/-	5/-	3024	Synthetic Grass	14/-	31/-	652
4	Broad-leaf soybean	32/-	91/-	63089	Trees	25/-	49/-	1170
5	Narrow-leaf soybean	3/-	6/-	4142	Soil	25/-	40/-	1177
6	Rice	6/-	16/-	11832	Water	7/-	8/-	310
7	Water	34/-	102/-	66920	Residential	26/-	66/-	1176
8	Roads and houses	4/-	8/-	7112	Commerical	25/-	56/-	1163
9	Mixed weed	8/-	7/-	5219	Road	26/-	57/-	1169
10					Highway	25/-	44/-	1158
11					Railway	25/-	46/-	1164
12					Parking Lot 1	25/-	41/-	1167
13					Parking Lot 2	10/-	16/-	443
14					Tennis Court	9/-	20/-	399
15					Running Track	14/-	35/-	611
-	Total	107/207	306/206	204129		308/458	601/451	14120

2) Evaluation Metric: Three common evaluation metrics were adopted to comprehensively evaluate the classification performance of each model, namely, OA, average accuracy (AA), and kappa coefficient ($\kappa \times 100$).

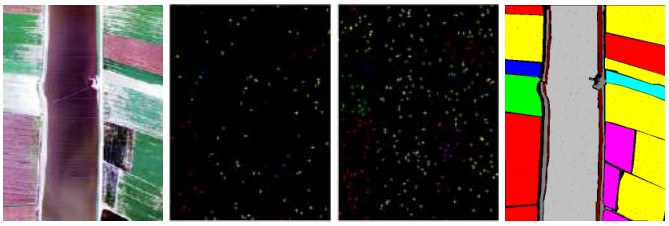


Fig. 6. Sample distribution on the Longkou dataset. (a)–(d) Pseudocolor map, training set, validation set, and test set.

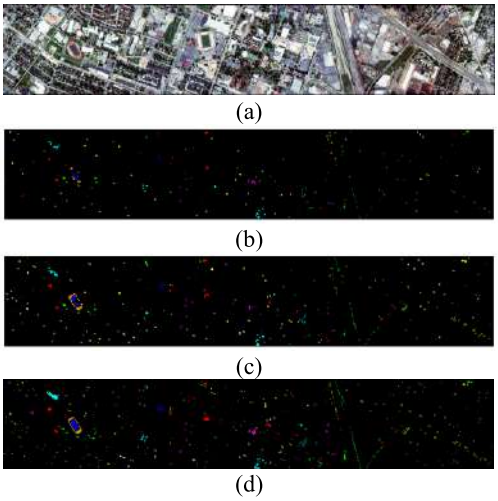


Fig. 7. Sample distribution on the HT dataset. (a)–(d) Pseudocolor map, training set, validation set, and test set.

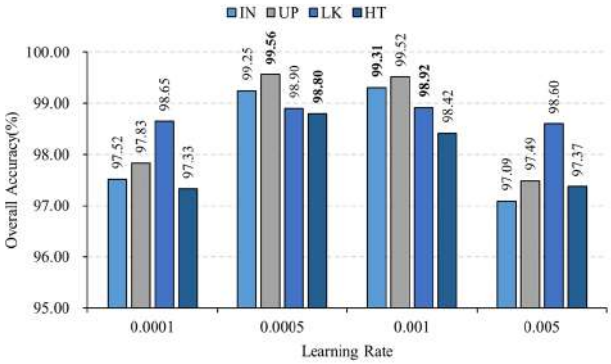


Fig. 8. Comparison of different learning rates on four datasets.

3) Parameter Configuration: To ensure a fair evaluation of the proposed methods, all comparison methods adopted their recommended optimal parameter configurations. We use the Adam optimizer to train our network, with training epochs set at 500. The learning rates of the IP and LK datasets were set to 1e-3, and the learning rates of the UP and HT datasets were set to 5e-4 (see Fig. 8). The optimal superpixel segmentation scales for the IP, UP, and LK datasets were 100, 400, and 300, respectively (see Table III). Considering the issue of memory overflow that tends to occur with graph-based methods when applying the HT dataset at segmentation scales ranging from 100 to 500, we uniformly set the segmentation scale to 600 for this dataset in our experiments. Fig. 9 illustrates the number of consecutive loss increases needed to trigger GTS. The optimal settings for the IP, UP, LK, and HT datasets were 7, 5, 7, and 9, respectively.

TABLE III
DIFFERENT SEGMENTATION SCALES

Dataset/Scale	100	200	300	400	500
Indian Pines	**99.31**	99.17	99.23	99.09	99.01
Pavia University	99.42	99.47	99.50	**99.56**	99.50
Long Kou	98.79	98.76	**98.92**	98.85	98.61

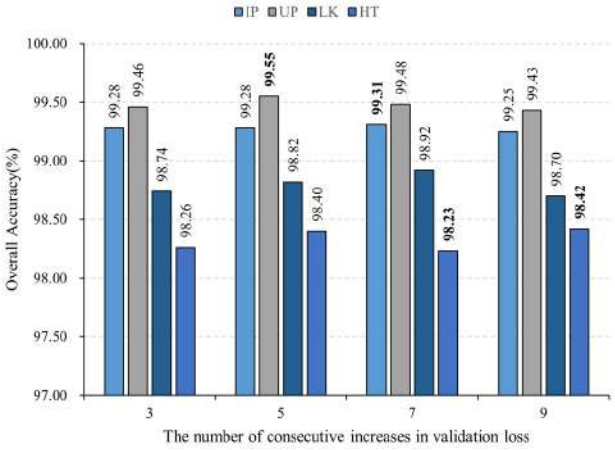


Fig. 9. Comparison of the number of consecutive increases in losses required to trigger GTS in different datasets.

4) Comparison With State-of-the-Art Methods: Several representative methods are selected for the following comparison experiments. They are HybridSN [26], DTAN [34], GNet [40], SF [44], SSFTT [45], CEGCN [50], FDGC [51], WFCG [54], and AMGCFN [53]. The innovative points in the structures of these comparative methods are given as follows.

1) HybridSN incorporates two residual blocks: one leveraging 3-D convolutions to capture both spectral and spatial features, and another employing 2-D convolution to extract spatial features.

2) DTAN consists of two branches. The spectral branch embeds the efficient CA (ECA) module into the DenseNet to realize cross-channel interaction. Afterward, the channel triple attention module is used to obtain the final spectral feature map. The spatial branch has a structure similar to the spectral branch, but no longer uses ECA modules.

3) GNet adopts a structure of multiple different grids in its design, each consisting of three $3 \times 3 \times 1$ sized 3-D convolution kernels on four edges and one $1 \times 1 \times 3$ sized 3-D convolution kernel. This design allows spectral and spatial features to be extracted twice at different depth levels, thereby increasing the diversity of the model.

4) SF consists of five layers of Transformer encoders. Cross-layer adaptive fusion (CAF) modules are employed between nonadjacent encoders to propagate information from shallow layers to deeper ones. In addition, the network adopts a groupwise spectral embedding strategy for its input.

TABLE IV
ABLATION EXPERIMENTS

Case.	Components.			Dataset.			
	GSA	MAF	GTS	IP	UP	LK	HT
1	√	—	—	81.66	85.76	90.92	76.31
2	—	√	—	96.55	97.68	98.03	96.46
3	√	√	—	98.51	98.78	98.64	97.10
4	√	—	√	82.20	86.51	91.42	77.45
5	—	√	√	99.16	98.97	98.76	97.99
6	√	√	√	**99.31**	**99.56**	**98.92**	**98.80**

5) SSFTT extracts shallow spatial–spectral features using 3-D and 2-D convolutional kernels. Then, it converts these shallow features into high-level semantic features through a Gaussian distribution weighted tokenization module. Finally, Transformer encoders are used to learn the relationships between high-level semantic features.

6) CEGCN consists of two branches. One branch employs GCN to extract coarse-grained information based on superpixels, while the other branch utilizes depthwise separable convolution to extract fine-grained information based on pixels. The combination of GCN and CNN was first proposed by CEGCN.

7) FDGC consists of three branches. Two branches employ convolution to extract channel information at different scales, while the third branch utilizes dynamic GCN to adaptively capture topological structure information.

8) WFCG adopts a similar architecture to CEGCN but replaces the GCN with a GAT to address the challenges of handling directed graphs. Moreover, WFCG integrates an attention mechanism into the convolutional branch to enable the model to focus on discriminative features.

9) AMGCFN consists of two branches. One branch captures pixel-level features using multiscale convolutional kernels; the other branch forms a cascade network by stacking multihop GCNs to effectively extract structural information. Finally, a cross-attention fusion module is employed to obtain discriminative features for classification.

C. Ablation Experiments

In this section, we validate the effectiveness of the three modules in the proposed method. The experimental data adopts OA as the evaluation metric, and the results are shown in Table IV. The experimental results demonstrate that the GSA and MAF branches exhibit distinct emphases in feature extraction, and their integration significantly enhances the classification performance of the model.

Specifically, Cases 1 and 2 showcase the individual effects of the GSA and MAF branches. The GSA branch primarily extracts coarse-grained information from images, focusing on overall features and trends, which is suitable for capturing global semantics. However, due to this focus, the features extracted by GSA overlook substantial details, thus limiting its performance. In contrast, the MAF branch effectively extracts local features through convolutional kernels, providing a fine-grained feature representation to the model. These fine-grained details significantly improve the classification performance of the model.

