Dual-Graph Convolutional Network based on Band Attention and Sparse Constraint for Hyperspectral Band Selection

Jie Feng[a], Zhanwei Ye[a], Shuai Liu[b], Xiangrong Zhang[a], Jiantong Chen[a], Ronghua Shang[a], Licheng Jiao[a]

[a]*School of Artificial Intelligence, Xidian University, Xi'an 710071, Shaanxi, China*
[b]*School of Software Engineering, Xi'an Jiaotong University, Xi'an 710049, Shaanxi, China*

Abstract

Band selection is a research hotspot in hyperspectral image processing. The continuity of the spectral bands causes the adjacent bands to be highly correlated, and correlation among long-range bands is possible with hundreds of spectral bands. Most existing deep learning methods fail to make full use of the inter-band correlation for band selection. In this paper, a novel dual-graph convolutional network based on band attention and a sparse constraint is proposed for band selection. The network consists of two branches. In the attention branch, band-based dual graphs are constructed to encode the contextual correlation of adjacent bands and the structural correlation of long-range bands into non-Euclidean space. Subsequently, the graph convolution-based band attention mechanism is devised to aggregate the band information in the band-based dual graphs and to generate the attention map for all bands. The band attention map is sparsely constrained and embedded as a mask into the trunk branch. In the trunk branch, sample-based dual graphs are constructed to represent the topological information of the samples in the spectral and spatial domains. Furthermore, a dense graph convolutional network is designed to extract and fuse the spatial–spectral and topological features from the shallow to deep layers for classification. A soft-shifting optimization strategy is implemented by defining a new loss from full bands and selected bands to solve the optimization problem caused by the sparse constraint. In this manner, band selection, feature extraction, and classification can be combined into an end-to-end trainable network. The experimental results on representative hyperspectral image datasets demonstrate the superiority of the proposed method over current state-of-the-art band selection methods.

Key words: Graph convolutional network, Band selection, Hyperspectral image classification, Attention mechanism

1. Introduction

In recent years, the rapid development of remote sensing technology has provided strong support for exploring and observing the surface space of the earth. As an important branch of the remote sensing field, hyperspectral imaging has received increasing attention from researchers. Hyperspectral images (HSIs) offer an irreplaceable advantage in land-cover discrimination owing to their high spectral resolution. They play a significant role in many fields, such as military [1], astronomy [2], agriculture [3], and mineralogy [4].

Although the abundant spectral bands of HSIs provide rich information for land-cover discrimination, irrelevant and redundant spectral bands limit further applications, such as classification [5] and target detection [6]. Therefore, feature selection and feature extraction are considered as crucial steps in HSI processing [7].

Feature extraction methods can be divided into three categories according to whether the label information is used: supervised, unsupervised, and semi-supervised methods. Supervised methods use the label information to extract low-dimensional discriminative features by mapping the original high-dimensional space. Representative methods include linear discriminant analysis [8], local Fisher discriminant analysis [9], and supervised Laplacian eigenmaps [10]. Unsupervised methods can implement a dimensionality-reduction model without any label information. Principal component analysis (PCA) [11] is a widely used unsupervised method and its variants have also been developing rapidly. SuperPCA [12] and superpixel kernel PCA [13] have been proposed by introducing superpixel

segmentation into PCA-based methods to enable the local spatial information to be used. Recently, spectral–spatial and SuperPCA (S^3-PCA) [14] was developed to expand the spatial–spectral feature extraction capability of Super PCA. Semi-supervised methods use labeled and unlabeled samples simultaneously. In [15] and [16], semi-supervised discriminant analysis and semi-supervised local discriminant analysis were proposed based on the further development of linear discriminant analysis. As a widely applied theory in semi-supervised learning, graph learning theory has also been applied for semi-supervised feature extraction, such as semi-supervised graph learning [17] and semi-supervised discriminant hypergraph learning [18].

Feature selection is known as band selection in HSI processing [19]. The aim of band selection is to search for and select the most discriminative or informative spectral bands to represent the overall spectral information. In contrast to feature extraction, band selection can maintain the integrity of the original band structure and prevent damage to the physical information [20], which makes it more appropriate for specific HSI applications.

Existing band selection methods include three main categories: filter, wrapper, and embedded [21], [22]. Filter methods implement feature selection through a pre-designed criterion, which can be regarded as the preprocessing procedure that is independent of the chosen classifier [23]. This type of method is efficient and can be flexibly combined with various classifiers. However, the pre-designed criterion is independent of the chosen classifier, which limits the classification performance. Representative filter methods include the minimal-redundancy maximal-relevance (mRMR) framework [24] and hypergraph model (HM) [25]. Wrapper methods evaluate the performance of candidate band subsets according to their classification abilities under the chosen classifier, which may be the support vector machine (SVM) [26], extreme learning machine (ELM) [27], or convolutional neural network (CNN) [28]. In wrapper methods, the same type of classifier is used for both the evaluation and classification of the selected bands. Thus, wrapper methods are more likely to achieve better classification performance than filter methods. However, the classifier in wrapper methods needs to be retrained on each candidate band subset, which requires excessive time. Embedded methods embed feature selection into the training of the chosen classifier [29], [30]. In embedded methods, the final selected band subset can be obtained while the classifier training is completed. Therefore, embedded methods are generally more efficient than wrapper methods. Moreover, the combined optimization of band selection and classification enables embedded methods to outperform filter methods in most cases. Band selection methods can be divided into numerous categories according to different criteria. In addition to filter, wrapper, and embedded division, band selection can also be divided into supervised [31], unsupervised [32], and semi-supervised [33] methods based on the use of labeled samples. Unsupervised methods can be further divided into clustering-based methods [34], [35], [36] and ranking-based methods [37] according to the search strategies.

In the past decade, deep neural network methods have achieved unprecedented progress in a plethora of domains. The CNN is one of the most representative deep learning methods [38], which has been actively applied in the band selection and classification of HSIs [39]. In several early applications for band selection, CNN was employed as the classifier and the search for candidate bands was performed by traditional methods [28]. With the extension of related research, CNNs have been used in embedded methods by producing sparse weights to determine the selected bands [40], [41]. Although many existing deep learning methods have been proposed for HSI band selection, several problems remain that have not been solved effectively, as follows: 1) Most deep learning-based methods fail to integrate the non-local correlation among all spectral bands, which may reduce the evaluation effectiveness of the band weights and result in a loss of discriminative bands. 2) These methods make the network parameters sparse to remove the redundant and noisy bands by using threshold-based functions or regularization terms, which may lead to an intractable optimization problem. 3) The performance of deep learning methods is strongly dependent on the quantity of the training samples. Furthermore, the contradiction between limited training samples in HSIs and the sample requirements of deep learning methods restricts the band selection performance.

In recent years, the graph convolutional network (GCN) [42] was proposed to deal with non-Euclidean structure data effectively by modeling the correlations among samples, which enables the GCN to overcome the limitation of the fixed receptive field in the CNN. Owing to its flexible feature extraction ability, the GCN has been successfully applied in HSI classification, such as multi-scale dynamic GCN [43], miniGCN [44], and context-aware GCN [45]. In [44], miniGCN was proposed by using minibatch samples to reduce the computational cost, which is effective for dealing with large graphs. In existing GCN-based HSI classification methods, the correlation among samples in the local and global regions

has been modeled effectively. However, learning the correlation among bands with GCN requires further investigation.

In this paper, a new dual-GCN based on band attention and sparse constraint (BSD-GCN) is proposed for the band selection of HSIs. The BSD-GCN consists of an attention branch and a trunk branch. In the attention branch, band-based spatial and spectral graphs are constructed using both labeled and unlabeled samples in non-Euclidean space. Graph convolution is implemented to extract the information of adjacent and long-range correlated spectral bands in this non-Euclidean space effectively. The band attention mechanism aggregates and propagates the band information to generate the band attention map automatically. This attention map is sparsely constrained by maintaining the top weights and forcing the remaining weights to zero so as to remove the noisy and redundant bands. Subsequently, it is expanded into a column sparse mask and inserted into the first layer of the trunk branch. In the trunk branch, sample-based spatial and spectral graphs are established according to the sparse attention map. Several spatial–spectral graph convolution modules are stacked in the trunk branch with dense connections, which fuses the spatial–spectral and topological features from different layers for classification. Soft-shifting optimization is implemented to update these bands with the assistance of all spectral bands to solve the problem of several unselected bands not being updated owing to the sparse constraint.

The main contributions in this paper are listed as follows:

- The BSD-GCN can make full use of the local and non-local correlations among spectral bands by constructing two types of band-based graphs and aggregating the band information with the topological structure. This provides a more comprehensive evaluation of the weight of each spectral band.
- A new soft-shifting optimization method is devised to solve the optimization problem caused by the sparsity of the band selection. It provides the gradients for updating the parameters of unselected bands with the loss of the full bands and results in optimizing band selection, feature extraction, and classification in an end-to-end, trainable process.
- To alleviate the contradiction between limited samples and deep learning model requirements, semi-supervised graph construction is implemented by leveraging limited labeled samples and numerous unlabeled samples. The samples are divided into different mini batches for graph construction and network training to further improve the efficiency and generalization ability.

2. Related Work

In this section, we introduce the three above-mentioned categories of band selection methods in detail and address deep learning-based methods in a separate paragraph to provide a comprehensive understanding.

Filter methods: The key to filter methods is the design of the evaluation criterion. Peng et al. [24] proposed the mRMR to maximize the dependency of the features with the target classes and to minimize the redundancy among features. In fact, only the redundancy that is correlated with the target classes needs to be minimized for the classification task. In [46], an adaptive subspace partition strategy was designed to divide all of the spectral bands into partitions gradually, according to the interclass and intraclass distances. Thereafter, candidate bands with minimal noise were selected in each partition. These two methods are efficient in selecting the band subset. In [25], the HM was proposed by establishing a hypergraph based on the spatial and spectral information of the samples. The band selection of HSIs is formalized as a sparse group-constrained linear regression problem on the hypergraph. The HM can use additional spatial–spectral information of HSIs, which is helpful for band selection. However, the construction of a hypergraph is very time consuming.

Wrapper methods: In [27], the band selection of HSIs was converted into the dynamic selection of ELM classifiers, which is abbreviated as DCS. In DCS, pseudo-labels are produced for the unlabeled samples. Thereafter, the band subset with the best classifier is selected using both the real and pseudo-labels. Although an ELM classifier can be trained rapidly, numerous classifiers cause DCS to require a long time in the training phase. In [26], feature selection was implemented using fractional-order Darwinian particle swarm optimization (FODPSO) and the SVM. In FODPSO, the classification result of the SVM is used as its fitness value. FODPSO enables this model to select the most discriminative bands automatically.

Embedded methods: Embedded methods usually combine band selection and classification in one process. In [29], a SVM based on recursive feature elimination (RFE-SVM) was proposed by maximizing the SVM margin in a sequential backward selection manner. Subsequently, a modified recursive SVM

(MR-SVM) [30] was proposed based on the RFE-SVM, which eliminates the features by considering the mean values of the features from different classes.

Deep learning methods: Owing to the powerful feature learning ability, deep learning-based methods have received increasing attention in the band selection field. In [19], an unsupervised CNN-based method was designed to transform band selection into a spectral reconstruction task. In [28], the self-improving CNN (SICNN) was proposed, which uses FODPSO to search the band subsets and uses a CNN as the evaluator for candidate band subsets in FODPSO. In SICNN, although the CNN can evaluate the performance of the band subsets effectively, it needs to be well trained for each candidate band subset, which is time consuming. To accelerate the band selection, another CNN based on distance density (DDCNN) was proposed [47]. In DDCNN, a one-dimensional CNN (1DCNN) is well trained with full-band HSI data. Thereafter, the distance density among all of the spectral bands is measured to select the candidate band subsets, which are evaluated using the well-trained 1DCNN without retraining. The DDCNN avoids the time of retraining, but it is difficult for the fixed 1DCNN in the DDCNN to evaluate all the different band subsets effectively. In [40], a ternary weight CNN (TWCNN) used the ternary weight in a depth-wise convolution layer to determine whether or not each band was selected. The subsequent convolution layers extracted the features of the HSI data with the selected bands. The prosperity in this field has encouraged more innovative ideas to break through the bottleneck of band selection, such as reinforcement learning [48] and the attention mechanism [49], [41]. In [49], an attention-based CNN (ABCNN) was applied to explore the most informative bands with one-dimensional convolution. The ultimate band selection was modeled as an anomaly detection process. As the ultimate band selection process is separate from the classification network, the ABCNN may easily lose discriminative bands. In [41], another 2DCNN-based band attention method was proposed to learn the weight of each band in the HSIs. Thereafter, band selection was implemented by setting the weights of the unselected spectral bands to zero. The band attention network was optimized by ignoring the problem whereby unselected bands may not be updated, which restricts the selection of discriminative band subsets.

3. Background of Graph Convolutional Network

Graph neural networks (GNNs) [50] have been proven to be a powerful tool in handling non-Euclidean data, such as molecular properties [51] and social influence prediction [52]. With the introduction of the convolution operator into GNNs, convolutional GNNs have significantly promoted the development of GNNs. This approach covers two categories: spectral-based and spatial-based. As a representative spectral-based method, the GCN has received increasing attention owing to its efficiency and simplicity.

Spectral methods play an important role in the graph signal processing field because of their rigorous mathematical basis [53]. Given a signal $\boldsymbol{x} \in R^n$ on an undirected graph $G = \{V, E\}$, x_i is the i-th node, and V, E refer to the sets of nodes and edges. The normalized Laplacian matrix can be used as a representation of the graph G, which is defined as:

$$\boldsymbol{L} = \boldsymbol{I}_n - \boldsymbol{D}^{-\frac{1}{2}}\boldsymbol{A}\boldsymbol{D}^{-\frac{1}{2}} \tag{1}$$

where $\boldsymbol{I}_n$ is the identity matrix, $\boldsymbol{A}$ is the adjacency matrix of graph G, and $\boldsymbol{D}$ is the degree matrix calculated by $\boldsymbol{D}_{ii} = \sum_j \boldsymbol{A}_{ij}$. The spectral graph convolution of $\boldsymbol{x}$ with filter $\boldsymbol{g}_\theta = diag(\boldsymbol{\theta})$ ($\boldsymbol{\theta} \in R^n$) can be defined as follows:

$$\boldsymbol{x} *_G \boldsymbol{g}_\theta = \boldsymbol{U}\boldsymbol{g}_\theta\boldsymbol{U}^T\boldsymbol{x} \tag{2}$$

where $\boldsymbol{U} \in R^{n \times n}$ is the eigenvector matrix of $\boldsymbol{L}$, which is decomposed by $\boldsymbol{L} = \boldsymbol{U}\boldsymbol{\Lambda}\boldsymbol{U}^T$, and $\boldsymbol{\Lambda}$ is a diagonal matrix that is composed of eigenvalues of $\boldsymbol{L}$. The basic concept of (2) is to multiply the graph Fourier transform results of the signal and filter, and subsequently, to use an inverse transform to obtain the graph convolution result. Interested readers can refer to [54] for a detailed derivation and analysis. Although the above formula can realize spectral graph convolution, the high computational cost caused by the eigenvalue decomposition limits its practical application. In [55], ChebNet was proposed to approximate $\boldsymbol{g}_\theta$ with K-order Chebyshev polynomials. In 2017, the GCN [42] was proposed by further simplifying the assumptions on $\boldsymbol{g}_\theta$, and providing a simple and efficient graph convolution application. In the GCN, (2) becomes:

$$\boldsymbol{x} *_G \boldsymbol{g}_\theta = \boldsymbol{\theta}\left(\boldsymbol{I}_n + \boldsymbol{D}^{-\frac{1}{2}}\boldsymbol{A}\boldsymbol{D}^{-\frac{1}{2}}\right)\boldsymbol{x} \tag{3}$$

Based on (3), the propagation rule in the GCN is defined as:

$$\boldsymbol{H}^{l+1} = o\left(\overline{\boldsymbol{A}}\boldsymbol{H}^{l}\boldsymbol{W}^{l+1}\right)$$
$$\overline{\boldsymbol{A}} = \overline{\boldsymbol{D}}^{-\frac{1}{2}}\left(\boldsymbol{I}_n + \boldsymbol{A}\right)\overline{\boldsymbol{D}}^{-\frac{1}{2}} \tag{4}$$

where $\boldsymbol{H}^{l}$ and $\boldsymbol{H}^{l+1}$ represent the input and output of the l+1-th layer, $\boldsymbol{W}^{l+1}$ denotes the learnable parameters, $\overline{\boldsymbol{D}}_{ii} = \sum_j (\boldsymbol{I}_n + \boldsymbol{A})_{ij}$, and $o(\cdot)$ refers to the activation function. Equation (4) is the widely used graph convolution operation in the GCN and many GCN-based applications. It avoids complicated eigenvalue decomposition and significantly reduces the difficulty of deploying graph convolution. A visual example of graph convolution in a GCN is presented in Fig. 1 for an improved understanding of the graph convolution operation.

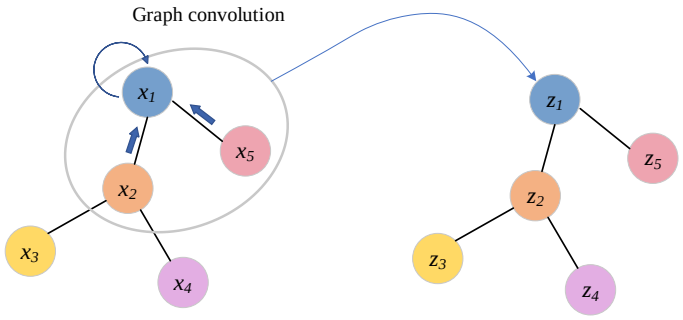

Fig. 1 Visual example of convolution operation on graph.

Fig. 1 depicts a five-node graph, in which each node is connected to numerous other nodes. When node x_1 is convolved, the features of node x_1 and its neighbor nodes x_2 and x_5 are summed and mapped into z_1. The other nodes are handled using the same concept. In this paper, this convolution operator is used to perform the band selection task.

4. Proposed BSD-GCN Method

The GCN can aggregate feature information from the nodes in neighborhoods and model long-range spatial correlations. Inspired by the GCN, an improved method, BSD-GCN, is proposed for the band selection of HSIs. The architecture of the BSD-GCN is illustrated in Fig. 2. As shown in Fig. 2, BSD-GCN constructs two branches: an attention branch for band selection, and a trunk branch for feature extraction and classification. The band-based and sample-based spatial and spectral graphs are constructed using the training samples from HSIs. The band-based spatial and spectral graphs are input into the attention branch, in which a dual-graph convolutional layer and an average pooling layer are stacked. An attention map is obtained after the average pooling layer. The attention map is sparsely constrained to remove redundant and noisy bands. In the trunk branch, the sample-based spatial and spectral graphs, which are constructed under the guidance of the sparse attention map, are used as the input. The trunk branch is composed of five densely connected spatial–spectral graph convolution modules. Each module is a dual-graph convolutional layer that extracts the spatial–spectral features for classification. Subsequently, the sparse attention map is multiplied into the input of the trunk branch to eliminate the values of the unselected bands. Finally, soft-shifting optimization is implemented to integrate the band selection, feature extraction, and classification processes in an end-to-end manner.

4.1. Band-Based and Sample-Based Dual-Graph Construction in HSIs

HSIs contain abundant information in the spectral and spatial domains simultaneously. In the spectral domain of HSIs, numerous spectral bands are acquired through dense sampling from the visible spectrum to the infrared spectrum. This mechanism not only leads to a high correlation between adjacent spectral bands, but also makes it possible for distant spectral bands to be relevant. In the spatial domain of HSIs, the samples that are located in the local region have a higher probability of belonging to the same category. However, samples in the same category may be distributed in various regions.

To make full use of the spatial–spectral characteristics of HSIs, two types of dual graphs are constructed in the BSD-GCN with the bands and samples as the nodes. In the band-based dual graphs, one of the graphs encodes the location correlation between the bands and the other represents the spectral

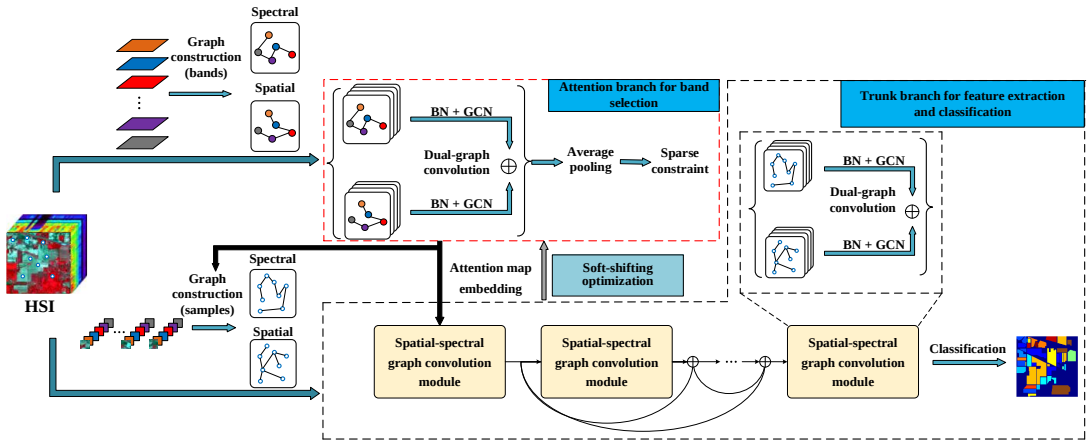

Fig. 2 Architecture of proposed BSD-GCN method.

similarity among the bands. In the sample-based dual graphs, one of the graphs reflects the spatial contextual correlation of the samples and the other expresses the spectral similarity among the samples. In the HSIs, the training set is defined as $\boldsymbol{X}^{tr}=\left\{\boldsymbol{x}_1^{tr},\boldsymbol{x}_2^{tr},...,\boldsymbol{x}_N^{tr}\right\}$. Each sample is a spectral vector of length B, where B is the number of spectral bands. The training set contains labeled and unlabeled samples. Inspired by miniGCN [26], minibatch learning is used in the BSD-GCN to reduce the computation and storage costs of large-scale graphs. A minibatch of labeled and unlabeled samples from the training samples are randomly selected as the input data. These samples are denoted by $\overline{\boldsymbol{X}}=\left\{\overline{\boldsymbol{x}}_1,\overline{\boldsymbol{x}}_2,...,\overline{\boldsymbol{x}}_{\overline{N}}\right\}$, where $\overline{N}$ is the number of samples in the current batch.

1) Band-based spatial and spectral graph construction: $G^{band-spa}=\{V^{band},E^{band-spa}\}$ and $G^{band-spe}=\{V^{band},E^{band-spe}\}$ refer to the band-based spatial and spectral graphs, respectively, where V^{band} represents the set of band nodes, and $E^{band-spa}$ and $E^{band-spe}$ represent the edge sets that are defined by the band position and spectral values. In $G^{band-spa}$ and $G^{band-spe}$, each band is regarded as a node. $\boldsymbol{A}^{band-spa}$ and $\boldsymbol{A}^{band-spe}$ are the corresponding adjacency matrices, which are calculated by:

$$\boldsymbol{A}_{ij}^{band-spa}=\begin{cases}\exp\left(-|i-j|\right), \text{if } i\neq j\\ 0, \quad \text{otherwise}\end{cases} \tag{5}$$

$$\boldsymbol{A}_{ij}^{band-spe}=\begin{cases}\exp\left(-\sqrt{\frac{1}{\overline{N}}\sum_{e=1}^{\overline{N}}\left(\overline{\boldsymbol{x}}_e^i-\overline{\boldsymbol{x}}_e^j\right)^2}\right), \text{if } i\neq j\\ 0, \quad \text{otherwise}\end{cases} \tag{6}$$

where $\boldsymbol{A}_{ij}^{band-spa}$ and $\boldsymbol{A}_{ij}^{band-spe}$ represent the similarity of the i-th and j-th spectral bands, respectively, with $1\le i,j\le B$. As indicated in (5), when the positions of the i-th and j-th bands are closer, the value of $\boldsymbol{A}_{ij}^{band-spa}$ is closer to 1; otherwise, $\boldsymbol{A}_{ij}^{band-spa}$ tends to 0. In (6), the spectral information is used to measure the similarity between the i-th and j-th bands, and $\overline{\boldsymbol{x}}_e^i$ and $\overline{\boldsymbol{x}}_e^j$ refer to the spectral values of the i-th and j-th bands in the e-th sample, respectively. To ensure that the similarity values are in the same range, a simplified version of the Gaussian kernel function mapping is used in $\boldsymbol{A}^{band-spa}$ and $\boldsymbol{A}^{band-spe}$.

2) Sample-based spatial and spectral graph construction: HSIs often have varied and complex land-cover distributions. Hence, sample-based dual graphs are designed to encode the local spatial and non-local spectral information. In the sample-based dual graphs, each sample is regarded as a node. Similarly, the sample-based spatial graph and spectral graph are denoted by $G^{sample-spa}=\{V^{sample},E^{sample-spa}\}$ and $G^{sample-spe}=\{V^{sample},E^{sample-spe}\}$, where V^{sample} represents the set of sample nodes, and $E^{sample-spa}$ and $E^{sample-spe}$ represent the edge sets that are defined by the spatial information and spectral values of the samples. The adjacency matrices $\boldsymbol{A}^{sample-spa}$ and $\boldsymbol{A}^{sample-spe}$ are calculated as follows:

$$\boldsymbol{A}_{ij}^{sample-spa}=\begin{cases}1, & 0<\text{dis}\left(\overline{\boldsymbol{x}}_i,\overline{\boldsymbol{x}}_j\right)\le\varphi\\ 0, & \text{otherwise}\end{cases} \tag{7}$$

$$\boldsymbol{A}_{ij}^{sample-spe} = \begin{cases} \exp\left(-\sqrt{\frac{1}{k}\sum_{b=1}^{k}(\bar{\boldsymbol{x}}_i^{s_b} - \bar{\boldsymbol{x}}_j^{s_b})^2} \right), & \text{if } i \neq j \\ 0, & \text{otherwise} \end{cases} \tag{8}$$

where $\boldsymbol{A}_{ij}^{sample-spa}$ and $\boldsymbol{A}_{ij}^{sample-spe}$ measure the similarity between the samples, with $1 \le i, j \le \bar{N}$, and $\mathrm{dis}\left(\bar{\boldsymbol{x}}_i, \bar{\boldsymbol{x}}_j\right)$ represents the Euclidean distance between the two sample locations in the HSIs. Furthermore, φ is a threshold and $\varphi = 20$, which controls whether the samples in the sample-based spatial graph are connected. φ is selected by a trial-and-error procedure in the candidate range of [4, 24] with an interval of 2. If the threshold φ is too small, the spatial relationship between the samples may not be fully utilized. In contrast, if the threshold φ is too large, some unrelated samples may be connected. Eventually, φ was set to 20 in our experiments. It can be observed that the elements of the similarity matrices are nonzero unless the two corresponding samples are sufficiently close. To obtain a more accurate representation, the current k selected bands are used to evaluate the spectral similarity between the samples. s_b is the index of the b-th selected band.

4.2. Band Selection with GCN-Based Attention Mechanism and Sparse Constraint

HSIs have abundant spectral bands and high spectral resolution, which provide sufficient information for land-cover classification. However, a large number of spectral bands will result in high computational pressure and many spectral bands with noise or redundant information may be included. These spectral bands may lead to a deterioration in the classification performance. To alleviate this problem, band selection methods aim to retain the most discriminative or informative bands from the original spectral bands.

In the BSD-GCN, an attention branch that is based on dual-graph convolution and sparse constraint is designed to achieve the band selection of HSIs. The structure of the attention branch of the BSD-GCN is depicted in Fig. 3. The band-based dual graphs are used as the input for the attention branch of the BSD-GCN. Dual-graph convolution with batch normalization extracts and integrates the features from the band-based graphs. Subsequently, the band attention map is produced using average pooling and mapping with the softmax activation function. Each weight of the band attention map is regarded as the measurement of the importance of the corresponding band. To remove the redundant and noisy bands, the weights of the band attention map are truncated to make them sparse. Finally, the sparse band attention map is used as a mask and inserted into the first layer of the truck branch of the BSD-GCN.

In the attention branch, the band-based graphs $G^{band-spa}$ and $G^{band-spe}$ are input as the spatial and spectral information representations of all spectral bands, respectively. The dual-graph convolution is implemented on the two input graphs and the corresponding band maps are produced. The dual-graph convolution aggregates the features of each band and its correlated bands, and transforms the aggregated features into a new feature space. These feature maps are fused using the element-wise addition operation as the output feature map $\boldsymbol{F}$. Thereafter, average pooling is applied to each row of the feature map $\boldsymbol{F}$. Each row of the feature map $\boldsymbol{F}$ can be considered as a descriptor of each corresponding band of the HSIs. The band attention map $\boldsymbol{g}$ is generated by mapping the pooled feature map with the softmax activation function. Thus, the elements in the band attention map $\boldsymbol{g}$ are constrained between 0 and 1. This process is calculated as follows:

$$\boldsymbol{g} = \mathrm{softmax}(AP(\boldsymbol{F})) \tag{9}$$

$$\begin{aligned} \boldsymbol{F} = {} & f\left(\bar{\boldsymbol{A}}^{band-spe}\bar{\boldsymbol{X}}\boldsymbol{W}^{band-spe} + \boldsymbol{b}^{band-spe}\right) \\ & + f\left(\bar{\boldsymbol{A}}^{band-spa}\bar{\boldsymbol{X}}\boldsymbol{W}^{band-spa} + \boldsymbol{b}^{band-spa}\right) \end{aligned} \tag{10}$$

where $\bar{\boldsymbol{A}}^{band-spe}$ and $\bar{\boldsymbol{A}}^{band-spa}$ are the normalized matrices of $\boldsymbol{A}^{band-spe}$ and $\boldsymbol{A}^{band-spa}$, respectively, whereas $\boldsymbol{W}^{band-spe}$, $\boldsymbol{W}^{band-spa}$, $\boldsymbol{b}^{band-spe}$, and $\boldsymbol{b}^{band-spa}$ are learnable parameters. $f(\cdot)$ is the rectified linear unit (ReLU) function and $AP(\cdot)$ refers to the average pooling for each row of the input.

The band attention map can be regarded as the evaluation of each corresponding spectral band. A smaller weight of the band attention map indicates that the corresponding band is more likely to be noisy or irrelevant. The band attention map is sparsely constrained to retain the discriminative bands, and to suppress the noisy and irrelevant bands. The sparse band attention map $\boldsymbol{m}$ is obtained by selecting the top k larger attention weights after ranking and setting the remaining weights to zero, which is formulated as:

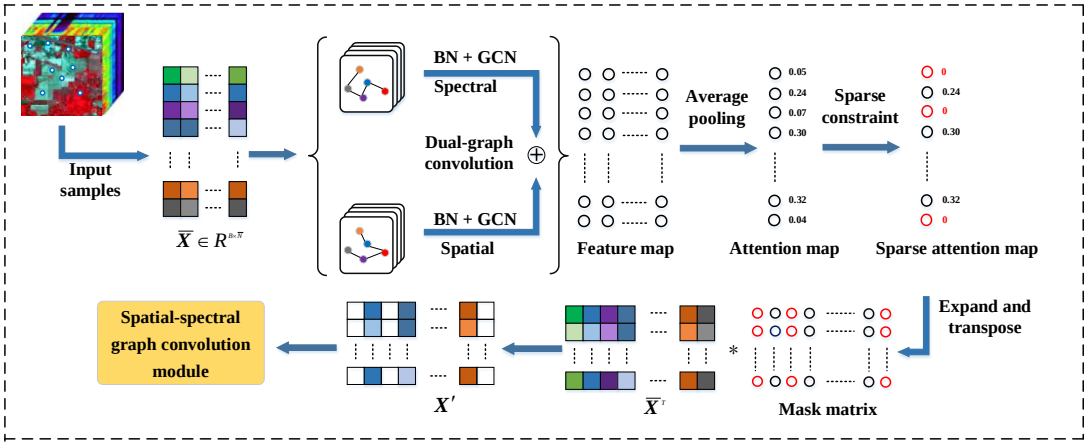

Fig. 3 Structure of attention branch.

$$\boldsymbol{m}_i = f_{top}(\boldsymbol{g}_i) = \begin{cases} \boldsymbol{g}_i, & \text{if } \boldsymbol{g}_i > \text{sort}(\boldsymbol{g}, k+1) \\ 0, & \text{otherwise} \end{cases} \tag{11}$$

where $\boldsymbol{m}_i$ and $\boldsymbol{g}_i$ refer to the i -th elements of vectors $\boldsymbol{m}$ and $\boldsymbol{g}$, and k is the number of selected bands. Moreover, $\text{sort}(\boldsymbol{g}, k+1)$ represents the operation of sorting the input vector $\boldsymbol{g}$ in descending order and selecting the top $k+1$ value, whereas $f_{top}(\cdot)$ represents the constraint function.