Subsequently, the fusion of the GSA and MAF branches further boosts the classification performance, as evidenced by the results of Case 3. Compared to using MAF alone, the model's performance shows a marked improvement, with increases of 1.96%, 1.10%, 0.61%, and 0.64% on four datasets, respectively. This underscores the effective integration of the proposed convolutional (MAF) and graph (GSA) branches. Notably, the fine-grained information provided by the MAF branch offers a more reliable basis for the model's classification decisions, while the GSA branch aids in filtering out less crucial features, thereby enhancing the model's robustness.

Furthermore, Cases 4 and 5 verify the efficacy of the GTS strategy when paired with either the GSA or MAF branch. Regardless of the partner branch, GTS consistently yields notable performance gains. Notably, the combination of the MAF branch and GTS strategy achieves the best experimental outcomes. Compared to using MAF alone, the model experiences improvements of 2.61%, 1.29%, 0.73%, and 1.53% on the four datasets, respectively. This performance enhancement even surpasses achieved by combining GSA and MAF, underscoring the effectiveness of the GTS strategy, which improves model performance by introducing samples at opportune moments during training.

Finally, in Case 6, the model incorporates both branches and adopts the GTS training strategy. Under these conditions, the model achieves the best classification performance among all six experiments. Compared to Experiment 3 without GTS, the model exhibits improvements of 0.8%, 0.77%, 0.28%, and 1.70% on the four datasets, respectively.

In this section, we validate the effectiveness of the three modules in the proposed method. The experimental data adopts OA as the evaluation metric, and the results are shown in Table IV.

The primary motivation behind the design of GSA is to alleviate the limitation of GAT, which is restricted to focusing solely on neighboring nodes. MHSA, on the other hand, excels at capturing dependencies between any two positions within a sequence. Therefore, combining GAT and MHSA is theoretically feasible. To validate this hypothesis, we conducted in-depth evaluations of GSA.

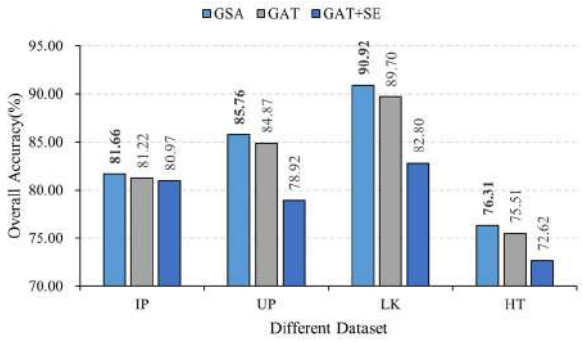


Fig. 10. Comparison of GSA, GAT, and GAT + SE in OA.

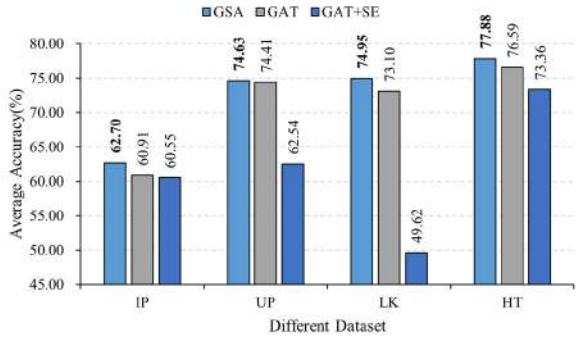


Fig. 11. Comparison of GSA, GAT, and GAT + SE in AA.

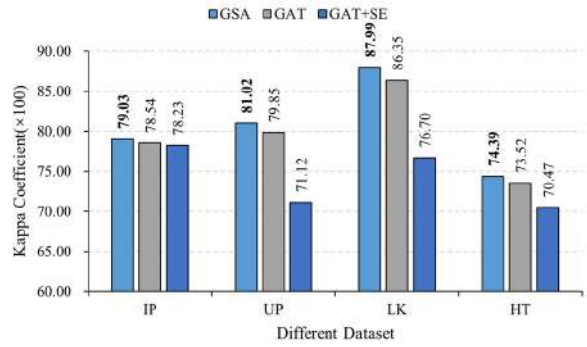


Fig. 12. Comparison of GSA, GAT, and GAT + SE in $\kappa \times 100$.

Experimental results (see Figs. 10–12) present a comparison of GSA, GAT, and its variant across multiple datasets in terms of OA, AA, and $\kappa \times 100$. The results demonstrate that GSA consistently outperforms GAT on all datasets. Moreover, the combination of GAT and SE significantly underperforms both GSA and GAT. Especially, on the LK dataset, GAT + SE achieves an AA of only 49.62%, a 25.33% decrease compared to GAT. These experimental results fully validate the effectiveness of GSA and indicate that combining GAT with MHSA is not a simple superposition or arbitrary selection. By integrating GAT's capability in modeling local relationships with MHSA's ability to capture global dependencies, GSA achieves a more comprehensive graph representation learning.

D. Quantitative Experiments

In this section, we conducted a quantitative evaluation of the proposed method and the comparison methods on four datasets. The experimental results, as shown in Tables V–VIII, demonstrate that the proposed method exhibits significant advantages across all four datasets.

Specifically, although HybridSN combines 2-D and 3-D convolutions, it suffers from a slight lack of classification performance due to the absence of an attention mechanism. DTAN, on the other hand, introduces an attention module to enable the network to capture the importance of different features more accurately, thus outperforming HybridSN. GNet explores heterogeneous spatial and spectral features from an anisotropic perspective and achieves good performance by fusing low-level features and high-level semantic features across layers. SF primarily focuses on spectral features, paying insufficient attention to spatial features, leading to suboptimal performance. In contrast, SSFTT achieves relatively good classification results by combining convolutions and feature tokenizers to extract both low- and high-level semantic features.

Methods that combine GNNs and CNNs have generally demonstrated superior performance. CEGCN, by combining GCN and CNN, fuses superpixels and pixels, achieving excellent results. WFCG, AMGCFN, and the proposed method adopt a similar structure. WFCG, building upon CEGCN, replaces GCN with GAT to alleviate the limitations of GCN in handling directed graphs and introduces an attention module, thereby further improving performance. AMGCFN constructs a multihop GCN and a multiscale CNN, providing new insights for research in this field and achieving outstanding performance. In contrast, the dynamic GCN in FDGC fails to demonstrate satisfactory classification performance.

Next, the proposed method is analyzed in this section. On the IP dataset, the proposed method outperforms the best comparison method by 1.16%, 2.48%, and 1.31% in OA, AA, and $\kappa \times 100$ metrics, respectively. In addition, the proposed GS-GraphSAT achieves optimal performance in 11 out of 16 classes. Especially, for the seventh and ninth classes, the total number of samples for these two classes is quite small. The accuracy of these two classes using the proposed method is as high as 98.43% and 98.35%, which exceed that of CEGCN and AMGCFN by 4.65% and 1.07%, respectively. This once again proves the feasibility of alleviating the problem of sample limitation by improving sample utilization.

On the UP dataset, GS-GraphSAT has also demonstrated significant advantages. Compared to the best comparison method, the three metrics have improved by 0.75%, 1.22%, and 1.00%, respectively. In the optimal class, GS-GraphSAT achieved the best performance in six out of nine classes. On the LK dataset, the proposed method still has advantages. Compared to the best comparison method, the three metrics are 0.63%, 1.41%, and 0.82% higher, respectively, and GS-GraphSAT leads in four out of nine classes. On the HT dataset, the three metrics were 1.70%, 1.94%, and 1.83% higher, respectively, and GS-GraphSAT achieved optimal performance in 11 out of 15 classes.

E. Comparison of Running Time and Model Complexity

To comprehensively evaluate the performance of each method, we conducted a quantitative analysis of their running

TABLE V
CLASSIFICATION PERFORMANCE OBTAINED BY DIFFERENT METHODS FOR THE IP DATASET (OPTIMAL RESULTS ARE BOLDED)