To allocate sparse attention weights to emphasize or suppress the corresponding spectral bands, the sparse attention map is first expanded and transformed from the vector to the matrix. Subsequently, the sparse attention matrix is inserted as a mask into the first layer of the truck branch. Finally, after using the mask, the input data of the trunk branch contains only the selected spectral bands. Specifically, the original sparse attention map $\boldsymbol{m}$ is a column vector. It is expanded to a mask matrix by multiplying a row vector in which all elements are 1, which is formulated as:

$$\boldsymbol{M} = (\boldsymbol{m} \circ \boldsymbol{l})^T \tag{12}$$

where $\circ$ represents the matrix multiplication and $\boldsymbol{l}$ is a row vector of size $\bar{N}$ in which all elements are 1. The mask matrix $\boldsymbol{M}$ is subsequently embedded as follows:

$$\boldsymbol{X}' = \bar{\boldsymbol{X}}^T * \boldsymbol{M} \tag{13}$$

In (13), the input data with selected bands $\boldsymbol{X}'$ are acquired using element-wise multiplication with the mask matrix $\boldsymbol{M}$, which is used as the input for the trunk branch of the BSD-GCN.

4.3. Spatial–Spectral Classification based on Dense GCN with Selected Bands

Owing to the imaging mechanism of HSIs, it is possible for certain samples with different spectral signatures to belong to the same category, whereas samples with similar spectral signatures may belong to different categories. In recent years, spatial–spectral classification has become a new trend in HSI processing. In the trunk branch of the BSD-GCN, a dense GCN is constructed for the spatial–spectral classification of HSIs, whereby spatial–spectral graph convolution modules are designed to extract the local spatial and non-local spectral features simultaneously. Sample-based spatial and spectral graphs are input into the spatial–spectral graph convolution module. If the sample-based spectral graph is constructed using all spectral bands of the HSIs, this graph may not be sufficiently accurate because of several redundant and noisy spectral bands. Thus, the sample-based spectral graph is constructed using only the selected bands that are generated by the attention branch of the BSD-GCN.

In the spatial–spectral graph convolution module, the sample-based dual graph models the spatial and spectral correlations among the samples into a topological structure. In this manner, the dual-graph convolutional layer can extract the spatial–spectral and topological features. In the trunk branch of the BSD-GCN, the spatial–spectral graph convolution module is stacked layer by layer to promote the classification performance. However, the stacking of excessive layers may result in degraded performance owing to the gradient vanishing problem [56]. Drawing on the concepts of CNNs, skip connections [57] are introduced into the GCN to alleviate this problem. Inspired by this idea, dense connections are applied to enhance the reuse of features and to exploit the information flow from the shallow to deep layers. This enables the BSD-GCN with multiple layers to converge reliably and achieve superior performance.

The specific structure of the trunk branch of the BSD-GCN is depicted in Fig. 4. The sample-based

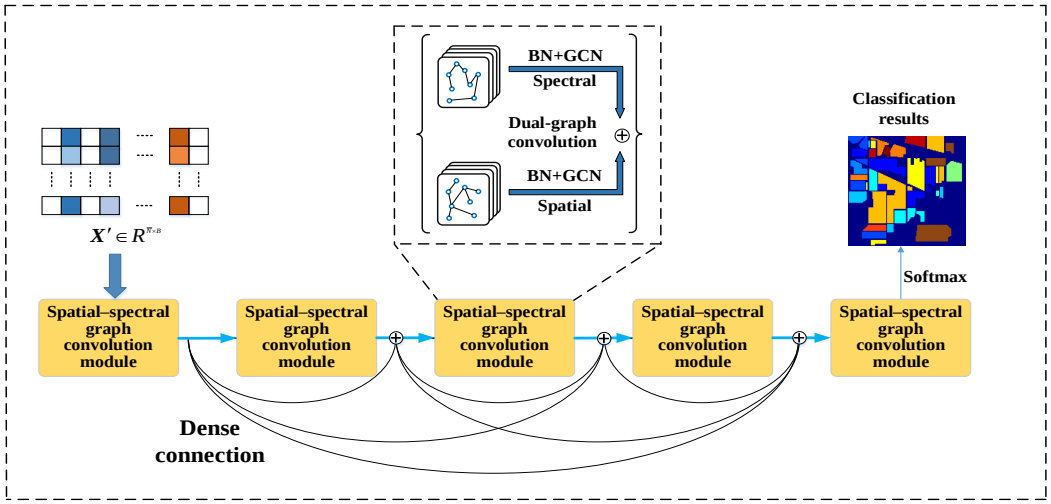

Fig. 4 Structure of trunk branch.

spatial graph $G^{sample-spa}$ and spectral graph $G^{sample-spe}$ are used as the input. Five spatial–spectral graph convolution modules are stacked with dense connections. Each spatial–spectral graph convolution module contains a dual-graph convolution layer based on the sample-based spatial graph and spectral graph. Among the five spatial–spectral graph convolution modules, the feature map that is generated by the current module is added to that of each subsequent module. Following five densely connected modules, the generated feature map is input into the softmax classification layer to predict the class labels.

The $l+1$-th spatial–spectral graph convolution module in the trunk branch of the BSD-GCN is calculated as follows:

$$\begin{aligned} \boldsymbol{H}^{l+1} = & f\left(\overline{\boldsymbol{A}}^{sample-spa}\left(\sum_{i=1}^{l}\boldsymbol{H}^{i}\right)\boldsymbol{W}_1^{l+1}+\boldsymbol{b}_1^{l+1}\right) \\ & + f\left(\overline{\boldsymbol{A}}^{sample-spe}\left(\sum_{i=1}^{l}\boldsymbol{H}^{i}\right)\boldsymbol{W}_2^{l+1}+\boldsymbol{b}_2^{l+1}\right) \end{aligned} \tag{14}$$

where $\boldsymbol{H}^{l+1}$ and $\boldsymbol{H}^{i}$ are the feature maps of the $l+1$-th and i-th layers, respectively. Furthermore, $\boldsymbol{H}^{0}=\boldsymbol{X}'$. , $\boldsymbol{W}_1^{l+1}$, $\boldsymbol{b}_1^{l+1}$, $\boldsymbol{W}_2^{l+1}$ and $\boldsymbol{b}_2^{l+1}$ are the learnable parameters in the $l+1$-th layer, whereas $\overline{\boldsymbol{A}}^{sample-spa}$ and $\overline{\boldsymbol{A}}^{sample-spe}$ are the normalized adjacency matrices of $\boldsymbol{A}^{sample-spa}$ and $\boldsymbol{A}^{sample-spe}$, respectively.

4.4. Soft-Shifting Optimization of BSD-GCN

In the BSD-GCN, the attention branch integrates the band information to produce a sparse attention map of bands and the trunk branch extracts the features of the HSI data with the selected bands for classification. The embedding of the sparse attention map into the trunk branch bridges the gap between the two branches with the aim of achieving end-to-end learning. However, the sparse constraint results in an optimization problem. In the back-propagation of the attention branch, the gradients will be zero for updating the neurons corresponding to the unselected bands, which limits the model to gradually search for the appropriate subsets among all spectral bands.

To solve this problem, soft-shifting optimization is implemented to update the parameters in the BSD-GCN. Soft-shifting optimization involves two losses: the full band loss and selected band loss. In soft-shifting optimization, the attention maps before and after the sparse constraint are embedded into the same trunk branch to produce these two losses. The neurons corresponding to the unselected bands can be updated with the full band loss, which alleviates the optimization difficulty caused by the sparse constraint. Moreover, an adjusted factor is applied to combine the two losses and make soft-shifting on the focus of the optimization.

As described previously, the gradients of the unselected bands are zero during the iteration process. Specifically, let $\boldsymbol{P}^{m}$ represent the ultimate classification probability matrix that is produced by the trunk branch, which can be regarded as a variable relating to the constraint function $f_{top}(\cdot)$. This is formulated as follows:

$$\begin{aligned}\boldsymbol{P}^m &= \mathrm{softmax}\left(\mathrm{F}_{trunk}\left(\boldsymbol{m}, \bar{\boldsymbol{X}}\right)\right) \\ &= \mathrm{softmax}\left(\mathrm{F}_{trunk}\left(f_{top}\left(\boldsymbol{g}\right), \bar{\boldsymbol{X}}\right)\right)\end{aligned} \tag{15}$$

$$\frac{\partial f_{top}\left(\boldsymbol{g}_i\right)}{\partial \boldsymbol{g}_i} = \begin{cases} 1, \text{ if } \boldsymbol{g}_i > \mathrm{sort}\left(\boldsymbol{g}, k+1\right) \\ 0, \text{ otherwise} \end{cases} \tag{16}$$

where $\mathrm{F}_{trunk}\left(\boldsymbol{m}, \bar{\boldsymbol{X}}\right)$ refers to the feature extraction process in the trunk branch with the sparse attention map. It can be observed from (16) that the derivative of $f_{top}\left(\cdot\right)$ for the unselected bands is always zero, which may invalidate the traditional gradient descent algorithm.

In the BSD-GCN, the soft-shifting optimization also embeds the attention map $\boldsymbol{g}$ before the sparse constraint into the truck branch. The loss L_{ss} in the soft-shifting optimization consists of two terms: L^m and L^g, which refer to the selected band loss with the sparse attention map $\boldsymbol{m}$ and full band loss with the non-sparse attention map $\boldsymbol{g}$, respectively. L_{ss} is defined as follows:

$$L_{ss} = L^m + \lambda L^g \tag{17}$$

where λ is the adjusted factor between these two losses L^m and L^g, and $\lambda = \exp(-\frac{t}{T})$. Furthermore, t and T indicate the current number of epochs and total number of epochs, respectively. In the early epochs, the sparse attention map is gradually updated to seek more discriminative bands, which requires more assistance of the full band loss. Following hundreds of epochs, the selected bands tend to be stable. In this case, the BSD-GCN needs to focus increasingly on feature extraction and classification with the selected bands. Therefore, λ decreases with the increase in t and makes soft-shifting on the focus of the optimization from the full band loss to selected band loss.

The specific calculations of these two losses are expressed as follows:

$$L^m = -\sum_{i \in Z}\sum_{j=1}^{C} \boldsymbol{Y}_{ij} \log \boldsymbol{P}_{ij}^m \tag{18}$$

$$L^g = -\sum_{i \in Z}\sum_{j=1}^{C} \boldsymbol{Y}_{ij} \log \boldsymbol{P}_{ij}^g \tag{19}$$

TABLE I
PROCEDURE OF BSD-GCN

INPUT: Training set $\boldsymbol{X}^{tr}$; training label set $\boldsymbol{Y}$; number of selected bands k, number of training epochs T;
OUTPUT: Predicted labels of test samples and indices of selected bands.
Begin:
Training of BSD-GCN
for every epoch $t < T$:
  Randomly select partial samples to be $\bar{\boldsymbol{X}}$
  Calculate the band-based adjacency matrices $\boldsymbol{A}^{band-spa}$ and $\boldsymbol{A}^{band-spe}$ with $\bar{\boldsymbol{X}}$ using equations (5) and (6)
  Calculate the attention map $\boldsymbol{g}$ using equations (9) and (10)
  Calculate the sparse attention map $\boldsymbol{m}$ and embed $\boldsymbol{m}$ into the trunk branch using equations (11) to (13)
  Calculate the sample-based adjacency matrices $\boldsymbol{A}^{sample-spa}$ and $\boldsymbol{A}^{sample-spe}$ using $\bar{\boldsymbol{X}}$ with selected bands using equations (7) and (8)
  Calculate the soft-shifting loss L_{ss} with $\boldsymbol{g}$ and $\boldsymbol{m}$ using equations (15) and (17) to (20)
  Update the parameters in the BSD-GCN with gradient descent
  $t = t + 1$
end for
Test of BSD-GCN:
  Predict the samples in test set $\boldsymbol{X}^{test}$ using well-trained BSD-GCN
  Output the predicted labels of the test samples and the indices of the selected bands
End

$$\boldsymbol{P}^{g} = \mathrm{softmax}\left(\mathrm{F}_{trunk}\left(\boldsymbol{g}, \bar{\boldsymbol{X}}\right)\right) \tag{20}$$

where C is the number of classes, Z refers to the index set of labeled training samples in the current batch, $\boldsymbol{P}^{g}$ is the classification probability matrix that is produced by the truck branch with the non-sparse attention map $\boldsymbol{g}$, and $\boldsymbol{Y}_{ij}$ refers to the value of the i-th row and j-th column in the training label set $\boldsymbol{Y}$, which is equal to 1 if the i-th sample belongs to the j-th class and 0 otherwise.

By using soft-shifting optimization, the BSD-GCN can optimize the band selection and classification in an end-to-end manner. The detailed procedure of the BSD-GCN is summarized in Table I.

5. Experimental Results and Analysis

Three widely used HSI datasets were selected to investigate the performance of the proposed BSD-GCN method. A detailed description of the three datasets is provided in Section 5.1. The experimental settings, including the experimental conditions, comparison methods, and hyperparameter fine-tuning, are presented in Section 5.2. Sections 5.3-5.7 provide an analysis of the experimental results according to different aspects: the classification results, time consumption, sensitivity to the number of selected

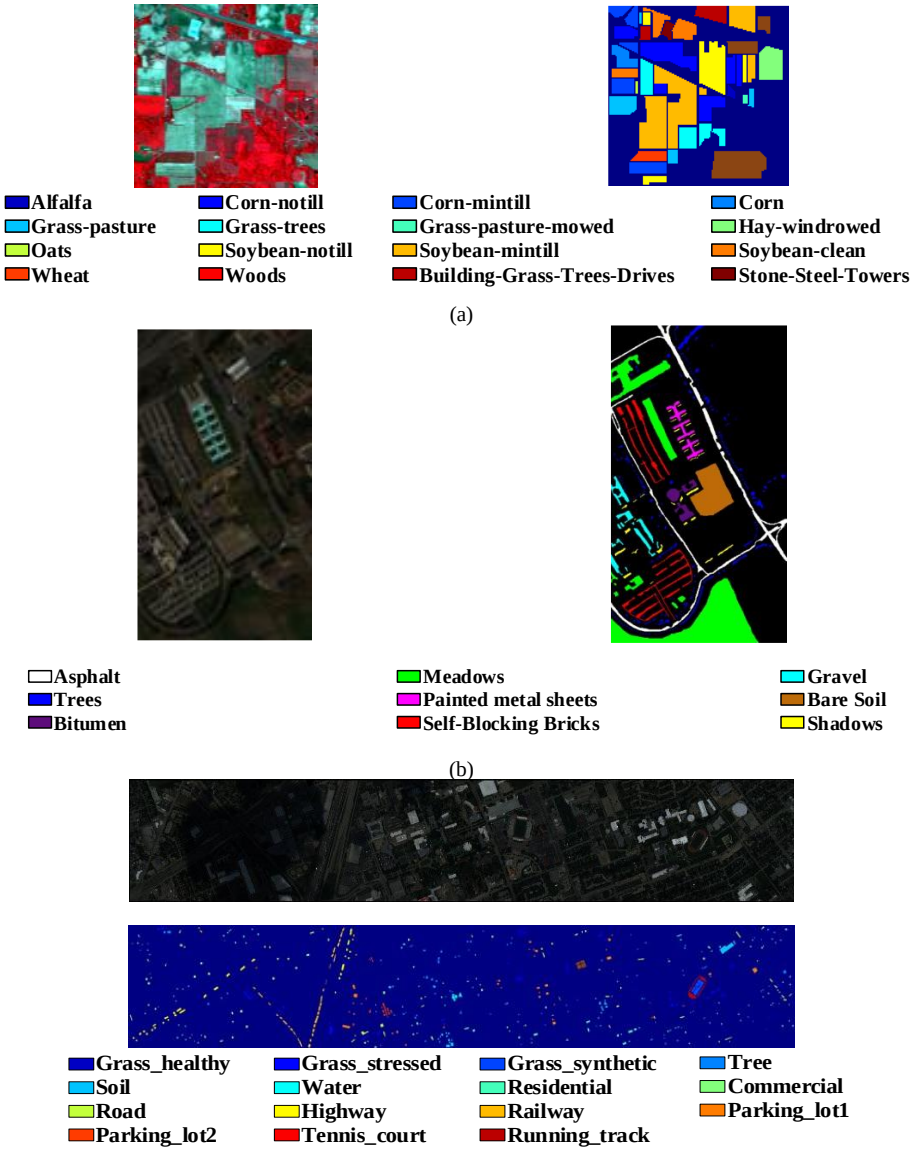

Fig. 5. False-color composite images and ground truth of three datasets: (a) Indian Pines, (b) Pavia University, and (c) University of Houston. The legends with different colors indicate different classes.

bands, selected spectral bands, and influence of different proportions of training samples. Sections 5.8 and 5.9 describe the use of an additional HSI dataset and state-of-the-art dimensionality reduction methods to verify the effectiveness of the BSD-GCN. Sections 5.10 and 5.11 present an analysis of the influence of the components and hyperparameters in the BSD-GCN.

5.1. Data Description

Three benchmark datasets were used in the experiments. Tables II-IV display the numbers of labeled and unlabeled training samples, and the test samples of each class in the three datasets.