No.	HybridSN	DTAN	GNet	SF	SSFTT	CEGCN	FDGC	WFCG	AMGCFN	Proposed
1	87.37±15.19	79.35±39.69	92.30±8.07	80.69±17.20	90.69±15.84	72.10±17.27	96.98±4.92	97.47±2.74	95.48±4.49	**98.02±1.85**
2	92.36±2.51	96.46±1.02	96.13±2.39	71.07±4.24	95.57±1.87	97.52±1.29	94.00±3.16	97.43±1.29	95.88±2.49	**98.96±0.62**
3	93.28±3.81	95.42±1.14	94.81±3.29	66.14±3.93	96.17±1.78	98.11±1.58	95.30±2.03	97.05±1.95	97.79±1.29	**98.79±1.11**
4	94.91±3.62	96.5±1.85	97.13±2.89	64.18±40.75	94.55±4.87	90.81±8.15	95.37±4.35	98.70±1.06	97.27±2.47	**99.35±0.75**
5	94.08±5.28	97.20±1.73	97.34±1.72	87.02±4.11	94.01±4.07	96.25±1.20	96.86±3.56	97.36±1.27	94.02±2.68	**98.63±1.62**
6	94.95±4.44	98.31±0.78	97.90±1.28	89.41±3.48	98.31±1.19	99.20±0.48	97.78±1.65	98.87±0.78	98.51±1.09	**99.66±0.38**
7	80.84±19.41	84.83±20.08	75.76±18.29	75.70±26.88	91.20±17.78	93.78±8.62	81.11±17.96	75.29±37.99	92.53±10.61	**98.43±3.13**
8	99.49±0.47	97.97±2.43	99.57±0.73	90.93±1.62	99.47±1.58	**100**	99.72±0.58	99.95±0.09	99.79±0.28	99.93±0.14
9	62.89±19.94	72.00±37.09	65.12±20.17	68.25±14.52	95.29±5.76	68.16±23.11	84.66±17.16	71.76±39.27	97.28±4.37	**98.35±3.44**
10	94.22±3.01	96.03±0.57	95.81±12.31	71.58±5.58	93.65±2.13	95.09±2.58	91.84±4.47	96.15±1.89	94.33±2.67	**98.62±1.01**
11	96.37±1.78	97.76±0.41	97.53±0.90	72.51±1.98	98.15±1.15	98.24±1.09	95.36±2.90	98.86±0.85	98.85±0.39	**99.74±0.19**
12	94.19±2.29	94.86±4.91	90.97±5.64	66.21±6.06	89.52±4.29	**98.63±1.27**	95.36±6.80	98.14±0.69	95.17±2.13	98.57±0.87
13	93.26±8.48	95.8±2.24	97.26±3.34	92.71±5.30	99.53±0.58	**99.78±0.43**	97.10±4.65	98.78±1.45	97.70±3.07	99.67±0.82
14	97.37±1.09	**99.91±0.09**	98.69±5.87	89.26±1.78	98.69±1.41	99.59±0.42	98.30±1.75	99.75±0.19	99.05±0.61	99.80±0.19
15	91.71±5.50	96.29±1.34	96.07±2.53	67.14±6.25	96.89±2.41	89.69±8.56	92.29±6.25	98.66±1.76	98.24±1.91	**99.79±0.22**
16	91.75±9.35	99.25±1.5	88.02±8.21	94.07±49.44	90.11±9.72	95.07±4.35	86.87±11.58	97.79±2.93	93.95±3.42	99.17±0.74
OA	94.61±0.95	97.16±0.31	96.35±0.49	76.48±1.23	96.42±0.53	97.40±0.66	94.63±0.82	98.15±0.36	97.36±0.52	**99.31±0.17**
AA	91.19±1.26	93.62±4.65	92.52±2.18	77.93±2.54	95.11±1.80	93.25±2.71	93.11±1.69	95.12±5.04	96.61±1.34	**99.09±0.25**
κ×100	93.85±1.09	96.77±0.35	95.84±0.57	73.02±1.38	95.91±0.59	97.04±0.76	93.87±0.95	97.90±0.41	96.99±0.60	**99.21±0.19**

TABLE VI
CLASSIFICATION PERFORMANCE OBTAINED BY DIFFERENT METHODS FOR THE UP DATASET (OPTIMAL RESULTS ARE BOLDED)

No.	HybridSN	DTAN	GNet	SF	SSFTT	CEGCN	FDGC	WFCG	AMGCFN	Proposed
1	88.43±4.39	71.04±2.43	96.48±1.86	87.40±3.55	93.62±2.53	98.73±1.07	84.62±6.06	98.86±0.82	98.01±1.12	**99.09±0.59**
2	97.58±1.26	98.65±0.30	98.62±0.61	87.21±1.63	99.56±0.30	99.92±0.05	98.26±0.82	99.91±0.05	99.96±0.03	**99.99±0.01**
3	82.15±5.69	85.39±8.43	97.27±1.55	55.95±6.82	83.4±5.80	82.17±8.90	93.85±5.32	96.24±3.77	95.71±3.11	**99.39±0.43**
4	86.85±5.74	98.01±0.63	**98.23±0.89**	91.41±4.72	91.15±3.45	92.11±2.98	83.76±11.59	94.37±1.69	93.65±1.89	97.99±0.71
5	97.51±2.76	96.93±1.58	97.22±4.31	96.10±2.62	99.98±0.04	**100**	97.88±2.53	99.94±0.12	99.74±0.23	99.98±0.03
6	98.20±0.79	91.49±2.78	99.52±0.43	69.49±7.66	95.06±5.43	99.94±0.12	96.24±3.62	99.68±0.56	99.92±0.12	**100**
7	79.14±6.07	96.14±3.40	99.38±1.33	65.61±12.32	98.16±1.94	96.78±4.49	99.07±1.68	99.67±0.26	97.09±2.96	**99.84±0.12**
8	79.50±6.75	65.99±9.44	89.31±2.91	71.91±3.34	85.09±6.42	96.38±2.93	69.62±7.23	96.65±1.21	96.88±1.61	**99.02±0.92**
9	69.13±6.49	**100**	98.14±2.22	99.44±0.54	82.39±9.61	95.74±4.69	72.14±22.32	97.47±3.91	93.35±3.01	98.44±1.67
OA	91.79±1.62	88.19±1.95	97.37±1.95	82.19±1.42	95.06±0.66	97.81±0.51	90.47±1.23	98.81±0.32	98.48±0.26	**99.56±0.11**
AA	86.50±2.25	89.29±2.08	97.13±2.08	80.50±1.79	92.05±1.22	95.75±1.24	88.38±2.35	98.09±0.55	97.15±0.52	**99.31±0.14**
κ×100	89.09±2.16	84.19±2.65	96.51±2.65	76.18±1.84	93.43±0.89	97.09±0.68	87.30±1.65	98.41±0.43	97.99±0.34	**99.41±0.15**

time and model complexity, as shown in Table IX. Methods employing superpixels and pixel fusion demonstrated significant advantages in terms of time consumption, primarily attributed to their full-pixel input approach, which eliminates the overhead of batch training.

Among the methods combining GNN and CNN, the proposed model consumed the most training time but performed the best. Due to the lack of attention mechanisms, the CEGCN model had the shortest training time but relatively average performance. The AMGCFN model exhibited the best testing time and relatively good performance. Although the proposed model had a slightly longer training time, it demonstrated significant advantages in classification performance.

On the other hand, the HybridSN and FDGC models have the highest model complexity, while the GNet model has the fewest parameters. The computational cost of our proposed model is relatively high due to the nature of the multihead attention mechanism.

TABLE VII
CLASSIFICATION PERFORMANCE OBTAINED BY DIFFERENT METHODS FOR THE LONGKOU DATASET (OPTIMAL RESULTS ARE BOLDED)

No.	HybridSN	DTAN	GNet	SF	SSFTT	CEGCN	FDGC	WFCG	AMGCFN	Proposed
1	94.20±2.16	97.53±2.51	99.48±0.61	95.26±1.81	99.61±0.21	99.76±0.15	97.88±2.87	99.82±0.12	99.64±0.26	**99.85±0.11**
2	80.59±10.49	87.70±6.53	**99.39±0.31**	67.67±11.02	93.00±8.05	91.61±6.48	89.87±12.61	93.29±4.89	96.31±3.26	97.89±1.89
3	83.91±10.95	**100**	99.44±1.13	60.41±20.07	95.70±4.36	83.28±5.05	92.83±18.69	90.41±3.07	94.61±6.88	89.26±5.74
4	94.66±3.11	97.34±0.68	98.00±0.97	90.28±2.05	97.45±1.70	99.21±0.39	94.96±3.29	99.23±0.26	98.78±4.09	**99.53±0.13**
5	77.66±14.50	**98.16±3.18**	97.66±3.15	54.01±8.64	85.55±6.94	84.91±14.45	75.40±19.58	86.06±11.03	93.30±4.22	94.14±3.53
6	84.24±8.27	97.35±1.72	98.86±1.19	93.84±3.46	96.77±3.19	99.31±0.48	99.34±0.78	**99.40±0.50**	97.15±1.04	98.51±2.41
7	99.05±0.76	99.73±0.39	99.56±0.52	99.21±0.61	98.81±1.09	**99.99±0.01**	99.40±0.56	99.95±0.04	99.80±0.25	99.97±0.04
8	78.27±9.63	76.42±11.99	81.89±5.28	68.59±10.49	82.57±11.67	93.48±2.49	75.38±13.27	92.78±3.75	83.20±3.78	**95.80±1.31**
9	82.45±9.39	75.93±19.15	76.09±9.91	67.23±13.26	76.17±12.59	59.68±7.52	53.32±16.63	82.81±11.06	80.47±10.19	**88.14±7.61**
OA	93.16±1.89	96.34±1.07	97.69±0.67	90.89±0.96	96.71±0.93	97.52±0.36	93.19±1.67	98.29±0.37	97.88±0.31	**98.92±0.25**
AA	86.11±4.06	92.24±3.01	94.49±1.69	77.39±3.76	91.74±2.08	90.14±1.56	86.49±2.96	93.75±1.72	93.70±1.54	**95.90±1.13**
κ×100	91.02±2.45	95.18±1.42	96.96±0.89	87.97±1.28	95.69±1.22	96.72±0.49	91.04±2.19	97.75±0.49	97.21±0.41	**98.57±0.33**

TABLE VIII
CLASSIFICATION PERFORMANCE OBTAINED BY DIFFERENT METHODS FOR THE HT DATASET (OPTIMAL RESULTS ARE BOLDED)

No.	HybridSN	DTAN	GNet	SF	SSFTT	CEGCN	FDGC	WFCG	AMGCFN	Proposed
1	93.96±2.33	98.68±0.65	95.28±2.31	92.83±1.74	96.79±2.34	97.18±2.27	91.10±4.36	96.18±2.24	96.91±1.42	**98.67±1.11**
2	96.21±2.01	98.31±0.21	98.23±2.25	96.90±1.85	98.75±0.73	98.03±0.90	93.68±5.64	98.99±0.42	98.29±1.62	**99.42±0.18**
3	98.10±2.08	**100**	99.86±0.41	98.69±1.40	99.21±0.69	99.91±0.12	99.22±1.09	99.96±0.06	99.53±0.69	**100**
4	91.22±2.46	97.25±0.36	97.29±1.59	97.78±1.11	97.24±1.99	**99.88±0.15**	91.02±4.70	96.99±1.95	98.18±2.16	99.84±0.18
5	96.76±3.50	95.24±0.00	98.67±2.55	95.24±1.29	99.99±0.02	99.88±0.23	99.01±1.86	99.79±0.25	99.54±0.37	**100**
6	95.84±5.14	99.53±0.33	**99.60±0.71**	98.22±2.51	94.08±5.31	91.04±3.84	98.38±2.56	94.01±4.82	92.74±8.62	97.53±2.14
7	84.23±4.55	93.25±1.37	91.71±2.19	89.79±2.68	92.86±3.32	98.14±1.49	80.24±5.91	96.28±1.91	98.53±0.97	**99.22±0.91**
8	96.06±3.75	98.13±0.18	**98.93±1.16**	77.25±2.41	88.40±3.25	90.32±5.57	98.00±2.19	91.09±1.84	88.86±2.31	93.62±2.51
9	86.31±5.63	90.02±0.62	93.19±2.73	79.99±2.67	88.19±3.83	93.83±2.31	88.09±2.86	91.41±2.27	93.25±2.76	**97.83±1.15**
10	91.59±4.23	91.44±0.68	93.02±3.59	81.76±4.76	98.31±2.43	99.58±0.83	90.67±6.44	99.60±0.34	98.12±1.72	**100**
11	94.48±3.34	98.93±0.87	95.61±2.05	83.35±2.34	98.58±1.68	98.07±2.09	95.03±1.71	98.57±1.19	97.28±1.67	**99.60±0.79**
12	95.07±2.99	91.22±0.94	96.25±1.43	76.39±3.35	95.76±2.19	96.58±2.59	93.49±6.15	96.81±2.29	95.79±2.00	**99.65±0.01**
13	91.20±4.58	95.80±1.50	95.72±2.63	46.27±6.36	94.08±2.51	89.47±1.72	82.75±8.88	93.16±3.57	93.93±4.27	**96.03±3.45**
14	97.08±3.58	**100**	99.04±1.41	94.44±4.72	**100**	**100**	96.80±4.63	**100**	99.85±0.29	99.55±0.89
15	93.76±3.08	**100**	97.19±1.97	98.75±1.04	99.68±0.93	99.96±0.06	96.05±4.72	**100**	99.77±0.31	**100**
OA	92.88±1.11	95.86±0.09	96.11±5.38	87.71±0.99	95.89±0.72	97.10±0.76	91.96±1.06	96.81±0.43	96.70±0.59	**98.80±0.41**
AA	93.46±1.19	96.52±0.04	96.64±4.36	87.18±1.11	96.13±0.79	96.79±0.69	92.90±1.24	96.85±0.64	96.71±0.91	**98.73±0.46**
κ×100	92.31±1.21	95.52±0.10	95.79±5.82	86.70±1.06	95.55±0.77	96.87±0.82	92.31±1.14	96.55±0.46	96.43±0.63	**98.70±0.44**

Experimental results showed that the proposed model achieved a good balance between performance and efficiency. Although running time and model complexity are not particularly low, it could more accurately identify ground object classes in complex scenarios, especially for small sample regions.

F. Comparison of Classification Maps

This section visually demonstrates the superiority of the proposed method by comparing classification maps across four datasets with other methods. As shown in Figs. 13–16, in regions with labeled samples, models combining GNNs and CNNs, including the proposed method, consistently outperform other methods, aligning with the quantitative results in Tables V–VIII. However, the true performance of a model should be evaluated based on its predictions in unlabeled regions.

We selected specific regions from the UP and HT datasets, zoomed in, and created pseudocolor maps. In the mixed region of "Meadows" and "Bare Soil" in the UP dataset, the proposed method clearly preserves edge information and accurately distinguishes between the two land cover types, especially for the "Meadows" mixed within "Bare Soil."

TABLE IX
COMPARISON OF RUNNING TIME AND COMPLEXITY OF VARIOUS METHODS ON FOUR DATASETS (OPTIMAL RESULTS ARE BOLDED)

Datasets	Metric	HybridSN	DTAN	GNet	SF	SSFTT	CEGCN	FDGC	WFCG	AMGCFN	Proposed
IN	Train(s)	0.20	0.88	0.30	0.21	0.17	**0.01**	0.13	0.03	0.06	0.09
	Test(s)	1.13	8.98	5.33	1.57	0.79	0.37	1.04	0.41	**0.01**	0.56
	Params(K)	6037.63	331.73	**49.58**	346.36	401.74	167.41	2383.68	102.42	368.22	326.94
	FLOPs(G)	0.99	1.14	3.24	2.15	**0.73**	1.61	3.77	1.62	2.05	2.26
UP	Train(s)	0.10	0.35	0.41	0.10	0.08	**0.05**	0.06	0.17	0.31	0.41
	Test(s)	5.17	40.34	24.30	5.92	4.25	2.62	4.82	3.31	**0.02**	3.89
	Params(K)	6037.63	331.73	**49.58**	346.36	401.74	153.44	2383.68	89.36	368.22	313.88
	FLOPs(G)	0.99	1.14	3.24	2.15	**0.73**	12.91	3.77	13.23	14.74	19.49
LK	Train(s)	0.33	0.82	0.42	0.13	0.04	0.11	0.09	0.30	**0.03**	0.36
	Test(s)	208.92	411.40	116.80	49.56	1.09	3.82	17.04	13.33	**0.02**	5.20
	Params(K)	6037.63	331.21	**49.35**	548.84	401.28	175.15	1921.61	111.06	376.68	335.59
	FLOPs(G)	0.99	1.14	3.24	3.36	**0.11**	18.57	2.85	18.88	23.17	25.54
HT	Train(s)	0.18	0.47	0.49	0.29	0.11	0.19	**0.08**	0.77	0.59	1.05
	Test(s)	3.53	7.09	4.99	2.98	0.76	10.01	0.69	16.61	**0.02**	10.18
	Params(K)	6037.63	331.65	**49.55**	226.68	401.67	159.94	2317.66	95.08	356.36	319.59
	FLOPs(G)	0.99	1.14	12.99	1.35	**0.72**	45.47	2.74	46.25	55.84	66.33

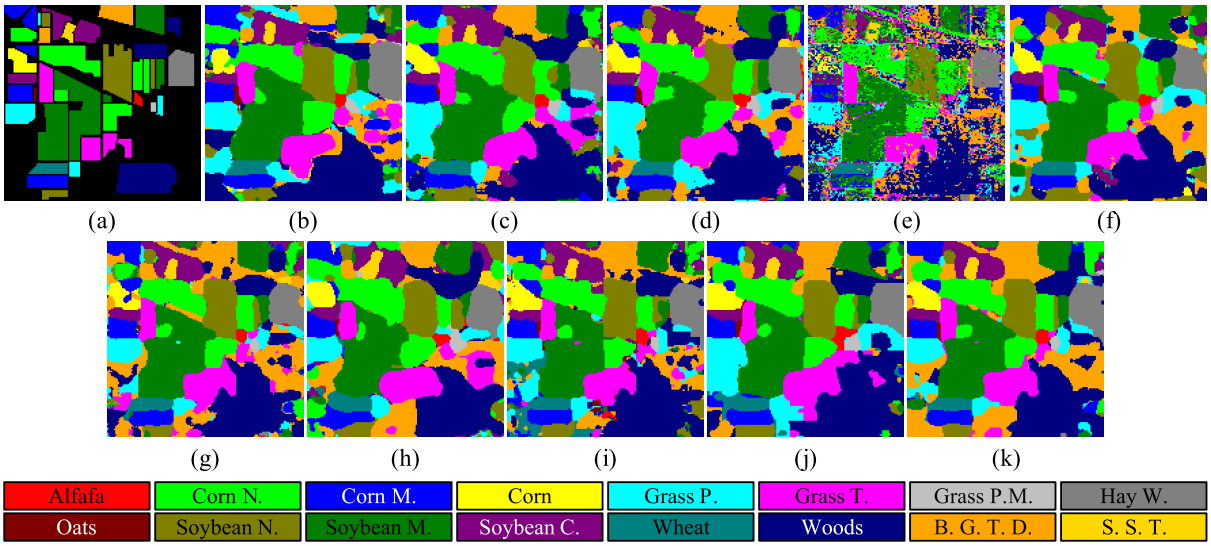


Fig. 13. Classification maps obtained by different methods on the IP dataset. (a)–(k) Ground truth, HybridSN, DTAN, GNet, SF, SSFTT, CEGCN, FDGC, WFCG, AMGCFN, and proposed.

In contrast, other GNN-based comparison methods perform poorly. Notably, CEGCN completely fails to identify "Bare Soil." While HybridSN can identify "Bare Soil," it cannot distinguish the "Meadows" within it. DTAN and SSFTT have incomplete classification map edges and overall poor performance. Although SF preserves edges, its classification results are chaotic.