1) Indian Pines: This dataset was acquired by the airborne visible/infrared imaging spectrometer sensor in 1992, recording the scene of northwestern Indiana, USA. It was provided by Marion F. Baumgardner, Larry L. Biehl, and David A. Landgrebe, Purdue University. This dataset consists of 145 × 145 pixels and 220 spectral bands. The spatial resolution of this dataset is 20 m, and the wavelength

TABLE II
DIFFERENT CLASSES AND NUMBERS OF LABELED TRAINING, UNLABELED TRAINING, AND TEST SAMPLES IN THE INDIAN PINES DATASET

#	Class	Labeled	Unlabeled	Test
1	Alfalfa	3	6	37
2	Corn-notill	72	144	1212
3	Corn-mintill	42	84	704
4	Corn	12	24	201
5	Grass-pasture	25	50	408
6	Grass-trees	37	74	619
7	Grass-pasture-mowed	2	4	22
8	Hay-windrowed	24	48	406
9	Oats	1	2	17
10	Soybean-notill	49	98	825
11	Soybean-mintill	123	246	2086
12	Soybean-clean	30	60	503
13	Wheat	11	22	172
14	Woods	64	128	1073
15	Buildings-Grass-Trees-Drives	20	40	326
16	Stone-Steel-Towers	5	10	78
	Total	520	1040	8689

TABLE III
DIFFERENT CLASSES AND NUMBERS OF LABELED TRAINING, UNLABELED TRAINING, AND TEST SAMPLES IN THE PAVIA UNIVERSITY DATASET

#	Class	Labeled	Unlabeled	Test
1	Asphalt	199	398	6034
2	Meadows	560	1120	16969
3	Gravel	63	126	1910
4	Trees	92	184	2788
5	Painted metal sheets	41	82	1222
6	Bare Soil	151	302	4576
7	Bitumen	40	80	1210
8	Self-Blocking Bricks	111	222	3349
9	Shadows	29	58	860
	Total	1286	2572	38918

TABLE IV
DIFFERENT CLASSES AND NUMBERS OF LABELED TRAINING, UNLABELED TRAINING, AND TEST SAMPLES IN THE UNIVERSITY OF HOUSTON DATASET

#	Class	Labeled	Unlabeled	Test
1	Grass_healthy	63	126	1062
2	Grass_stressed	63	126	1065
3	Grass_synthetic	35	70	592
4	Tree	63	126	1055
5	Soil	63	126	1053
6	Water	17	34	274
7	Residential	64	128	1076
8	Commercial	63	126	1055
9	Road	63	126	1063
10	Highway	62	124	1041
11	Railway	62	124	1049
12	Parking_lot1	62	124	1047
13	Parking_lot2	24	48	397
14	Tennis_court	22	44	362
15	Running_track	34	68	558
	Total	760	1520	12749

ranges from 400 to 2500 nm with a spectral resolution of 10 nm. After removing 20 water absorption and noisy bands, namely [104-108], [150-163], and 220, the remaining 200 bands are used in practice. This scene contains 16 land-cover classes. Fig. 5(a) depicts the false-color image consisting of bands 50, 27, and 17 of the Indian Pines dataset and the ground truth. For this dataset, 5% and 10% of the labeled samples are randomly selected as the labeled and unlabeled training sets, respectively. The remaining labeled samples are used for testing. This dataset can be obtained on the website http://cobweb.ecn.p-urdue.edu/~biehl/MultiSpec/ documentation.html.

2) Pavia University: This dataset was gathered by the Reflective Optical System Imaging Spectrometer in 2003 over Pavia University in northern Italy and provided by Prof. Paolo Gamba from the Telecommunications and Remote Sensing Laboratory of Pavia university. It consists of 610 × 340 pixels and 103 spectral bands. The spatial resolution of this dataset is 1.3 m, which is higher than that of most HSI datasets. The wavelength of the dataset ranges from 430 to 860 nm. Furthermore, there are 9 land-cover classes in this scene. The false-color image consisting of bands 53, 31, and 8 and the ground truth are shown in Fig. 5 (b). For this dataset, 3% and 6% of the labeled samples are randomly selected as the labeled and unlabeled training sets, respectively. The remaining labeled samples are used for testing. This dataset is available on the website http://www.ehu.eus/ccwintco/index.php/Hype-rspectral_Remote_S-ensing_Scenes.

3) University of Houston: This dataset was collected by the compact airborne spectrographic imager ITRES CASI-1500 in 2012, which covers the University of Houston campus. It was provided in the IEEE GRSS Data Fusion Contest in 2013 by National Center for Airborne Laser Mapping at the University of Houston. This dataset includes 349 × 1905 pixels and 144 spectral bands ranging from 364 to 1046 nm. The spatial and spectral resolutions of this dataset are 2.5 m and 10 nm, respectively. This dataset is a cloud-free hyperspectral product, which was processed by removing several small structures according to the illumination-related threshold maps that were computed based on the spectral signatures. Fig. 5 (c) presents the ground truth and false-color image consisting of bands 28, 45, and 65, which contains 15 classes. For this dataset, 5% and 10% of the labeled samples are randomly selected as the labeled and unlabeled training sets, respectively. The remaining labeled samples are used for testing. This dataset can be downloaded on the website https://hyperspectral.ee.uh.edu/?page_id=459.

5.2. Experimental Settings

In the experiments, eight band selection methods were selected for a performance comparison with the proposed BSD-GCN method, including two filter methods, namely the mRMR [24] and HM [25], three wrapper methods, namely the SICNN [28], DDCNN [47], and DCS [27], two embedded methods, namely the TWCNN [40] and MR-SVM [30], and a special method, the ABCNN [49]. To compare the full-band performance, the SVM with the radial basis function (RBF-SVM) [58] was implemented to classify the samples with full spectral bands. Among these methods, the HM and DCS are semi-supervised methods, whereas the others are supervised methods. The experiments were implemented on a computer equipped with an AMD Ryzen 9 3950X CPU and a Nvidia RTX TITAN GPU. The proposed BSD-GCN method was built in Python with the TensorFlow framework.

We randomly selected 20% training samples as the validation set to fine-tune the hyperparameters in the comparison methods. Three main methods were used for determining the hyperparameters: the grid search, trial-and-error procedure, and recommendations of original papers. The grid search was used to build the SVM classifier in the RBF-SVM, mRMR, MR-SVM, HM, and DCS, which searched c in the range of {1, 10, 100, 1000, 10000} and gamma in the range of {0.01, 0.1, 1, 10, 100}. The trial-and-error procedure was employed to explore the optimal hyperparameters in the CNNs, such as the batch size, learning rate, and number of training epochs, according to their classification performance on the validation set. Moreover, this strategy was used to determine the number of partitions of full bands of the DDCNN, which was finally set to 5, and the number of neighbors in the *K*-nearest neighbors of the DCS, which was finally set to 9. The recommendations of original papers were adopted when it was difficult to determine the hyperparameters. This approach was used for the parameters of FODPSO of the SICNN in [28], the weight of the constraint term of the TWCNN in [40], and the parameters of anomaly detection of the ABCNN in [49]. For the BSD-GCN, the ultimate learning rate and batch size were set to 0.001 and 128, respectively, and the total number of epochs was set to 1000. The implementation guidelines and Python code of the BSD-GCN are publicly available on the website https://github.com/xidian-yzw/BSD-GCN.git.

5.3. Classification Results

Three widely recognized indicators, namely the overall accuracy (OA), average accuracy (AA), and

TABLE V
CLASSIFICATION RESULTS OF RBF-SVM, MRMR, TWCNN, MR-SVM, SICNN, DDCNN, ABCNN, HM, DCS, AND BSD-GCN ON INDIAN PINES DATASET

Class	RBF-SVM	mRMR	TWCNN	MR-SVM	SICNN	DDCNN	ABCNN	HM	DCS	BSD-GCN
1	55.8±14.1	29.5±5.8	68.2±21.4	67.4±6.3	83.4±7.0	75.0±6.9	23.3±11.2	30.3±8.9	36.4±14.5	80.4±3.8
2	74.6±1.7	69.0±2.8	89.1±0.5	70.5±4.3	81.9±4.9	84.8±5.3	75.1±3.1	71.9±2.6	72.4±0.8	96.7±1.2
3	63.4±4.4	60.8±4.5	92.9±1.7	54.0±2.1	82.8±3.2	80.7±2.4	73.0±4.3	60.7±3.7	56.3±3.3	97.4±0.7
4	42.1±5.1	34.7±2.8	90.9±2.5	23.8±0.8	85.4±2.3	85.3±1.1	41.9±4.2	39.5±5.8	51.4±11.7	97.6±0.5
5	88.7±2.7	84.6±4.8	86.1±5.9	88.8±4.6	79.9±8.4	85.8±1.4	88.0±1.1	85.3±1.9	85.8±1.5	95.8±0.4
6	95.6±2.2	94.3±3.6	95.5±3.0	93.7±0.8	96.3±2.0	92.6±2.6	93.0±1.7	95.7±2.2	93.3±2.0	99.8±0.2
7	70.8±11.1	40.7±2.8	93.8±8.7	65.4±7.9	82.4±3.9	40.7±7.3	75.0±1.6	38.3±3.6	55.8±26.0	96.1±3.8
8	97.4±1.0	93.9±3.6	100.0±0.0	96.4±0.4	86.0±9.8	94.0±3.9	97.8±0.8	94.2±4.0	93.0±3.3	100.0±0.0
9	24.2±13.7	21.1±6.4	87.7±2.5	10.5±1.7	42.2±9.8	42.1±9.3	41.2±8.9	16.5±9.4	32.9±16.9	83.3±9.4
10	72.1±3.3	69.0±4.0	89.5±2.7	64.9±2.7	78.4±3.7	49.5±3.1	78.9±3.0	66.8±4.1	66.6±3.8	96.0±1.7
11	79.1±2.3	72.7±2.9	96.4±2.8	84.5±2.6	94.1±3.4	84.7±1.6	82.9±0.7	75.4±3.0	81.1±2.2	98.5±0.2
12	60.9±3.8	52.7±7.1	76.3±4.0	59.8±3.1	78.3±6.8	70.8±3.2	73.8±4.6	59.0±5.2	63.1±10.6	96.6±0.9
13	97.2±1.5	95.3±2.6	98.6±0.9	95.3±0.1	82.3±8.0	98.9±7.6	100±0.0	91.8±3.6	94.8±4.8	91.4±0.3
14	91.4±2.7	93.1±1.8	97.2±1.4	95.5±0.4	97.7±1.4	98.0±1.3	95.8±1.2	92.7±2.4	92.3±2.8	99.9±0.1
15	43.3±6.3	43.0±3.5	87.5±6.3	48.8±8.3	83.6±3.3	76.8±4.6	58.3±2.3	46.3±4.6	39.1±9.8	99.1±0.5
16	89.2±5.0	85.2±6.4	91.4±11.4	89.7±9.1	88.3±6.3	90.9±4.5	89.2±0.4	82.8±3.6	89.6±5.3	95.2±0.3
OA (%)	77.5±0.4	73.5±0.7	92.4±1.0	75.8±1.0	87.6±1.0	82.7±1.0	81.5±1.7	74.9±1.5	75.2±1.2	97.7±0.4
AA (%)	71.6±1.6	65.0±2.1	90.1±2.4	69.3±2.4	82.7±2.6	78.2±2.1	74.2±2.0	65.4±1.7	69.0±3.0	95.3±2.8
Kappa (%)	74.3±0.5	69.8±0.8	91.3±1.1	73.3±1.7	85.9±1.1	80.4±1.5	78.8±1.4	71.3±1.7	72.7±1.4	97.5±0.9

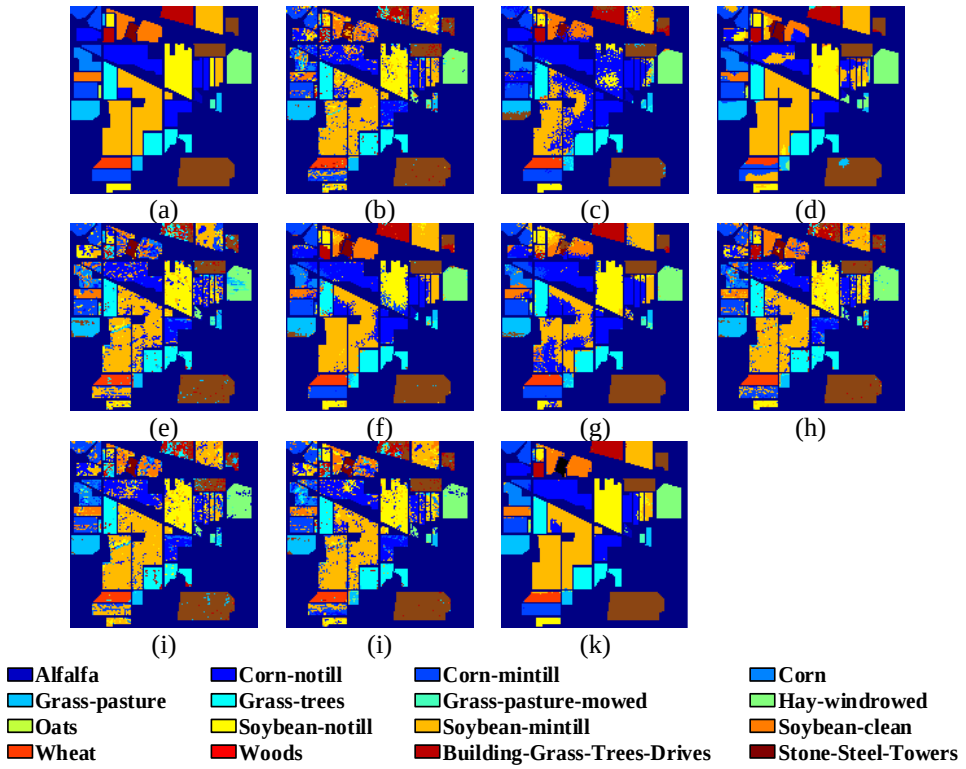

Fig. 6 Visual classification maps of the Indian Pines dataset: (a) ground truth, (b) RBF-SVM, (c) mRMR, (d) TWCNN, (e) MR-SVM, (f) SICNN, (g) DDCNN, (h) ABCNN, (i) HM, (j) DCS, and (k) BSD-GCN.

Kappa coefficient (Kappa), were employed to compare the performance of the different algorithms.

1) Indian Pines: Among all 200 spectral bands, 60 bands were selected for all of the band selection algorithms. The average classification results of the 30 independent runs are listed in Table V. The best results for each class in terms of the OA, AA, and Kappa are marked in gray. As indicated in Table V, the filter methods, namely the mRMR and HM, performed worse than the other methods because of the independence between the band selection and classification. The HM constructs a hypergraph to model the relations among samples and optimizes a projection matrix for band selection, which is independent of the classification. The DCS outperformed the HM through band selection guided by classification. Compared to the mRMR, HM, and DCS, the MR-SVM performed better by combining band selection and classification into one process. Benefiting from the strong feature extraction abilities of CNNs, the TWCNN, SICNN, DDCNN, and ABCNN surpassed the other comparison methods. The SICNN utilizes a 2D-CNN for classification, which involves more spatial information than the 1D-CNN in the DDCNN and ABCNN. In the TWCNN, depth-wise convolution is used for band selection. Compared to the SICNN and DDCNN, the TWCNN improved the classification performance. Compared to the CNN-based algorithms, the BSD-GCN exhibited an improvement of at least 5.3% in terms of the OA, 5.2% in terms of the AA, and 6.2% in terms of Kappa.