In the mixed region of "Healthy grass" and "Parking Lot 1" in the HT dataset, almost all methods can accurately predict "Healthy grass," but their performance on the unlabeled "Parking Lot 1" region varies significantly. HybridSN, DTAN, and AMGCFN incorrectly predict the "Residential" class in the lower right corner. Other baseline models fail to identify the "Running Track" surrounded by "Parking Lot 1."

These experimental results further validate the robustness of our proposed method. Even in regions with sparse samples, our proposed method can accurately capture subtle differences between land cover types, thereby achieving more precise classification results.

G. Comparison of Confusion Matrices

This section visually demonstrates the superiority of the proposed method in classification tasks by comparing confusion matrices on the IP and UP datasets. As shown in Fig. 17, our

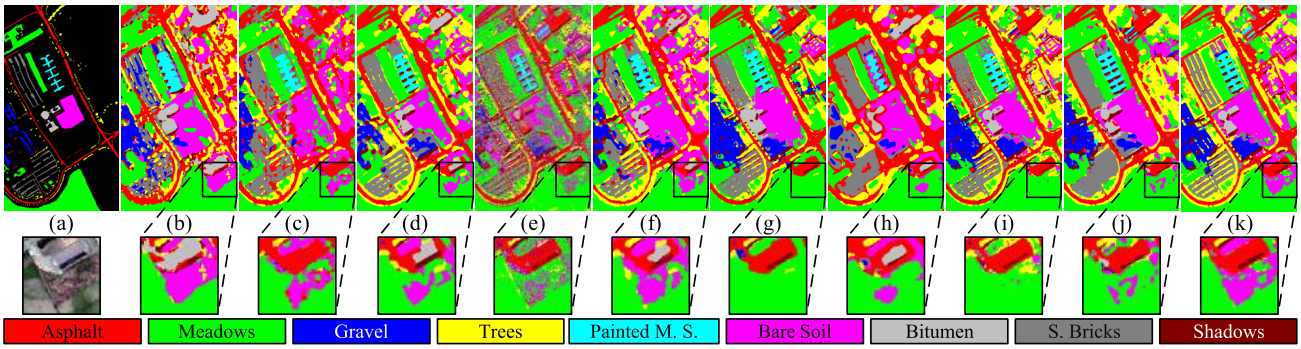


Fig. 14. Classification maps obtained by different methods on the UP dataset. (a)–(k) Ground truth, HybridSN, DTAN, GNet, SF, SSFTT, CEGCN, FDGC, WFCG, AMGCFN, and proposed.

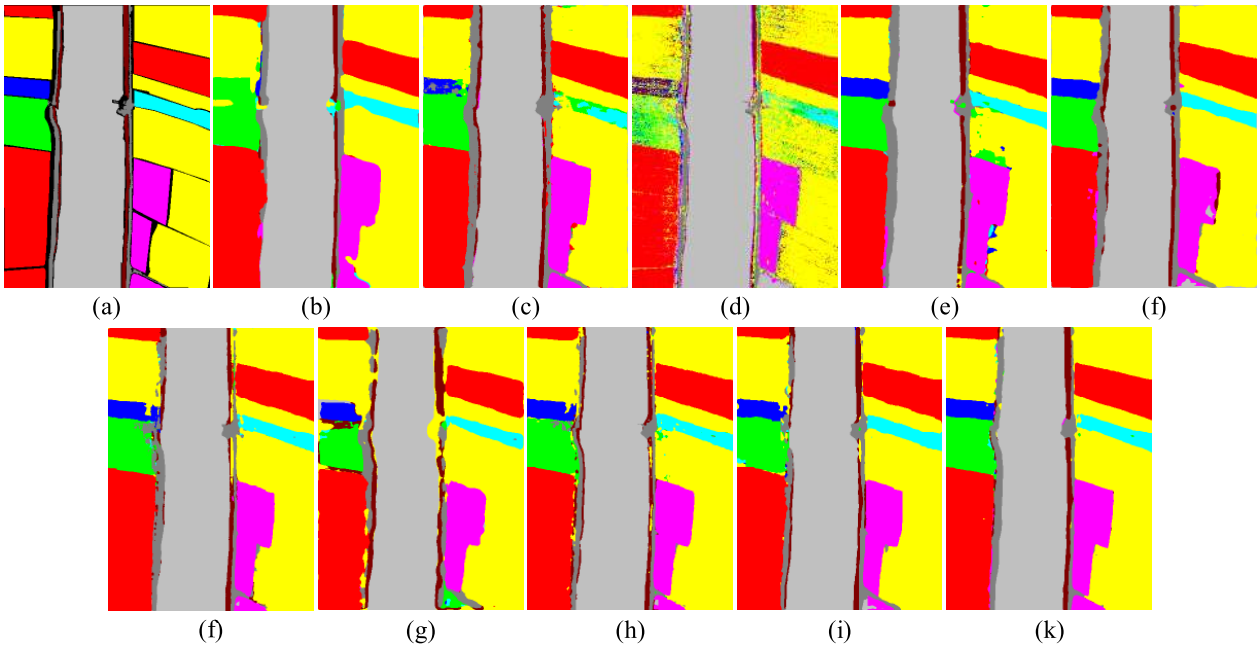


Fig. 15. Classification maps obtained by different methods on the LK dataset. (a)–(k) Ground truth, HybridSN, DTAN, GNet, SF, SSFTT, CEGCN, FDGC, WFCG, AMGCFN, and proposed.

method exhibits the fewest misclassifications on both datasets compared to DTAN, SSFTT, and WFCG.

Specifically, on the IP dataset, DTAN, SSFTT, and WFCG suffer from significant confusion in multiple classes. For instance, DTAN misclassifies 8% of samples in class 7 as class 5. SSFTT exhibits a more complex situation, with 18% misclassifications in both classes 9 and 12. Although WFCG shows some improvement, it still misclassifies 9% of samples in class 12. In contrast, the proposed method has significantly fewer misclassifications, with only 4% misclassifications in the most confused class 4.

Similarly, on the UP dataset, DTAN and SSFTT have more severe confusion, especially DTAN with 28% misclassifications in class 8. While WFCG shows some improvement, it still misclassifies 4% of samples in class 8. The proposed method consistently achieves the lowest confusion rate on the UP dataset.

These experimental results strongly support the superior classification accuracy of the proposed method. Compared to other methods, our method can more effectively distinguish between different classes, especially demonstrating stronger robustness in scenarios with highly similar or overlapping classes.

H. Comparison of Robustness

Spectral data often suffer from various degradations, noise, and variations during the imaging process, which can significantly degrade data quality and subsequently impact the performance of classifiers [59], [60], [61]. To verify the robustness of the proposed method in noisy environments, we added different proportions of Gaussian noise to the IP and UP datasets. Experimental results shown in Fig. 18 demonstrate that our proposed method exhibits stronger robustness in terms of three evaluation metrics (OA, AA, and $\kappa \times 100$).

Specifically, as the level of Gaussian noise increases, the performance of most comparison methods fluctuates significantly, especially on the IP dataset. For example, the performance of the CEGCN method drops significantly after introducing 20% Gaussian noise, with the three metrics decreasing by approximately 10%, 20%, and 10%, respectively. In contrast, our method maintains high accuracy

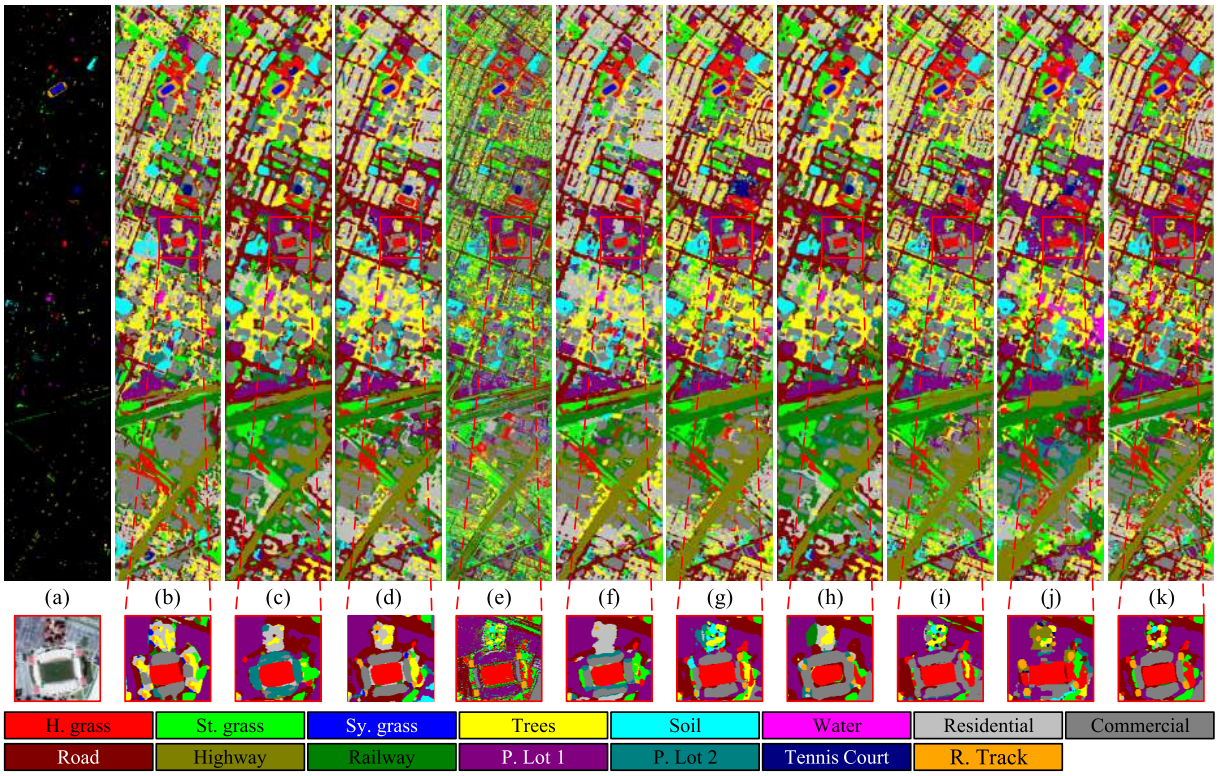


Fig. 16. Classification maps obtained by different methods on the LK dataset. (a)–(k) Ground truth, HybridSN, DTAN, GNet, SF, SSFTT, CEGCN, FDGC, WFCG, AMGCFN, and proposed.