TABLE VI
CLASSIFICATION RESULTS OF RBF-SVM, MRMR, TWCNN, MR-SVM, SICNN, DDCNN, ABCNN, HM, DCS, AND BSD-GCN ON THE PAVIA UNIVERSITY DATASET

Class	RBF-SVM	mRMR	TWCNN	MR-SVM	SICNN	DDCNN	ABCNN	HM	DCS	BSD-GCN
1	88.5±1.0	94.7±2.3	96.2±0.9	83.2±2.3	93.0±1.9	99.2±0.3	93.2±0.6	88.7±1.7	87.2±2.1	98.6±0.7
2	97.5±0.2	97.9±0.7	99.6±0.2	94.4±0.1	99.0±0.4	97.6±0.9	96.8±1.0	96.3±1.6	94.9±1.8	99.7±0.3
3	65.5±3.1	34.3±1.8	65.2±8.9	89.1±2.0	96.1±2.5	74.2±2.5	73.5±2.3	61.7±0.2	57.9±5.3	90.7±4.0
4	90.0±1.6	94.5±0.7	97.6±0.6	94.2±0.9	94.8±2.6	85.5±2.7	88.0±4.7	86.7±2.3	85.1±3.0	95.2±0.7
5	99.3±0.1	99.5±0.1	100.0±0.0	96.3±0.1	99.2±0.2	82.9±0.4	99.5±0.1	98.7±0.2	98.7±0.2	98.8±1.0
6	61.1±1.1	41.9±0.7	96.0±1.3	78.0±5.5	87.2±1.8	66.5±0.6	86.6±2.2	50.9±11.9	70.8±4.9	99.2±0.7
7	76.6±4.5	84.4±8.3	84.8±3.3	30.4±9.3	78.1±1.7	69.8±4.3	76.2±6.1	71.5±1.8	74.0±4.7	99.0±0.9
8	85.5±2.6	89.0±2.3	94.4±0.3	83.9±2.4	96.6±2.0	93.6±0.1	79.9±3.4	82.1±2.1	83.2±3.5	97.3±1.4
9	95.9±2.1	87.4±0.2	98.6±1.0	98.5±0.4	93.6±4.1	86.2±1.8	99.5±0.1	99.6±0.1	99.6±0.1	89.6±4.0
OA (%)	88.0±0.5	86.1±0.4	95.9±0.8	85.7±0.6	95.3±0.9	90.3±0.7	91.3±0.1	85.6±0.6	86.2±0.6	98.1±0.4
AA (%)	84.4±0.6	80.4±1.2	92.5±1.5	83.1±1.5	93.1±0.7	83.9±0.7	88.1±0.4	81.8±0.6	83.5±0.8	97.5±0.6
Kappa (%)	83.9±0.6	81.0±0.8	94.5±1.0	82.8±1.0	93.7±0.7	87.1±0.8	88.5±1.3	80.5±0.9	82.5±0.8	96.3±0.8

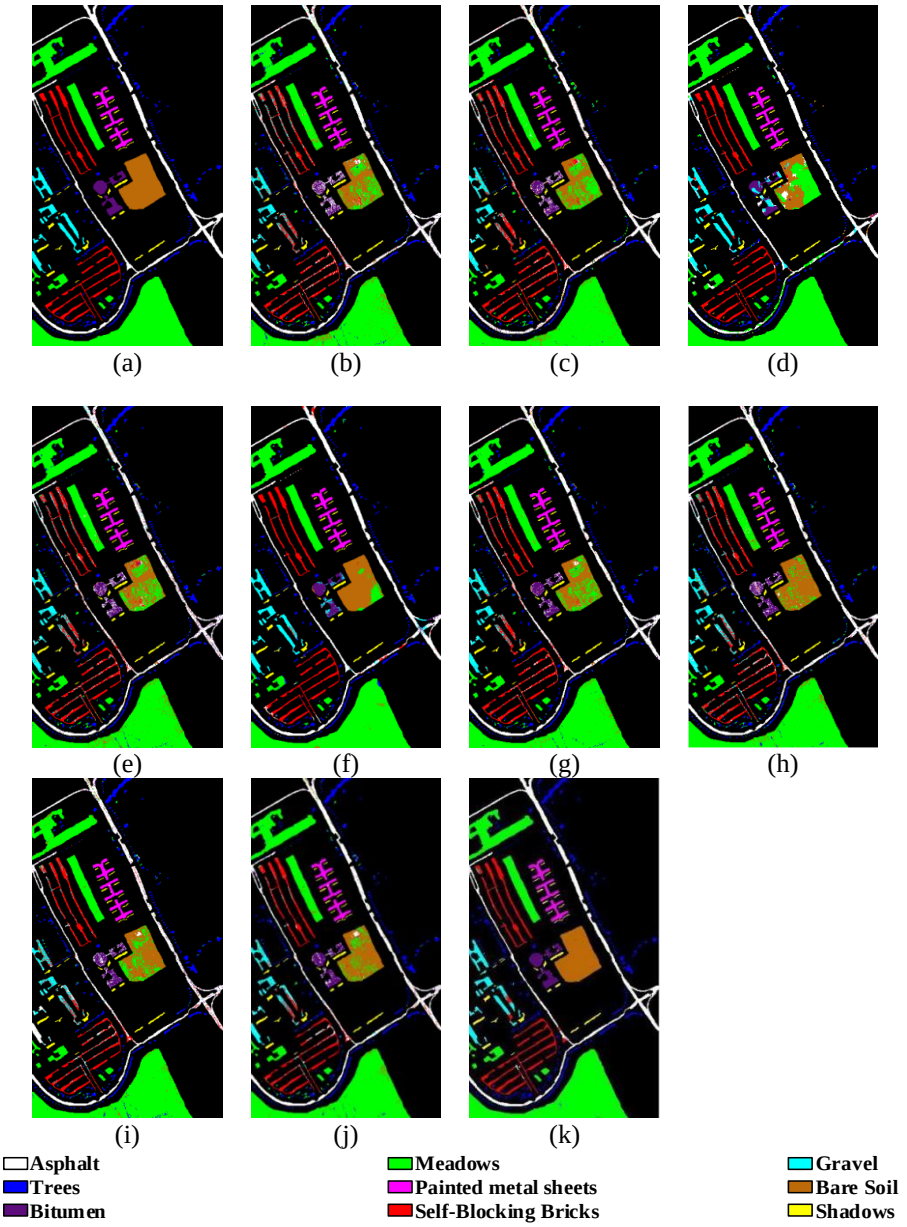

Fig. 7 Visual classification maps of the Pavia University dataset: (a) ground truth, (b) RBF-SVM, (c) mRMR, (d) TWCNN, (e) MR-SVM, (f) SICNN, (g) DDCNN, (h) ABCNN, (i) HM, (j) DCS, and (k) BSD-GCN.

Fig. 6 shows the visual classification maps of the 10 methods and the ground truth of the Indian Pines dataset. As illustrated in Fig. 6 (b), (c), (e), and (g)-(j), each map contained many misclassified samples. For the RBF-SVM, mRMR, MR-SVM, DDCNN, ABCNN, HM, and DCS, the input samples of the classifier were at the pixel level, which resulted in noisy points in many regions. As illustrated in Fig. 6 (d) and (f), the TWCNN and SICNN used spatial windows as the input and improved the regional consistency. However, several misclassifications occurred in the regions adjacent to the boundary, such

TABLE VII
CLASSIFICATION RESULTS OF RBF-SVM, MRMR, TWCNN, MR-SVM, SICNN, DDCNN, ABCNN, HM, DCS, AND BSD-GCN ON THE UNIVERSITY OF HOUSTON DATASET

Class	RBF-SVM	mRMR	TWCNN	MR-SVM	SICNN	DDCNN	ABCNN	HM	DCS	BSD-GCN
1	93.1±1.7	92.8±0.3	93.1±3.0	91.8±0.1	97.6±0.7	92.9±1.1	97.9±1.2	95.5±1.8	97.5±1.5	98.0±1.2
2	96.7±0.4	95.5±0.4	97.8±1.3	96.3±0.3	98.5±0.5	96.1±1.5	89.9±2.4	96.1±1.5	95.2±2.5	99.0±0.3
3	97.4±0.3	96.7±0.1	99.8±0.2	97.3±0.1	98.5±0.4	99.8±0.1	98.7±0.5	97.1±4.5	99.4±0.3	99.8±0.1
4	92.6±2.1	88.7±0.3	92.4±2.3	94.1±0.3	98.7±0.1	89.1±2.0	95.5±1.3	94.1±4.2	95.8±0.7	99.0±1.0
5	97.2±0.2	99.6±0.2	99.3±0.4	98.1±0.7	96.5±0.5	98.0±0.8	99.2±0.4	97.4±1.8	98.8±0.2	99.7±0.3
6	94.2±0.4	95.1±0.4	88.1±1.2	93.9±0.9	93.6±1.4	89.3±0.6	96.8±1.0	92.2±3.2	94.9±1.1	98.5±1.3
7	94.6±0.6	92.7±1.4	93.3±1.5	86.1±1.1	91.4±0.9	94.8±0.6	95.7±2.5	87.6±4.7	92.9±0.7	94.4±2.4
8	76.2±3.8	61.5±2.1	88.5±2.1	77.8±3.7	83.8±2.9	80.0±3.4	88.4±2.9	75.9±4.2	78.5±4.0	93.1±2.2
9	89.5±1.7	85.8±1.4	90.0±3.7	94.5±1.3	92.4±1.7	88.6±0.7	73.0±5.6	76.3±7.9	84.6±3.2	90.6±3.4
10	95.4±2.8	97.2±0.9	94.3±4.2	92.7±3.3	94.3±2.3	92.9±0.4	97.1±1.8	88.2±3.2	89.2±1.1	96.8±1.8
11	96.8±0.6	96.6±1.3	97.1±1.5	96.1±2.4	86.7±1.3	98.7±0.4	89.2±1.1	84.2±4.4	87.5±1.1	96.7±0.9
12	86.4±3.6	83.3±5.4	92.5±2.7	86.7±3.1	89.9±4.4	90.3±1.5	91.6±2.1	78.8±4.4	85.9±0.4	94.8±2.5
13	73.1±8.3	72.6±8.8	93.2±5.1	72.9±7.3	97.6±1.2	82.5±2.1	55.5±6.9	34.2±6.4	45.5±5.3	95.6±1.9
14	99.0±0.1	99.5±0.1	96.3±2.6	98.5±0.2	97.1±0.4	97.3±1.9	100±0.0	96.6±2.0	98.8±0.9	100.0±0.0
15	99.5±0.3	96.0±0.5	96.3±4.3	99.5±0.3	97.8±0.1	98.2±0.2	99.0±0.3	98.5±0.6	98.5±0.4	99. 7±0.1
OA (%)	92.1±0.7	88.9±0.6	94.1±1.3	90.7±0.8	93.8±0.7	92.5±0.9	91.6±0.4	87.1±1.3	90.3±0.7	96.7±0.3
AA (%)	92.1±1.2	90.2±1.4	94.1±1.4	91.7±1.3	92.7±1.7	92.6±1.1	91.2±0.9	86.2±1.0	89.5±0.7	97.1±0.4
Kappa (%)	91.5±0.8	88.1±1.1	93.7±1.1	91.0±0.9	93.4±0.9	91.9±1.1	90.9±0.8	86.0±1.4	89.5±0.8	96.4±0.4

TABLE VIII
THE RANKING OF CLASSIFICATION PERFORMANCE OF DIFFERENT METHODS ACROSS HSI DATASETS

Dataset	**Indian Pines**			**Pavia University**			**University of Houston**		
Index	**OA**	**AA**	**Kappa**	**OA**	**AA**	**Kappa**	**OA**	**AA**	**Kappa**
BSD-GCN	1	1	1	1	1	1	1	1	1
TWCNN	2	2	2	2	3	2	2	2	2
SICNN	3	3	3	3	2	3	3	3	3
DDCNN	4	4	4	5	6	5	4	4	4
ABCNN	5	5	5	4	4	4	6	7	7
RBF-SVM	6	6	6	6	5	6	5	5	5
MR-SVM	7	7	7	9	8	7	7	6	6
DCS	8	8	8	8	7	8	8	9	8
mRMR	10	10	10	7	10	9	9	8	9
HM	9	9	9	10	9	10	10	10	10

as the Soybean-notill and Soybean-mintill classes. In the BSD-GCN, although pixel-wise samples were provided as nodes, the spatial information among samples was also considered. Thus, compared to the TWCNN and SICNN, the BSD-GCN achieved better division in most classes and provided more accurate boundary localization.

2) Pavia University: In this dataset, 30 bands were selected from the original 103 bands. The quantitative classification results of the algorithms on this dataset are presented in Table VI. The SICNN, TWCNN, and BSD-GCN achieved outstanding classification performance. In particular, the BSD-GCN outperformed the other methods in all the terms. Compared to the SICNN and TWCNN, the BSD-GCN exhibited an improvement of at least 2.2%, 4.4%, and 1.8% in terms of the OA, AA, and Kappa, respectively. For several categories, such as the Bitumen and Bare Soil classes, it was difficult for other methods to achieve satisfactory classification performance. However, the BSD-GCN could achieve competitive classification results by effectively learning the correlation between bands and extracting the discriminative features in the flexible receptive fields.

The ground truth and visual classification maps of the Pavia University dataset are depicted in Fig. 7. As illustrated in Fig. 7 (b)-(j), the RBF-SVM, mRMR, MR-SVM, DDCNN, ABCNN, HM, and DCS had many noisy scattered points, whereas the TWCNN and SICNN caused some regional misclassifications, which were obvious in the regions of Bare Soil (marked in brown) and Meadows (marked in green). Fig. 7 (j) presents the classification map of the BSD-GCN. Compared to the other methods, the BSD-GCN maintained higher regional consistency. The BSD-GCN provided a visual effect that was closer to the ground truth, especially for the Bare Soil and Meadows classes.

3) University of Houston: For this dataset, 40 spectral bands were selected from the original 144 spectral bands. The quantitative classification results of nine methods on this dataset are listed in Table

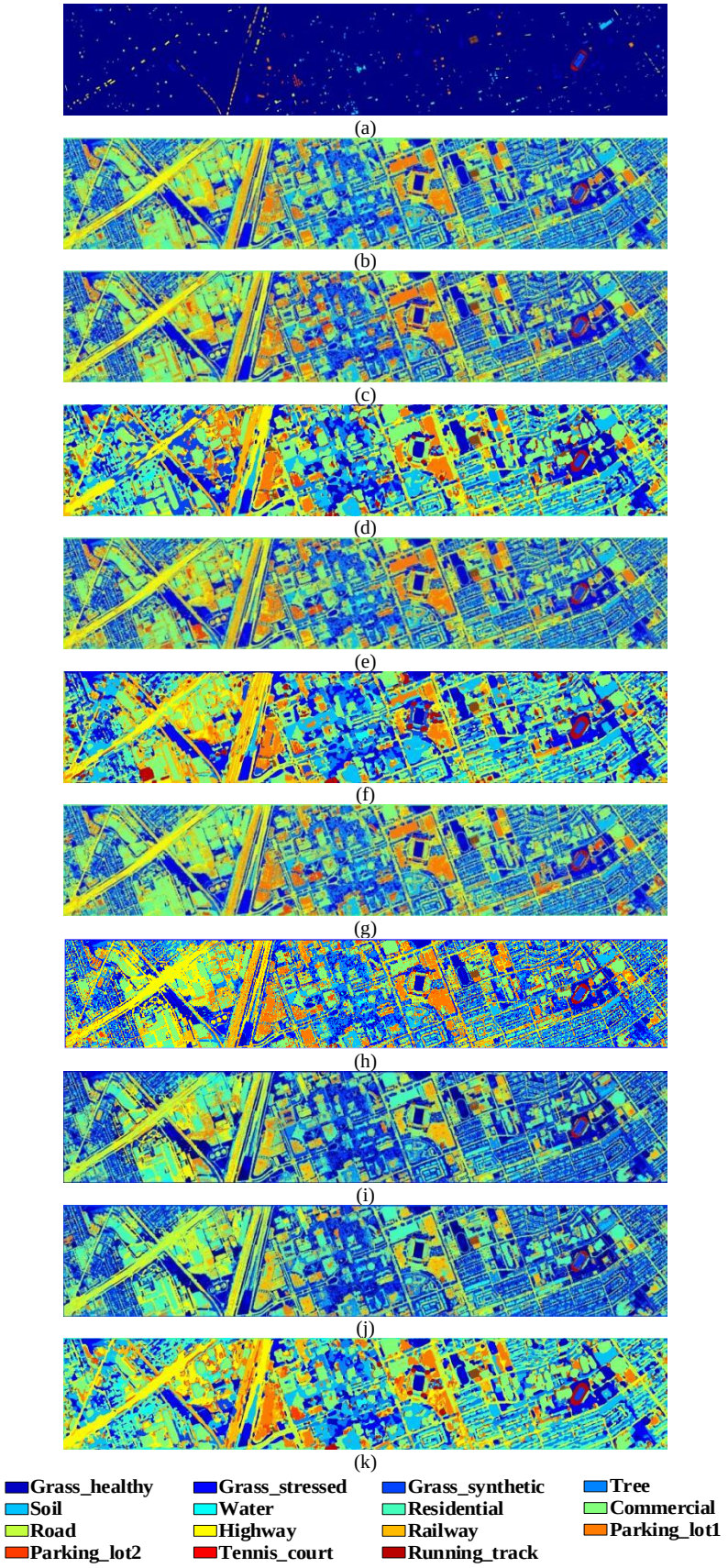

Fig. 8 Visual classification maps of the University of Houston dataset: (a) ground truth, (b) RBF-SVM, (c) mRMR, (d) TWCNN, (e) MR-SVM, (f) SICNN, (g) DDCNN, (h) ABCNN, (i) HM, (j) DCS, and (k) BSD-GCN.

VII. The BSD-GCN achieved the best classification performance in terms of the OA, AA, and Kappa among all of the methods. The BSD-GCN achieved the best classification accuracies in most categories

and maintained classification accuracies above 90% in all categories. For the Water and Commercial classes, the BSD-GCN improved the classification accuracies by at least 3.4% and 4.6%. For the Tennis_court class, the BSD-GCN achieved a completely correct result.

Fig. 8 shows the visual classification maps and ground truth of the University of Houston dataset. In this dataset, the distribution of the labeled samples is not as centralized as in the previous datasets. Thus, we used full-image prediction to visualize the classification results. As illustrated in Fig. 8, the classification maps that were produced by the mRMR, ABCNN, and MR-SVM contained numerous noticeable misclassifications in the regions of the labeled samples. The SICNN, TWCNN, and DDCNN maintained better regional consistency. There were many noisy points in the classification maps that were generated by the HM and DCS. Compared to the other methods, the BSD-GCN achieved better regional consistency and boundary localization.

To provide an overall and intuitive understanding of the performance of each algorithm, we added the Table VIII to record the results of experiments across all the hyperspectral scenes sets in the form of rankings. As indicated in Table VIII, BSD-GCN maintained the highest ranking in three indexes on three datasets. TWCNN and SICNN covered all the second and third places and these two methods were followed by ABCNN and RBF-SVM. MR-SVM, DCS, HM, and mRMR were mainly at the bottom of the rankings. Compared to other methods, the rankings of MR-SVM, mRMR and ABCNN are more unstable.

5.4. Investigation of Running Time

TABLE IX
RUNNING TIME OF RBF-SVM, MRMR, TWCNN, MR-SVM, SICNN, DDCNN, ABCNN, HM, DCS, AND BSD-GCN

Dataset	Methods	Training time (s)	Inference time (s)
Indian Pines	RBF-SVM	0.4±0.1	0.8±0.1
	mRMR	1.2±0.1	0.3±0.1
	TWCNN	147.7±2.5	1.4±0.1
	MR-SVM	101.0±0.1	0.3±0.1
	SICNN	513.6±0.8	1.7±0.2
	DDCNN	26.3±3.7	1.4±0.3
	ABCNN	61.4±2.8	1.3±0.1
	HM	3411.7±55.0	0.8±0.1
	DCS	11810.7±125.0	1.8±0.1
	BSD-GCN	1204.6±8.7	1.6±0.2
Pavia University	RBF-SVM	0.5±0.1	1.2±0.1
	mRMR	1.9±0.1	0.8±0.1
	TWCNN	234.6±7.4	2.5±0.2
	MR-SVM	156.2±0.1	0.8±0.1
	SICNN	757.3±1.0	2.7±0.2
	DDCNN	36.2±4.1	1.5±0.1
	ABCNN	102±3.4	1.7±0.1
	HM	10175.1±100.2	1.3±0.1
	DCS	16902.5±56.0	1.3±0.2
	BSD-GCN	3716±39.2	2.6±0.6
University of Houston	RBF-SVM	0.6±0.1	1.3±0.1
	mRMR	1.5±0.1	0.9±0.1
	TWCNN	207.0±0.4	1.5±0.1
	MR-SVM	126.1±0.6	0.9±0.1
	SICNN	604.8±2.1	2.1±0.2
	DDCNN	30.9±2.7	1.2±0.2
	ABCNN	82.1±2.1	1.4±0.3
	HM	12798±64.0	1.5±0.2
	DCS	17161.4±150.5	1.3±0.2
	BSD-GCN	1576.4±50.3	1.6±0.1

The running time of the nine methods for the three datasets is listed in Table IX. The running time included the training and test time. As indicated in Table IX, the RBF-SVM required the least time in the training stage owing to the lack of a band selection process. Among all the band selection methods, the mRMR was the fastest because of the efficiency of the incremental selection strategy. Compared to the mRMR, the MR-SVM required more time to retrain the SVM classifier. For the CNN-based methods, the DDCNN achieved a clear advantage in the training time owing to the use of a well-trained 1D-CNN. 1D convolution was also conducted in the ABCNN to improve efficiency. Anomaly detection for band selection cost some time in the training process. Each candidate band subset was used to retrain the CNN classifier in the SICNN, which was more time consuming. The TWCNN saved on the training time because it completed the band selection and classification synchronously. Compared to the other methods,

the HM, DCS, and BSD-GCN required more training time owing to the use of abundant unlabeled samples. Among all the band selection methods, the DCS required the most training time. This is because numerous base classifiers were trained in the DCS to evaluate different combinations of band subsets. The HM made use of hypergraphs to construct the relationships among samples, which occupied most of the training time. In the BSD-GCN, the construction of different types of graphs resulted in a certain time consumption. Compared to the HM, the BSD-GCN used minibatch learning to promote training efficiency. For the inference time, all methods completed the inference process within seconds, and the methods based on deep learning required more time for the inference of the networks.

5.5. Sensitivity to Number of Selected Bands

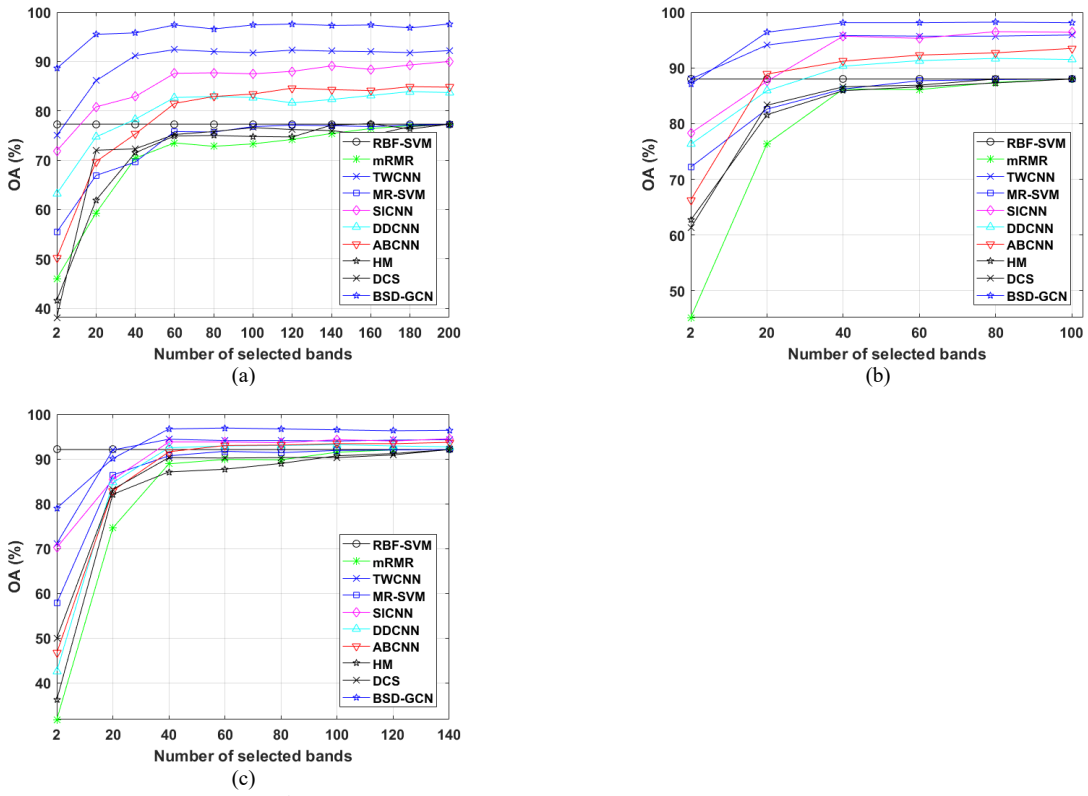

Fig. 9 Sensitivity to number of selected bands under single spilt: (a) Indian Pines, (b) Pavia University, and (c) University of Houston.

In this section, the performance of the band selection methods with different numbers of selected bands was analyzed using two sample division methods: single split and five-fold cross-validation, to provide a comprehensive observation.

Fig. 9 presents the variations in the OA of the different methods with different numbers of selected bands under the single split. As mentioned previously, the RBF-SVM employed all of the bands for classification; thus, the OA values of the RBF-SVM remained unchanged. For the Indian Pines dataset, the range of the number of selected bands was set to [2, 200]. For the Pavia University and University of Houston datasets, the ranges were [2,100] and [2, 140], respectively. In each range, the OA value was recorded at intervals of 20 bands.

As illustrated in Fig. 9, the OA values of the mRMR, MR-SVM, HM, and DCS increased sharply in the range of 2 to 20 for the three datasets. The DDCNN and ABCNN also exhibited this trend on the University of Houston dataset. This could have resulted from the fact that it is difficult for too few bands to support the classifiers to distinguish the samples of different classes. Compared to these methods, the TWCNN, SICNN, and BSD-GCN could obtain better classification results with few bands, which may be owing to the more discriminative bands that they selected and their spatial–spectral feature extraction abilities. The OA values of the TWCNN, SICNN, and BSD-GCN gradually improved as the number of selected bands increased further. When the number of selected bands exceeded 60, the growth of all of the OA curves on the Indian Pines dataset tended to be slow. The classification of most methods also tended to be stable when 40 bands were selected on the Pavia University and University of Houston

TABLE X
OVERALL ACCURACIES (%) OF FIVE-FOLD CROSS-VALIDATION WITH DIFFERENT NUMBERS OF SELECTED BANDS ON THE INDIAN PINES DATASET

Method	20	40	60	80	100
RBF-SVM	91.5±0.8	91.5±0.8	91.5±0.8	91.5±0.8	91.5±0.8
mRMR	86.4±0.9	89.7±0.6	90.5±0.4	90.6±0.7	90.7±0.3
TWCNN	96.6±0.4	97.8±0.3	98.0±0.2	98.2±0.1	98.3±0.1
MR-SVM	87.1±0.6	90.7±0.5	90.9±0.3	91.1±0.2	91.1±0.2
SICNN	95.7±0.6	96.9±1.0	97.0±0.8	97.2±0.3	97.2±0.2
DDCNN	93.7±1.9	95.2±1.0	96.5±0.8	96.7±0.6	96.8±0.3
ABCNN	93.5±1.7	94.8±0.4	95.9±0.2	96.1±0.2	96.3±0.2
HM	88.9±0.9	90.1±0.3	90.8±0.2	91.0±0.1	91.1±0.1
DCS	87.0±0.5	90.8±0.2	91.1±0.1	91.2±0.2	91.3±0.1
BSD-GCN	97.8±0.9	98.1±0.7	99.3±0.2	99.7±0.1	99.7±0.1

TABLE XI
OVERALL ACCURACIES (%) OF FIVE-FOLD CROSS-VALIDATION WITH DIFFERENT NUMBERS OF SELECTED BANDS ON PAVIA UNIVERSITY DATASET

Method	10	30	50	70	90
RBF-SVM	93.4±0.7	93.4±0.7	93.4±0.7	93.4±0.7	93.4±0.7
mRMR	88.2±1.3	90.9±0.6	91.1±0.5	91.4±0.3	91.7±0.2
TWCNN	97.3±0.4	98.6±0.2	98.9±0.1	99.0±0.1	99.2±0.1
MR-SVM	89.7±0.6	92.1±0.6	92.3±0.3	92.7±0.3	92.8±0.5
SICNN	96.5±1.1	98.5±0.8	98.9±0.4	99.0±0.2	99.1±0.1
DDCNN	94.3±1.4	97.9±0.5	98.1±0.7	98.2±0.6	98.2±0.7
ABCNN	94.7±0.9	97.1±0.8	98.5±0.4	98.7±0.2	98.8±0.2
HM	89.1±1.3	91.2±0.5	92.0±0.3	92.8±0.1	93.0±0.1
DCS	89.7±0.8	91.6±0.6	92.2±0.7	92.9±0.2	93.2±0.1
BSD-GCN	98.1±1.1	98.8±0.4	99.6±0.1	99.8±0.1	99.9±0.1

TABLE XII
OVERALL ACCURACIES (%) OF FIVE-FOLD CROSS-VALIDATION WITH DIFFERENT NUMBERS OF SELECTED BANDS ON THE UNIVERSITY OF HOUSTON DATASET

Method	20	40	60	80	100
RBF-SVM	98.6±0.3	98.6±0.3	98.6±0.3	98.6±0.3	98.6±0.3
mRMR	92.1±1.5	95.7±1.0	96.1±0.8	96.6±0.7	97.0±0.2
TWCNN	98.0±0.3	99.2±0.1	99.6±0.1	99.7±0.1	99.7±0.1
MR-SVM	95.4±0.8	96.2±0.7	97.3±0.2	97.5±0.1	97.5±0.1
SICNN	96.9±0.5	98.6±0.4	98.9±0.2	99.1±0.1	99.3±0.2
DDCNN	96.1±1.0	97.9±0.2	98.1±0.1	98.4±0.1	98.4±0.1
ABCNN	95.2±1.7	97.6±0.5	97.9±0.1	98.0±0.1	98.2±0.1
HM	92.1±1.2	96.5±0.8	97.2±1.1	97.9±0.3	98.3±0.2
DCS	94.8±0.5	97.2±0.3	97.6±0.3	98.0±0.1	98.5±0.2
BSD-GCN	98.9±0.4	99.7±0.1	99.7±0.1	99.8±0.1	99.8±0.1

datasets. Compared to the SICNN and TWCNN, the BSD-GCN maintained the best OA performance under most conditions. The BSD-GCN is a more suitable choice for selecting a more discriminative band subset.

Tables X-XII display the classification results with different numbers of selected bands under five-fold cross-validation. The range of the number of selected bands in the Indian Pines and University of Houston datasets was set to [20, 100] with intervals of 20 bands. The range of the number of selected bands on the Pavia University dataset was [10, 90] with intervals of 20 bands. As indicated in Tables X-XII, the classification performance of all methods improved because a larger proportion of training samples was provided in the multi-fold cross-validation. The classification performance of each method improved with the increase in the selected bands. For the Pavia University dataset, there was little room

for improvement in the OA in most methods after 30 bands were selected. Further improvement was also limited after 60 bands were selected on the Indian Pines dataset and 40 bands were selected on the University of Houston dataset. Compared to the CNN-based methods, the OA values of the mRMR, MR-SVM, HM, and DCS changed more dramatically with different selected bands. For example, the OA value of the mRMR with 100 selected bands was 4.9% higher than that with 20 selected bands on the University of Houston dataset. Although the CNN-based methods exhibited relatively smooth changes, significant differences in the classification performance could still be observed with different selected bands.

In summary, 60, 30, and 40 were determined and recommended as the numbers of selected bands on the Indian Pines, Pavia University, and University of Houston datasets, respectively.

5.6. Analysis of the Selected Spectral Bands

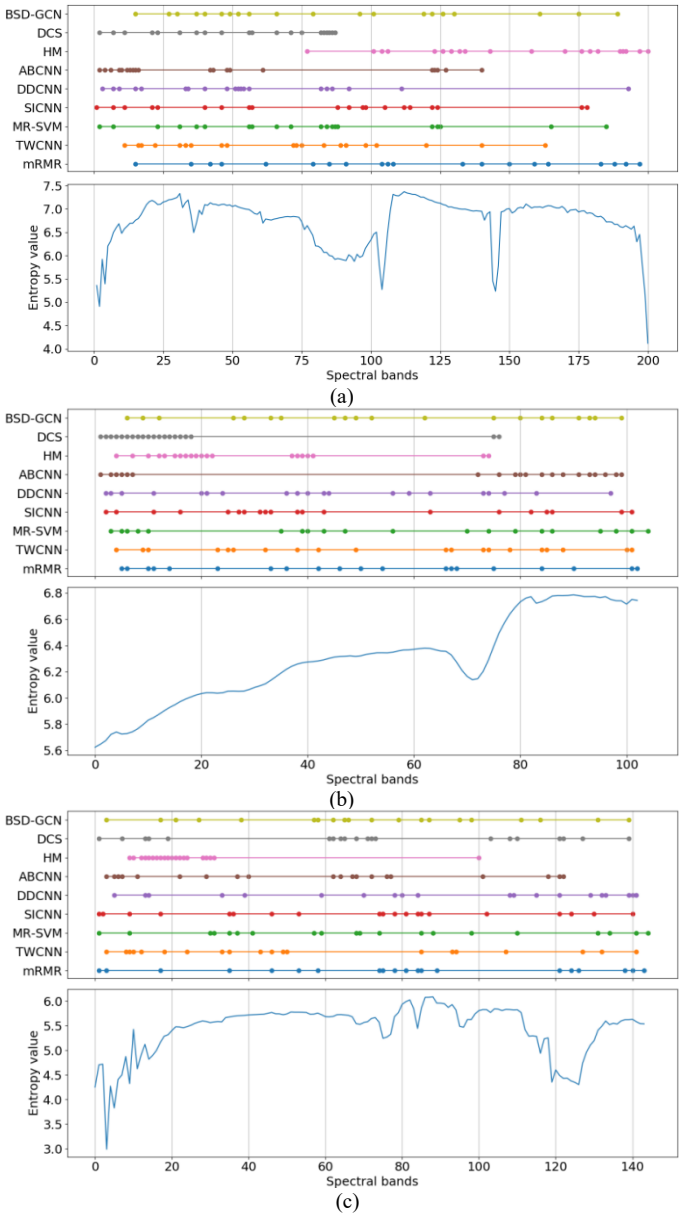

Fig. 10 The 20 spectral bands selected by different methods in above subfigures and entropy value of each band in below subfigures on (a) Indian Pines, (b) Pavia University, and (c) University of Houston datasets.