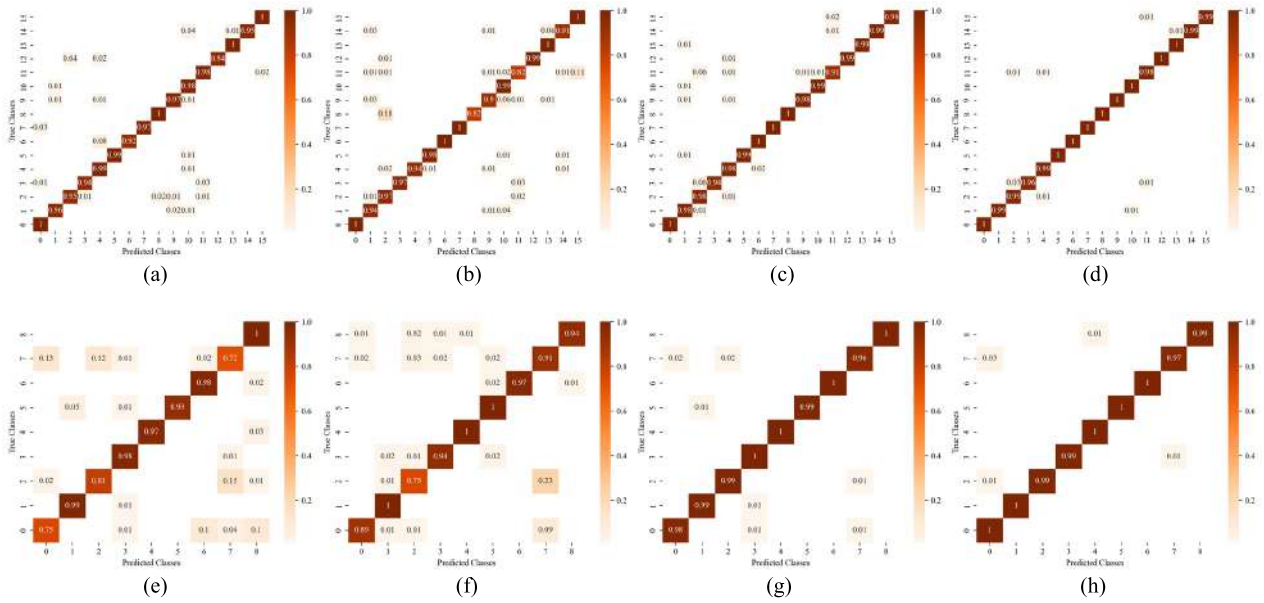


Fig. 17. Comparison of confusion matrices of different methods on IP and UP datasets (from top to bottom). (a) and (e) DTAN, (b) and (f) SSFTT, (c) and (g) WFCG, and (d) and (h) proposed.

even at the same noise level, demonstrating its robustness to noise. Similarly, on the UP dataset, the performance of DTAN and SSFTT declines more significantly. Especially for SSFTT, the decrease in the AA metric after introducing 20% noise is much larger than that of DTAN. Our proposed method also exhibits strong robustness on the UP dataset, with less impact from noise.

Experimental results show that the proposed method exhibits stronger robustness in complex noisy environments, effectively suppressing the impact of noise on classification results and maintaining high classification accuracy.

I. Comparison of Heatmaps

To visualize the roles of the proposed GSA and MAF modules more intuitively, we conducted heatmap visualizations on the IP and UP datasets, as shown in Fig. 18.

Fig. 19(a) and (e) presents the heatmaps of a model containing only a single-branch depthwise separable convolution.

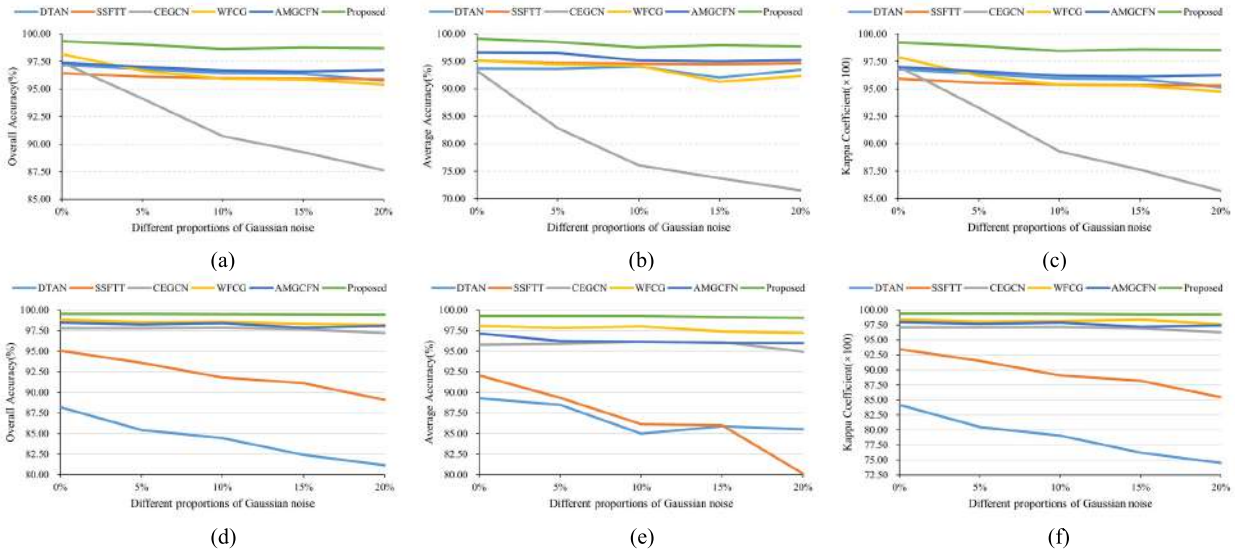


Fig. 18. Comparison of the robustness of different metrics on the IP and UP datasets (from top to bottom). (a) and (d) OA, (b) and (e) AA, and (c) and (f) $\kappa \times 100$.

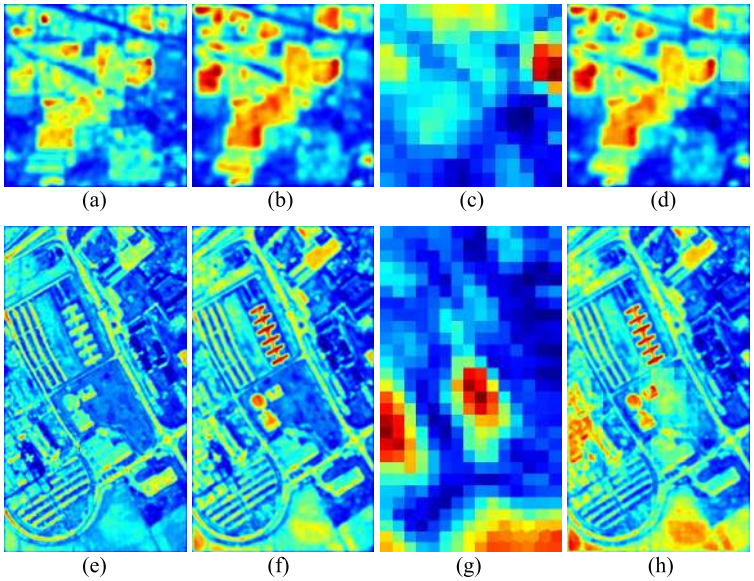


Fig. 19. Comparison of heatmaps of different modules on the IP and UP datasets (from top to bottom). (a) and (e) Depthwise separable convolution, (b) and (f) MAF, (c) and (g) GSA, and (d) and (h) GSA + MAF.

It can be observed that the model has a weaker ability to capture local details, and the heatmap responses are relatively sparse. Fig. 19(b) and (f) shows the heatmaps of the pixel-based MAF branch. It is evident that MAF excels in extracting local detail features, with richer heatmap responses. This indicates that the MAF module is highly effective in extracting fine-grained features.