In this section, the selected bands were analyzed based on the location and entropy value of each band. Shannon entropy was used to evaluate the amount of information in the selected spectral bands. The upper parts of Fig. 10 (a), (b), and (c) depict the locations of the selected spectral bands, where each point corresponds to the location of each selected band. The lower parts of Fig. 10 (a), (b), and (c) depict the entropy values of the entire spectral bands. To show the results clearly, the band selection methods were

set to select 20 bands. As illustrated in Fig. 10, the bands selected by the DCS, HM, and ABCNN were located more densely, which would cause greater information redundancy owing to the high correlation between adjacent bands. The BSD-GCN, DDCNN, SICNN, and MR-SVM selected bands in a wider and more scattered range. However, the DDCNN and MR-SVM selected more adjacent bands than the BSD-GCN. Spectral bands with larger entropy values contain more information. The mRMR, DCS, and TWCNN selected several spectral bands with low entropy values, which was likely to limit the classification performance. Compared to the other methods, the BSD-GCN tended to select bands with larger entropy values and achieved more discrete band distribution.

5.7. Sensitivity to Different Proportions of Training Samples

TABLE XIII
OVERALL ACCURACIES (%) WITH DIFFERENT PROPORTIONS OF TRAINING SAMPLES ON THE INDIAN PINES DATASET

Method	1%	3%	5%	10%	20%	40%
RBF-SVM	63.7±3.1	69.8±1.8	77.5±0.4	80.1±0.9	83.6±0.8	89.6±0.3
mRMR	62.6±3.2	67.4±1.0	73.5±0.7	78.3±0.8	81.9±1.0	88.3±0.2
TWCNN	60.2±2.4	77.4±2.3	92.4±1.0	94.5±1.2	96.2±0.9	97.2±0.1
MR-SVM	63.3±2.0	68.0±1.7	75.8±1.0	77.6±1.1	82.0±0.6	88.6±0.2
SICNN	58.5±1.1	75.2±2.4	87.6±1.0	90.2±1.2	92.5±1.0	96.5±0.2
DDCNN	55.7±2.6	71.8±2.3	82.7±1.0	88.3±1.1	91.4±0.6	95.1±0.5
ABCNN	54.3±1.8	69.5±2.0	81.5±1.7	87.9±1.3	90.3±1.7	92.9±0.9
HM	60.3±2.0	65.6±1.9	74.9±1.5	78.3±1.4	81.1±1.4	88.9±0.6
DCS	61.2±2.7	67.8±2.0	75.2±1.2	79.4±1.6	81.2±1.3	89.0±0.3
BSD-GCN	70.5±1.9	86.7±1.5	97.7±0.4	98.0±0.2	98.3±0.2	99.1±0.1

TABLE XIV
OVERALL ACCURACIES (%) WITH DIFFERENT PROPORTIONS OF TRAINING SAMPLES ON THE PAVIA UNIVERSITY DATASET

Method	1%	3%	5%	10%	20%	40%
RBF-SVM	84.2±1.3	88.0±0.5	88.7±0.2	90.1±0.3	90.7±0.2	91.5±0.1
mRMR	83.5±1.1	86.1±0.4	87.4±0.5	88.3±1.0	89.6±0.8	90.8±0.4
TWCNN	90.4±2.1	95.9±0.8	96.5±1.0	97.3±0.2	97.8±0.3	99.0±0.1
MR-SVM	84.0±0.9	85.7±0.6	86.3±0.7	88.0±0.3	89.7±0.5	90.6±0.3
SICNN	81.5±1.7	95.3±0.9	96.2±1.1	96.9±1.0	97.3±0.9	98.7±0.2
DDCNN	79.2±1.0	90.3±0.7	94.3±1.3	95.1±0.6	96.2±0.7	96.8±0.3
ABCNN	79.6±0.8	91.3±0.1	94.5±0.4	95.3±0.7	96.9±0.8	97.0±0.2
HM	82.9±1.6	85.6±0.6	86.7±0.9	88.4±1.2	88.9±0.7	90.6±0.1
DCS	83.2±0.9	86.2±0.6	87.1±1.0	89.0±0.3	90.3±0.6	90.9±0.1
BSD-GCN	91.7±1.2	98.1±0.4	98.7±0.2	99.0±0.1	99.2±0.1	99.2±0.1

TABLE XV
OVERALL ACCURACIES (%) WITH DIFFERENT PROPORTIONS OF TRAINING SAMPLES ON THE UNIVERSITY OF HOUSTON DATASET

Method	1%	3%	5%	10%	20%	40%
RBF-SVM	77.6±0.6	86.7±1.2	92.1±0.7	94.8±0.5	96.0±0.4	97.5±0.3
mRMR	74.8±1.3	83.7±1.0	88.9±0.6	91.5±1.1	94.6±0.7	95.2±0.3
TWCNN	78.5±2.0	87.4±1.1	94.1±1.3	95.6±0.7	97.0±0.4	98.9±0.1
MR-SVM	75.0±0.4	84.3±0.9	90.7±0.8	93.1±0.9	95.5±0.8	96.7±0.4
SICNN	76.8±1.2	86.9±0.8	93.8±0.7	95.0±0.8	96.4±0.5	97.1±0.2
DDCNN	72.0±2.6	84.9±1.0	92.5±0.9	93.8±0.7	95.5±0.8	96.4±0.3
ABCNN	69.6±2.9	85.3±1.0	91.6±0.4	93.2±0.9	94.4±0.5	95.7±0.8
HM	73.2±2.0	81.0±1.5	87.1±1.3	90.5±1.1	93.9±0.7	95.3±0.3
DCS	75.1±1.4	84.9±0.8	90.3±0.7	92.3±0.7	94.3±0.8	96.0±0.2
BSD-GCN	81.2±1.7	90.2±0.9	96.7±0.3	97.6±0.3	97.9±0.2	99.0±0.1

In this section, the classification performance of all methods with different proportions of training samples was analyzed. The corresponding experimental results for the three HSI datasets are listed in Tables XIII-XV.

The classification performance of all methods improved with an increase in the training samples. In the Indian Pines dataset, the RBF-SVM, mRMR, MR-SVM, HM, and DCS outperformed the CNN-based methods when only 1% of the labeled samples were used for training. In this case, compared to the other methods, the BSD-GCN exhibited the best performance and achieved an improvement of at least 7.2% in terms of the OA. There were more available samples for training in the Pavia University and the University of Houston datasets. When the training samples were sufficient, the SICNN, TWCNN, and BSD-GCN exhibited obvious improvements over the traditional methods. Compared to the other band selection methods, the BSD-GCN continuously maintained the best classification performance with different proportions of training samples.

5.8. Effectiveness Analysis on the Salinas Dataset

The Salinas dataset was obtained by an airborne visible/infrared imaging spectrometer sensor, which was the same as that of the Indian Pines dataset. The spatial resolution in the Salinas dataset is 3.7 m, which is much higher than that of the Indian Pines dataset. This dataset covers the Salinas Valley in California, USA, and includes 512 × 217 pixels with 224 spectral bands. After 20 water absorption bands were removed, 204 bands remained for further processing. The Salinas dataset contains 16 land-cover classes. The false-color composite image (bands 69, 27, 11) and the ground truth of the Salinas dataset are presented in Fig. 11. For this dataset, 3% and 6% of the labeled samples were randomly selected as the labeled and unlabeled training sets, respectively. The number of selected bands was set to 60. The remaining labeled samples were used for testing. This dataset is available on the website http://www.ehu.eu-s/ccwintco/index.php/Hyperspectral_Remote_Sensing_Scenes. The classification performance of the different methods is shown in Table XVI.

Fig. 11 False-color composite image and ground truth of Salinas dataset.

TABLE XVI
CLASSIFICATION RESULTS OF RBF-SVM, MRMR, TWCNN, MR-SVM, SICNN, DDCNN, ABCNN, HM, DCS, AND BSD-GCN ON SALINAS DATASET

Class	RBF-SVM	mRMR	TWCNN	MR-SVM	SICNN	DDCNN	ABCNN	HM	DCS	BSD-GCN
OA (%)	91.3±0.5	89.1±0.5	96.6±1.0	88.2±2.0	91.7±2.1	90.8±2.0	90.5±1.1	89.5±0.3	90.3±1.0	98.4±0.1
AA (%)	94.6±0.3	93.4±0.7	97.1±0.8	90.7±1.6	94.2±1.8	93.1±1.1	92.4±1.0	92.0±0.2	92.1±1.5	99.0±0.1
Kappa (%)	89.2±0.3	88.7±0.3	92.2±1.5	87.3±1.7	90.5±0.7	88.2±1.3	89.8±0.9	88.2±0.3	89.4±1.2	98.2±0.3

Compared to the SICNN and DDCNN, the TWCNN exhibited an obvious improvement in the classification performance. All of the methods achieved higher AA values than OA values. This may be because the samples of each class in the Salinas dataset are more evenly distributed. Compared to the other methods, the BSD-GCN exhibited improvements of at least 1.8%, 1.9%, and 6.0% in terms of the OA, AA, and Kappa, respectively.

5.9. Comparison with Other State-of-the-Art Dimensionality Reduction Methods

In this section, BSD-GCN was compared with two other state-of-the-art dimensionality reduction

TABLE XVII
CLASSIFICATION RESULTS OF S^3-PCA, MRMR, TWCNN, MR-SVM, SICNN, DDCNN, ABCNN, HM, ASPS_MN, DCS, AND BSD-GCN ON THE INDIAN PINES DATASET

Class	S^3-PCA	mRMR	TWCNN	MR-SVM	SICNN	DDCNN	ABCNN	HM	ASPS_MN	DCS	BSD-GCN
OA (%)	94.2±0.5	73.5±0.7	92.4±1.0	75.8±1.0	87.6±1.0	82.7±1.0	81.5±1.7	74.9±1.5	76.0±1.3	75.2±1.2	97.7±0.4
AA (%)	95.1±0.3	65.0±2.1	90.1±2.4	69.3±2.4	82.7±2.6	78.2±2.1	74.2±2.0	65.4±1.7	71.0±1.0	69.0±3.0	95.3±2.8
Kappa (%)	93.7±0.6	69.8±0.8	91.3±1.1	73.3±1.7	85.9±1.1	80.4±1.5	78.8±1.4	71.3±1.7	73.1±0.6	72.7±1.4	97.5±0.9

TABLE XVIII
CLASSIFICATION RESULTS OF S^3-PCA, MRMR, TWCNN, MR-SVM, SICNN, DDCNN, ABCNN, HM, ASPS_MN, DCS, AND BSD-GCN ON THE PAVIA UNIVERSITY DATASET

Class	S^3-PCA	mRMR	TWCNN	MR-SVM	SICNN	DDCNN	ABCNN	HM	ASPS_MN	DCS	BSD-GCN
OA (%)	95.1±0.2	86.1±0.4	95.9±0.8	85.7±0.6	95.3±0.9	90.3±0.7	91.3±0.1	85.6±0.6	87.0±0.9	86.2±0.6	98.1±0.4
AA (%)	93.0±1.0	80.4±1.2	92.5±1.5	83.1±1.5	93.1±0.7	83.9±0.7	88.1±0.4	81.8±0.6	84.1±1.1	83.5±0.8	97.5±0.6
Kappa (%)	92.9±0.7	81.0±0.8	94.5±1.0	82.8±1.0	93.7±0.7	87.1±0.8	88.5±1.3	80.5±0.9	82.3±0.9	82.5±0.8	96.3±0.8

TABLE XIX
CLASSIFICATION RESULTS OF S^3-PCA, MRMR, TWCNN, MR-SVM, SICNN, DDCNN, ABCNN, HM, ASPS_MN, DCS, AND BSD-GCN ON THE UNIVERSITY OF HOUSTON DATASET

Class	S^3-PCA	mRMR	TWCNN	MR-SVM	SICNN	DDCNN	ABCNN	HM	ASPS_MN	DCS	BSD-GCN
OA (%)	95.8±0.5	88.9±0.6	94.1±1.3	90.7±0.8	93.8±0.7	92.5±0.9	91.6±0.4	87.1±1.3	90.9±0.5	90.3±0.7	96.7±0.3
AA (%)	96.0±0.2	90.2±1.4	94.1±1.4	91.7±1.3	92.7±1.7	92.6±1.1	91.2±0.9	86.2±1.0	88.6±0.8	89.5±0.7	97.1±0.4
Kappa (%)	93.1±1.3	88.1±1.1	93.7±1.1	91.0±0.9	93.4±0.9	91.9±1.1	90.9±0.8	86.0±1.4	87.3±0.9	89.5±0.8	96.4±0.4

methods, namely S^3-PCA [14] and the adaptive subspace partition strategy with minimizing noise (ASPS_MN) [46]. The experimental results for the three datasets are listed in Tables XVII-XIX.

The S^3-PCA achieved better classification performance than the DCS and SICNN owing to its effective spatial segmentation and use of spectral information. Compared to S^3-PCA, the BSD-GCN performed better in terms of the OA, AA, and Kappa for all three datasets. The ASPS_MN achieved better classification performance than the mRMR and HM by using an adaptive subspace partition to select informative and distinctive bands. The BSD-GCN outperformed the ASPS_MN by at least 5.8%, 8.5%, and 9.1% in terms of the OA, AA, and Kappa on the three datasets.

5.10. Ablation Experiments in BSD-GCN

Ablation experiments were conducted to investigate the effectiveness of the spectral graphs, spatial graphs, and full-band loss in the BSD-GCN. Specifically, the BSD-GCN has three core contributions, namely the spectral and spatial graphs in the attention and trunk branches, and the full-band loss in the optimization. To analyze the contribution of each component, the classification results of the BSD-GCN without different components are listed in Table XX. Furthermore, the performance of the original BSD-GCN is presented to provide a more intuitive comparison.

TABLE XX
CLASSIFICATION PERFORMANCE IN ABLATION EXPERIMENTS

Dataset	**Conditions**	**OA (%)**	**AA (%)**	**Kappa (%)**
Indian Pines	BSD-GCN without spectral graphs	92.4±0.8	91.3±0.4	91.6±1.2
	BSD-GCN without spatial graphs	85.7±1.2	84.6±1.1	84.0±0.9
	BSD-GCN without full-band loss	94.2±0.6	93.1±0.7	93.5±0.6
	BSD-GCN	97.7±0.4	95.3±2.8	97.5±0.9
Pavia University	BSD-GCN without spectral graphs	95.2±0.6	92.4±1.3	92.3±1.4
	BSD-GCN without spatial graphs	88.1±0.5	86.0±0.7	86.2±0.8
	BSD-GCN without full-band loss	96.9±1.0	94.8±0.9	95.1±0.9
	BSD-GCN	98.1±0.4	97.3±0.6	96.3±0.8
University of Houston	BSD-GCN without spectral graphs	93.6±0.9	92.0±1.5	92.2±0.6
	BSD-GCN without spatial graphs	87.2±2.4	85.1±1.9	83.7±1.7
	BSD-GCN without full-band loss	94.6±2.1	94.0±2.3	93.9±2.5
	BSD-GCN	96.7±0.3	97.1±0.4	96.4±0.4

As indicated in Table XX, when the spatial graphs were removed and the spectral graphs were retained, the performance of the BSD-GCN decreased significantly. This means that the pure spectral characteristics were not sufficient to provide discriminative information for the different classes. The proposed method could achieve high performance when the spatial graphs were retained. This may be

owing to the fact that the neighborhood of samples in the local regions was more likely to be of the same class. Therefore, spatial aggregation enhanced the feature discrimination. Thus, it is recommended that spatial graphs be selected and used in the BSD-GCN. When the full-band loss was removed, the performance of the BSD-GCN declined because the updating of the parameters was limited. It is difficult for the BSD-GCN to select the most discriminative bands. Overall, the spatial graphs, spectral graphs, and soft-shifting optimization complemented one another to enable the proposed method to achieve outstanding classification results.

5.11. Analysis of Hyperparameters in BSD-GCN

In this section, the effects of the training epoch number, learning rate, and batch size on the classification performance were investigated. The corresponding experimental results are presented in Fig. 12.