Fig. 19(c) and (g) presents the heatmaps of the GSA branch. The two heatmaps show some block effects because they are the results of ablation experiments conducted using the proposed GSA alone. The GSA branch takes superpixels as its basic units. Consequently, the heatmaps are presented at the superpixel level, visually appearing as larger pixel blocks. Moreover, it can be observed that the heatmaps lack many local details and reflect an overview of the features. This demonstrates that the GSA module is more adept at extracting coarse-grained information but falls short in capturing local features. The introduction of the MAF module is precisely to compensate for the deficiency of the GSA module in local feature extraction.

Fig. 19(d) and (h) shows the heatmaps of the fused GSA and MAF modules. It can be observed that the fused model not only captures rich local details but also exhibits stronger heatmap responses in regions with labeled samples. This fully demonstrates that the fusion of GSA and MAF modules can effectively extract features and improve the model's representational capacity.

IV. Conclusion

To alleviate the problem of limited samples in HSIC, this article proposes a GS-GraphSAT. This method deeply explores the correlations between graph nodes through the

GSA mechanism and effectively extracts local detail features of the image using the MAF module. In addition, the introduction of GTS can timely supplement samples during the training process, improving the efficiency of sample utilization. The experimental results on multiple datasets show that the proposed method outperforms some state-of-the-art methods in terms of classification accuracy and robustness. Although this method has shown good performance, its complexity still needs to be optimized. Future research will focus on exploring lighter network architectures to reduce model complexity while ensuring performance.

REFERENCES

[1] P. Ghamisi et al., "Advances in hyperspectral image and signal processing: A comprehensive overview of the state of the art," *IEEE Geosci. Remote Sens. Mag.*, vol. 5, no. 4, pp. 37–78, Dec. 2017.

[2] G. Zhan, Y. Uwamoto, and Y.-W. Chen, "HyperUNet for medical hyperspectral image segmentation on a choledochal database," in *Proc. IEEE Int. Conf. Consum. Electron. (ICCE)*, Las Vegas, NV, USA, Jan. 2022, pp. 1–5.

[3] Y. Li, R. Wu, Q. Tan, Z. Yang, and H. Huang, "Masked spectral bands modeling with shifted windows: An excellent self-supervised learner for classification of medical hyperspectral images," *IEEE Signal Process. Lett.*, vol. 30, pp. 543–547, 2023.

[4] L. Zhang, M. Zhang, J. Huang, C. Zhang, F. Ye, and W. Pan, "A new approach for mineral mapping using drill-core hyperspectral image," *IEEE Geosci. Remote Sens. Lett.*, vol. 20, pp. 1–5, 2023.

[5] L. Chen, K. Tan, X. Wang, and C. Pan, "Estimation soil organic matter using airborne hyperspectral imagery," in *Proc. 13th Workshop Hyperspectral Imag. Signal Process., Evol. Remote Sens. (WHISPERS)*, Athens, Greece, Nov. 2023, pp. 1–5.

[6] M. Shimoni, R. Haelterman, and C. Perneel, "Hypersectral imaging for military and security applications: Combining myriad processing and sensing techniques," *IEEE Geosci. Remote Sens. Mag.*, vol. 7, no. 2, pp. 101–117, Jun. 2019.

[7] S. Li, W. Song, L. Fang, Y. Chen, P. Ghamisi, and J. A. Benediktsson, "Deep learning for hyperspectral image classification: An overview," *IEEE Trans. Geosci. Remote Sens.*, vol. 57, no. 9, pp. 6690–6709, Sep. 2019.

[8] R. Elkadiri et al., "A remote sensing-based approach for debris-flow susceptibility assessment using artificial neural networks and logistic regression modeling," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 7, no. 12, pp. 4818–4835, Dec. 2014.

[9] M. H. R. Sales, S. de Bruin, C. Souza, and M. Herold, "Land use and land cover area estimates from class membership probability of a random forest classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 60, 2022, Art. no. 4402711.

[10] M. Sheykhmousa, M. Mahdianpari, H. Ghanbari, F. Mohammadimanesh, P. Ghamisi, and S. Homayouni, "Support vector machine versus random forest for remote sensing image classification: A meta-analysis and systematic review," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 13, pp. 6308–6325, 2020.

[11] J. Peng et al., "Low-rank and sparse representation for hyperspectral image processing: A review," *IEEE Geosci. Remote Sens. Mag.*, vol. 10, no. 1, pp. 10–43, Mar. 2022.

[12] L. Zhuang, L. Gao, B. Zhang, X. Fu, and J. M. Bioucas-Dias, "Hyperspectral image denoising and anomaly detection based on low-rank and sparse representations," *IEEE Trans. Geosci. Remote Sens.*, vol. 60, 2022, Art. no. 5500117.

[13] J. Feng, Z. Gao, R. Shang, X. Zhang, and L. Jiao, "Multi-complementary generative adversarial networks with contrastive learning for hyperspectral image classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 61, 2023, Art. no. 5520018.

[14] J. Feng et al., "Class-aligned and class-balancing generative domain adaptation for hyperspectral image classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 62, 2024, Art. no. 5509617.

[15] D. Hong et al., "SpectralGPT: Spectral remote sensing foundation model," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 46, no. 8, pp. 5227–5244, Aug. 2024.

[16] J. Yao, D. Hong, C. Li, and J. Chanussot, "SpectralMamba: Efficient mamba for hyperspectral image classification," 2024, *arXiv:2404.08489*.

[17] D. Hong et al., "Cross-city matters: A multimodal remote sensing benchmark dataset for cross-city semantic segmentation using high-resolution domain adaptation networks," *Remote Sens. Environ.*, vol. 299, Dec. 2023, Art. no. 113856.

[18] C. Shi, T. Wang, and L. Wang, "Branch feature fusion convolution network for remote sensing scene classification," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 13, pp. 5194–5210, 2020.

[19] A. Vaswani et al., "Attention is all you need," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 30, 2017, pp. 1–11.

[20] C. Li et al., "CasFormer: Cascaded transformers for fusion-aware computational hyperspectral imaging," *Inf. Fusion*, vol. 108, Aug. 2024, Art. no. 102408.

[21] Z. Zhang, P. Cui, and W. Zhu, "Deep learning on graphs: A survey," *IEEE Trans. Knowl. Data Eng.*, vol. 34, no. 1, pp. 249–270, Jan. 2022.

[22] Y. Ding et al., "Multi-scale receptive fields: Graph attention neural network for hyperspectral image classification," *Expert Syst. Appl.*, vol. 223, Aug. 2023, Art. no. 119858.

[23] A. Yang, M. Li, Y. Ding, D. Hong, Y. Lv, and Y. He, "GTFN: GCN and transformer fusion network with spatial–spectral features for hyperspectral image classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 61, 2023, Art. no. 6600115.

[24] Z. Zhong, J. Li, Z. Luo, and M. Chapman, "Spectral–spatial residual network for hyperspectral image classification: A 3-D deep learning framework," *IEEE Trans. Geosci. Remote Sens.*, vol. 56, no. 2, pp. 847–858, Feb. 2018.

[25] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Las Vegas, NV, USA, Jun. 2016, pp. 770–778.

[26] S. K. Roy, G. Krishna, S. R. Dubey, and B. B. Chaudhuri, "HybridSN: Exploring 3-D-2-D CNN feature hierarchy for hyperspectral image classification," *IEEE Geosci. Remote Sens. Lett.*, vol. 17, no. 2, pp. 277–281, Feb. 2020.

[27] J. Yao et al., "Semi-active convolutional neural networks for hyperspectral image classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 60, 2022, Art. no. 5537915.

[28] J. Hu, L. Shen, and G. Sun, "Squeeze-and-excitation networks," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Salt Lake City, UT, USA, Jun. 2018, pp. 7132–7141.

[29] J. Fu, J. Liu, J. Jiang, Y. Li, Y. Bao, and H. Lu, "Scene segmentation with dual relation-aware attention network," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 32, no. 6, pp. 2547–2560, Jun. 2021.

[30] X. Jin, T. Sun, W. Chen, H. Ma, Y. Wang, and Y. Zheng, "Parameter adaptive non-model-based state estimation combining attention mechanism and LSTM," *IECE Trans. Intell. Systematics*, vol. 1, no. 1, pp. 40–48, 2024.

[31] G. Zhou, S. Bu, and T. Kirubarajan, "Simultaneous spatiotemporal bias compensation and data fusion for asynchronous multisensor systems," *Chin. J. Inf. Fusion*, vol. 1, no. 1, pp. 16–32, 2024.

[32] H. Ren, Y. Wang, and H. Ma, "Deep prediction network based on covariance intersection fusion for sensor data," *IECE Trans. Intell. Systematics*, vol. 1, no. 1, pp. 10–18, 2024.

[33] X. Guo, F. Yang, and L. Ji, "A mimic fusion algorithm for dual channel video based on possibility distribution synthesis theory," *Chin. J. Inf. Fusion*, vol. 1, no. 1, pp. 33–49, 2024.

[34] Y. Cui, Z. Yu, J. Han, S. Gao, and L. Wang, "Dual-triple attention network for hyperspectral image classification using limited training samples," *IEEE Geosci. Remote Sens. Lett.*, vol. 19, pp. 1–5, 2022.

[35] L. Liang, S. Zhang, J. Li, A. Plaza, and Z. Cui, "Multi-scale spectral–spatial attention network for hyperspectral image classification combining 2D octave and 3D convolutional neural networks," *Remote Sens.*, vol. 15, no. 7, p. 1758, Mar. 2023.

[36] G. Huang, Z. Liu, L. Van Der Maaten, and K. Q. Weinberger, "Densely connected convolutional networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Honolulu, HI, USA, Jul. 2017, pp. 2261–2269.

[37] C. Shi, D. Liao, T. Zhang, and L. Wang, "Hyperspectral image classification based on expansion convolution network," *IEEE Trans. Geosci. Remote Sens.*, vol. 60, 2022, Art. no. 5528316.

[38] J. Bai et al., "Few-shot hyperspectral image classification based on adaptive subspaces and feature transformation," *IEEE Trans. Geosci. Remote Sens.*, vol. 60, 2022, Art. no. 5523917.

[39] X. Wu, D. Hong, and J. Chanussot, "Convolutional neural networks for multimodal remote sensing data classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 60, 2022, Art. no. 5517010.

[40] Z. Chen, D. Hong, and H. Gao, "Grid network: Feature extraction in anisotropic perspective for hyperspectral image classification," *IEEE Geosci. Remote Sens. Lett.*, vol. 20, pp. 1–5, 2023.

[41] A. Dosovitskiy et al., "An image is worth 16×16 words: Transformers for image recognition at scale," 2020, *arXiv:2010.11929*.

[42] L. Yuan et al., "Tokens-to-token ViT: Training vision transformers from scratch on ImageNet," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Montreal, QC, Canada, Oct. 2021, pp. 538–547.

[43] Z. Liu et al., "Swin transformer: Hierarchical vision transformer using shifted windows," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Montreal, QC, Canada, Oct. 2021, pp. 9992–10002.

[44] D. Hong etal., "SpectralFormer: Rethinking hyperspectral image classification with transformers," *IEEE Trans. Geosci. Remote Sens.*, vol. 60, 2022, Art. no. 5518615.

[45] L. Sun, G. Zhao, Y. Zheng, and Z. Wu, "Spectral-spatial feature tokenization transformer for hyperspectral image classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 60, 2022, Art. no. 5522214.

[46] C. Shi, S. Yue, and L. Wang, "A dual branch multiscale Transformer network for hyperspectral image classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 62, 2024, Art. no. 5504520.

[47] Y. Ding et al., "Self-supervised locality preserving low-pass graph convolutional embedding for large-scale hyperspectral image clustering," *IEEE Trans. Geosci. Remote Sens.*, vol. 60, 2022, Art. no. 5536016.

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

[49] D. Yao et al., "Deep hybrid: Multi-graph neural network collaboration for hyperspectral image classification," *Defence Technol.*, vol. 23, pp. 164–176, May 2023.

[50] Q. Liu, L. Xiao, J. Yang, and Z. Wei, "CNN-enhanced graph convolutional network with pixel- and superpixel-level feature fusion for hyperspectral image classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 59, no. 10, pp. 8657–8671, Oct. 2021.

[51] Q. Liu, Y. Dong, Y. Zhang, and H. Luo, "A fast dynamic graph convolutional network and CNN parallel network for hyperspectral image classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 60, 2022, Art. no. 5530215.

[52] Y. Ding, X. Zhao, Z. Zhang, W. Cai, N. Yang, and Y. Zhan, "Semi-supervised locality preserving dense graph neural network with ARMA filters and context-aware learning for hyperspectral image classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 60, 2022, Art. no. 5511812.

[53] H. Zhou, F. Luo, H. Zhuang, Z. Weng, X. Gong, and Z. Lin, "Attention multihop graph and multiscale convolutional fusion network for hyperspectral image classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 61, 2023, Art. no. 5508614.

[54] Y. Dong, Q. Liu, B. Du, and L. Zhang, "Weighted feature fusion of convolutional neural network and graph attention network for hyperspectral image classification," *IEEE Trans. Image Process.*, vol. 31, pp. 1559–1572, 2022.

[55] C. Shi, H. Wu, and L. Wang, "CEGAT: A CNN and enhanced-GAT based on key sample selection strategy for hyperspectral image classification," *Neural Netw.*, vol. 168, pp. 105–122, Nov. 2023.

[56] Z. Chen, G. Wu, H. Gao, Y. Ding, D. Hong, and B. Zhang, "Local aggregation and global attention network for hyperspectral image classification with spectral-induced aligned superpixel segmentation," *Expert Syst. Appl.*, vol. 232, Dec. 2023, Art. no. 120828.

[57] G. Licciardi, P. R. Marpu, J. Chanussot, and J. A. Benediktsson, "Linear versus nonlinear PCA for the classification of hyperspectral data based on the extended morphological profiles," *IEEE Geosci. Remote Sens. Lett.*, vol. 9, no. 3, pp. 447–451, May 2012.

[58] Y.-J. Liu, C.-C. Yu, M.-J. Yu, and Y. He, "Manifold SLIC: A fast method to compute content-sensitive superpixels," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Las Vegas, NV, USA, Jun. 2016, pp. 651–659.

[59] D. Hong, N. Yokoya, J. Chanussot, and X. X. Zhu, "An augmented linear mixing model to address spectral variability for hyperspectral unmixing," *IEEE Trans. Image Process.*, vol. 28, no. 4, pp. 1923–1938, Apr. 2019.

[60] D. Hong, J. Yao, C. Li, D. Meng, N. Yokoya, and J. Chanussot, "Decoupled-and-coupled networks: Self-supervised hyperspectral image super-resolution with subpixel fusion," *IEEE Trans. Geosci. Remote Sens.*, vol. 61, 2023, Art. no. 5527812.

[61] C. Li, B. Zhang, D. Hong, X. Jia, A. Plaza, and J. Chanussot, "Learning disentangled priors for hyperspectral anomaly detection: A coupling model-driven and data-driven paradigm," *IEEE Trans. Neural Netw. Learn. Syst.*, early access, Jun. 4, 2024, doi: 10.1109/TNNLS.2024.3401589.

Fei Zhu received the bachelor's degree from Luoyang Institute of Science and Technology, Luoyang, China, in 2021. He is currently pursuing the master's degree with Qiqihar University, Qiqihar, China.

His research interests include hyperspectral image processing and machine learning.

Cuiping Shi (Member, IEEE) received the M.S. degree from Yangzhou University, Yangzhou, China, in 2007, and the Ph.D. degree from Harbin Institute of Technology (HIT), Harbin, China, in 2016.

From 2017 to 2020, she was a Post-Doctoral Researcher with the College of Information and Communications Engineering, Harbin Engineering University, Harbin. She is a Professor with the Department of Communication Engineering, Qiqihar University, Qiqihar, China. She works with the College of Information Engineering, Huzhou University, Huzhou, China. She has published two academic books about remote sensing image processing and more than 90 papers in journals and conference proceedings. Her main research interests include remote sensing image processing, pattern recognition, and machine learning.

Dr. Shi's doctoral dissertation won the Nomination Award of Excellent Doctoral Dissertation of Harbin University of Technology (HIT) in 2016.

Liguo Wang (Member, IEEE) received the M.S. and Ph.D. degrees in signal and information processing from Harbin Institute of Technology, Harbin, China, in 2002 and 2005, respectively.

From 2006 to 2008, he held a post-doctoral position at Harbin Engineering University, Harbin. From 2020, he worked at the College of Information and Communication Engineering, Dalian Nationalities University, Dalian, China. He has published two books about hyperspectral image processing and more than 130 papers in journals and conference proceedings. His main research interests include remote sensing image processing.

Kaijie Shi received the bachelor's degree from Heilongjiang University of Science and Technology, Harbin, China, in 2021. He is currently pursuing the master's degree with Qiqihar University, Qiqihar, China.

His research interests include remote sensing image compression and machine learning.

SCI 收录报告

经查 Web of Science-Core Collection ，石翠萍提供的如下文章已经被 SCI-Expanded（科学引文索引）收录，其收录记录简要信息摘选如下：

标题：A Greedy Strategy Guided Graph Self-Attention Network for Few-Shot Hyperspectral Image Classification
作者：Zhu, F (Zhu, Fei); Shi, CP (Shi, Cuiping); Wang, LG (Wang, Liguo); Shi, KJ (Shi, Kaijie)
来源出版物：IEEE TRANSACTIONS ON GEOSCIENCE AND REMOTE SENSING 卷：62
文献号：5539620 DOI: 10.1109/TGRS.2024.3505539 Published Date: 2024
Web of Science 核心合集中的 "被引频次": 0
被引频次合计：0
入藏号：WOS:001373843000013
语言：English
文献类型：Article
地址：[Zhu, Fei; Shi, Kaijie] Qiqihar Univ, Dept Commun Engn, Qiqihar 161000, Peoples R China.
[Shi, Cuiping] Huzhou Univ, Coll Informat Engn, Huzhou 313000, Peoples R China.
[Wang, Liguo] Dalian Nationalities Univ, Coll Informat & Commun Engn, Dalian 116000, Peoples R China.
通讯作者地址：Shi, CP (通讯作者)，Huzhou Univ, Coll Informat Engn, Huzhou 313000, Peoples R China.
电子邮件地址：2022935750@qqhru.edu.cn; shicuiping@zjhu.edu.cn; wangliguo@hrbeu.edu.cn; 2022910313@qqhru.edu.cn
Affiliations: Qiqihar University; Huzhou University; Dalian Minzu University
ISSN: 0196-2892
eISSN: 1558-0644
来源出版物页码计数：20

特此证明

湖州师范学院图书馆

查证人：钱绍

2024 年 12 月 24 日