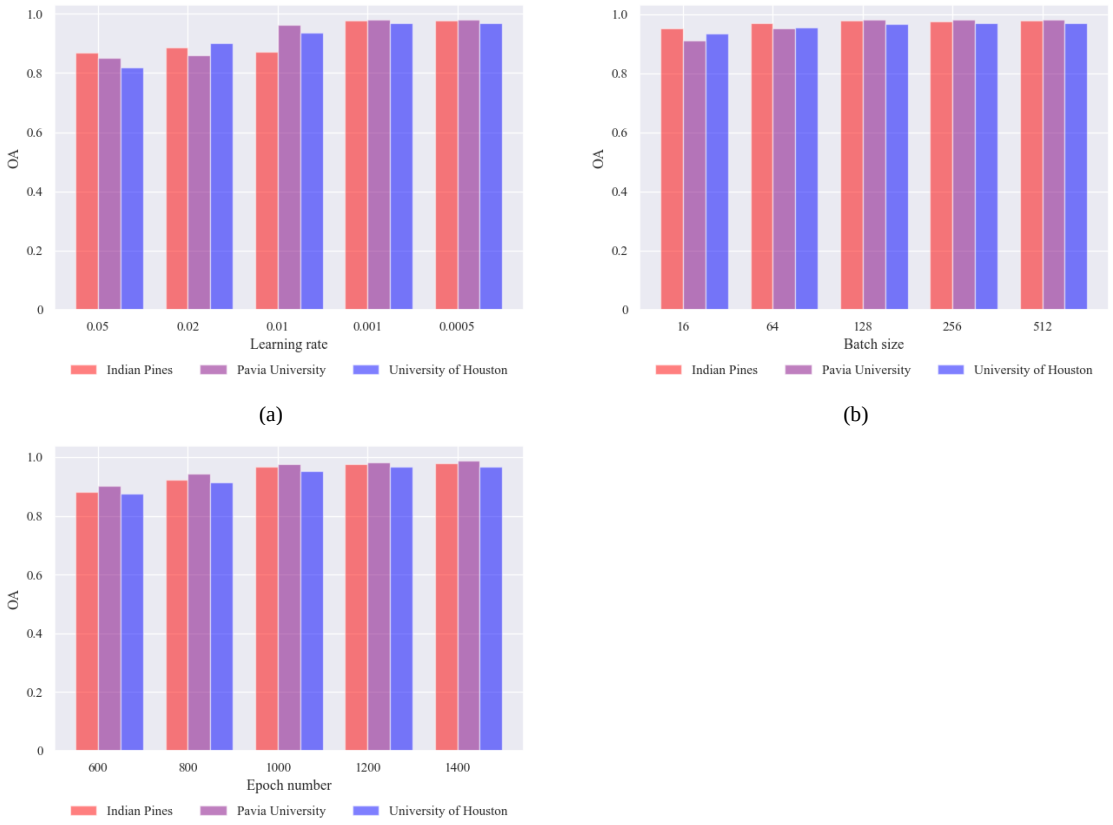

Fig. 12 Classification performance of BSD-GCN under different values of (a) learning rate, (b) batch size, and (c) epoch number.

As illustrated in Fig. 12, the change in the learning rate and epoch number had a significant influence on the classification performance of the BSD-GCN. If the learning rate is too large, the BSD-GCN may not converge. Conversely, if the learning rate is too small, the convergence speed will be too slow. Thus, a learning rate of 0.001 was set. It is difficult for the BSD-GCN to converge if the epoch number is too small. Meanwhile, too many iterations would lead to wasted time. Therefore, 1000 was selected as the training epoch number. The effect of the batch size was weaker than that of the other two parameters. As illustrated in Fig. 12 (b), satisfactory performance could be achieved with a batch size of 128. An excessive batch size would incur an additional cost in calculating the sample-based dual graphs and failed to improve the performance significantly. Thus, 128 was eventually set as the batch size.

6. Conclusion

In this paper, a novel BSD-GCN method has been proposed for the band selection of HSIs. To make full use of the correlations among bands in adjacent and long-range positions, band-based dual graphs are constructed to transform the band information into non-Euclidean space. Thereafter, graph

convolution is employed to integrate the features among different bands to generate the attention map. To retain the most discriminative band subset, the attention map is sparsely constrained and inserted into the first layer of the trunk branch. Spatial–spectral graph convolution modules based on sample-based dual graphs are stacked and densely connected for feature extraction and classification to evaluate the quality of the selected bands. To alleviate the difficulty in updating the parameters of the unselected bands, soft-shifting optimization is implemented to assist in updating the unselected bands by adding the loss of full bands, which optimizes the BSD-GCN in an end-to-end manner. The experimental results on three representative HSI datasets demonstrated the effectiveness of the proposed BSD-GCN method. In the future, we will focus on designing a GCN model for the band selection of specific categories. Moreover, band selection that is specific to semantic segmentation and object detection problems will be further investigated.

7. Acknowledgments

This work was supported in part by the National Natural Science Foundation of China under Grant 61871306, Grant 61836009, Grant 61772400, Grant 61773304, Grant 61703328, and Grant 61703328, by the Innovation Capability Support Program of Shaanxi (Program No. 2021KJXX-08), by the Xi'an Association for Science and Technology Youth Talent Support Program Project under Grant 095920201301, by the Innovation Fund of Shanghai Aerospace Science and Technology under Grant SAST2019-093, by the Fundamental Research Funds for the Central Universities under Grant JB211901, and by the Aeronautical Science Fund of China under Grant 2019ZC081002.

8. References

[1] I. Makki, R. Younes, C. Francis, T. Bianchi, and M. Zucchetti, “A survey of landmine detection using hyperspectral imaging,” *ISPRS J. Photogramm. Remote Sens.*, vol. 124, pp. 40–53, Feb. 2017.

[2] A. J. Brown, M. R. Walter, and T. J. Cudahy, “Hyperspectral imaging spectroscopy of a Mars analogue environment at the North Pole Dome, Pilbara Craton, Western Australia,” *Aust. J. Earth Sci.*, vol. 52, no. 3, pp. 353–364, Jun. 2005.

[3] C. McCann, K. S. Repasky, R. L. Lawrence, and S. Powell, “Multi-temporal mesoscale hyperspectral data of mixed agricultural and grass-land regions for anomaly detection,” *ISPRS J. Photogramm. Remote Sens.*, vol. 131, pp. 121–133, Sep. 2017.

[4] S. Schneider, R. J. Murphy, and A. Melkumyan, “Evaluating the performance of a new classifier—The GP-OAD: A comparison with existing methods for classifying rock type and mineralogy from hyper-spectral imagery,” *ISPRS J. Photogramm. Remote Sens.*, vol. 98, pp. 145–156, Dec. 2014.

[5] Y. Zhang, W. Li, H. Li, R. Tao, and Q. Du, “Discriminative marginalized least squares regression for hyperspectral image classification,” *IEEE Trans. Geosci. Remote Sens*, vol. 58, no. 5, pp. 3148-3161, May. 2020 .

[6] W. Li, Q. Du, and B. Zhang, ‘‘Combined sparse and collaborative representation for hyperspectral target detection,’’ *Pattern Recognit.*, vol. 48, no. 12, pp. 3904–3916, 2015.

[7] M. Wang, C. Wu, L Wang, D Xiang, and X Hua, “A feature selection approach for hyperspectral image based on modified ant lion optimizer,” *Knowledge-Based Syst*., vol. 168, pp. 39–48, 2019.

[8] S. Prasad and L. M. Bruce, “Limitations of principal components analysis for hyperspectral target recognition,” *IEEE Geosci. Remote Sens. Lett.*, vol. 5, no. 4, pp. 625–629, Oct. 2008.

[9] W. Li, S. Prasad, J. E. Fowler, and L. M. Bruce, “Locality preserving dimensionality reduction and classification for hyperspectral image analysis,” *IEEE Trans. Geosci. Remote Sens.*, vol. 50, no. 4, pp. 1185–1198, Apr. 2012.

[10] B. Raducanu and F. Dornaika, “A supervised non-linear dimensionality reduction approach for manifold learning,” *Pattern Recognit.*, vol. 45, no. 6, pp. 2432–2444, 2012.

[11] J. Zabalza *et al.* “Novel folded-PCA for improved feature extraction and data reduction with hyperspectral imaging and SAR in remote sensing,” *ISPRS-J. Photogramm. Remote Sens.*, vol. 93, pp. 112–122, 2014.

[12] J. Jiang, J. Ma, C. Chen, Z. Wang, Z. Cai, and L. Wang, “SuperPCA: A superpixelwise PCA approach for unsupervised feature extraction of hyperspectral imagery,” *IEEE Trans. Geosci. Remote Sens.*, vol. 56, no. 8, pp. 4581–4593, Aug. 2018.

[13] L. Zhang, H. Su, and J. Shen, “Hyperspectral dimensionality reduction based on multiscale superpixelwise kernel principal component analysis,” *Remote Sens.*, vol. 11, no. 10, p. 1219, May 2019.

[14] X. Zhang, X. Jiang, J. Jiang, Y. Zhang, X. Liu and Z. Cai, "Spectral-spatial and superpixelwise pca for unsupervised feature extraction of hyperspectral imagery," *IEEE Trans. Geosci. Remote Sens*, doi: 10.1109/TGRS.2021.3057701.

[15] D. Cai, X. He, and J. Han, “Semi-supervised discriminant analysis,” in *Proc. IEEE 11th Int. Conf. Comput. Vis.*, 2007, pp. 1–7.

[16] W. Liao, A. Pižurica, P. Scheunders,W. Philips, and Y. Pi, “Semisupervised local discriminant analysis for feature extraction in hyperspectral images,” *IEEE Trans. Geosci. Remote Sens.*, vol. 51, no. 1, pp. 184–198, Jan. 2013.

[17] R. Luo, W. Liao, X. Huang, Y. Pi, and W. Philips, “Feature extraction of hyperspectral images with semisupervised graph learning,” *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 9, no. 9, pp. 4389–4399, Sep. 2016.

[18] F. Luo, T. Guo, Z. Lin, J. Ren and X. Zhou, "Semisupervised hypergraph discriminant learning for dimensionality reduction of hyperspectral image," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens*, vol. 13, pp. 4242-4256, 2020.

[19] Y. Cao, X. Liu, and Z. Cai, “BS-Nets: An end-to-end framework for band selection of hyperspectral image,” *IEEE Trans. Geosci. Remote Sens.*,vol. 58, no. 3, pp. 1969–1984, Mar. 2020.

[20] B. Barman, and S. Patra, “Variable precision rough set based unsupervised band selection technique for hyperspectral image classification,” *Knowledge-Based Syst.*, vol. 193, pp. 105414, 2020.

[21] L. Song, A. Smola, A. Gretton, J. Bedo, and K. Borgwardt, “Feature selection via dependence maximization,” *J. Mach. Learn. Res.*, vol. 13, pp. 1393–1434, May 2012.

[22] W. Sun and Q. Du, "Hyperspectral band selection: a review," in *IEEE Geoscience and Remote Sensing Magazine*, vol. 7, no. 2, pp. 118-139, June 2019.
[23] J. Feng, L. Jiao, X. Zhang, T. Sun, "Hyperspectral band selection based on trivariate mutual information and clonal selection, " *IEEE Trans. Geosci. Remote Sens.*,vol. 52, no. 7, pp. 4092–4105, July. 2014.
[24] H. Peng, F. Long, and C. Ding, "Feature selection based on mutual information criteria of max-dependency, max-relevance, and min-redundancy," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 27, no. 8,pp. 1226–1238, Aug. 2005.
[25] X. Bai, Z. Guo, Y. Wang *et al*., "Semisupervised hyperspectral band selection via spectral–spatial hypergraph model," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 8, no. 6, pp. 2774-2783, 2015.
[26] P. Ghamisi, M. S. Couceiro, and J. A. Benediktsson, "A novel feature selection approach based on FODPSO and SVM," *IEEE Trans. Geosci. Remote Sens,* vol. 53,no. 5, pp. 2935–2947, May 2015.
[27] X. Cao, C. Wei, Y. Ge, J. Feng, J. Zhao, and L. Jiao, "Semi-Supervised Hyperspectral Band Selection Based on Dynamic Classifier Selection," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 12, no. 4, pp. 1289-1298, 2019.
[28] P. Ghamisi, Y. Chen, and X. X. Zhu, "A self-improving convolution neural network for the classification of hyperspectral data," *IEEE Geosci. Remote Sens. Lett.*, vol. 13, no. 10, pp. 1537–1541, Oct. 2016.
[29] I. Guyon, J. Weston, S. Barnhill, and V. Vapnik, "Gene selection for cancer classification using support vector machines," *Mach. Learn.*, vol. 46, nos. 1–3, pp. 389–422, 2002.
[30] R. Zhang, and J. Ma, "Feature selection for hyperspectral data based on recursive support vector machines," *Int. J. Remote Sens.*, vol. 30, no. 14, pp. 3669–3677, 2009.
[31] W. Sun, G. Yang, J. Peng and Q. Du, "Hyperspectral band selection using weighted kernel regularization," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens*, vol. 12, no. 9, pp. 3665-3676, Sept. 2019.
[32] Q. Wang, Q. Li and X. Li, "A fast neighborhood grouping method for hyperspectral band selection," *IEEE Trans. Geosci. Remote Sens*, vol. 59, no. 6, pp. 5028-5039, June 2021.
[33] X. Zhang, Y. He, N. Zhou, and Y. Zheng, "Semisupervised dimensionality reduction of hyperspectral images via local scaling cut criterion," *IEEE Geosci. Remote Sens. Lett.*, vol. 10, no. 6, pp. 1547–1551, Nov. 2013.
[34] S. Jia, G. Tang, J. Zhu and Q. Li, "A novel ranking-based clustering approach for hyperspectral band selection," *IEEE Trans. Geosci. Remote Sens*, vol. 54, no. 1, pp. 88-102, Jan. 2016.
[35] S. Jia, *et al*., "3-D gaussian–gabor feature extraction and selection for hyperspectral imagery classification," *IEEE Trans. Geosci. Remote Sens*, vol. 57, no. 11, pp. 8813-8826, Nov. 2019.
[36] Q. Wang, F. Zhang and X. Li, "Optimal clustering framework for hyperspectral band selection," *IEEE Trans. Geosci. Remote Sens*, vol. 56, no. 10, pp. 5910-5922, Oct. 2018.
[37] B. Xu, X. Li, W. Hou, Y. Wang and Y. Wei, "A Similarity-based ranking method for hyperspectral band selection," *IEEE Trans. Geosci. Remote Sens*, doi: 10.1109/TGRS.2020.3048138.
[38] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, pp. 436–444, May 2015.
[39] J. Feng, X. Wu, R. Shang, C. Sun, J. Li, L. Jiao, and X. Zhang, "Attention multibranch convolutional neural network for hyperspectral image classification based on adaptive region search," *IEEE Trans. Geosci. Remote Sens.*, 2020, doi: 10.1109/TGRS.2020.3011943.
[40] J. Feng, D. Li, J. Chen, X. Zhang, X. Tang, and X. Wu, ''Hyperspectral band selection based on ternary weight convolutional neural network,'' in *Proc. IEEE Int. Geosci. Remote Sens. Symp.*, Jul. 2019, pp. 3804–3807
[41] J. Wang, J. Zhou, and W. Huang, ",Attend in bands: Hyperspectral band weighting and selection for image classification," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*vol. 12, no. 12, pp. 4712–4727, Dec. 2019.
[42] T. N. Kipf, and M. Welling, "Semi-supervised classification with graph convolutional networks," in *Proc. Int. Conf. Learn. Represent. (ICLR)*, 2017, pp. 1–14.
[43] S. Wan, C. Gong, P. Zhong, B Du, L. Zhang, and J. Yang, "Multiscale dynamic graph convolutional network for hyperspectral image classification," *IEEE Trans. Geosci. Remote Sens*, vol. 58,no. 5, pp. 3162–3177, May 2020.
[44] D. Hong, L. Gao, "Graph convolutional networks for hyperspectral image classification," 2020, *arXiv:2008.02457* [Online]. Available: https://arxiv.org/cs/2020.02457
[45] S. Wang, C. Gong, P. Zhong, S. Pan, G. Li, and J. Yang, "Hyperspectral image classification with context-aware dynamic graph convolutional network," 2019, *arXiv:1909.11953* [Online]. Available: https://arxiv.org/cs/19.9.11953
[46] Q. Wang, Q. Li and X. Li, "Hyperspectral band selection via adaptive subspace partition strategy," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens*, vol. 12, no. 12, pp. 4940-4950, Dec. 2019.
[47] Y. Zhan, D. Hu, H. Xing, and X. Yu, "Hyperspectral band selection based on deep convolutional neural network and distance density," *IEEE Geosci. Remote Sens. Lett.*, vol. 14, no. 12, pp. 2365–2369, Dec. 2017.
[48] J. Feng, D. Li, J. Gu, X. Cao, R. Shang, X. Zhang, and J. Chen, "Deep reinforcement learning for semisupervised hyperspectral band selection," *IEEE Trans. Geosci. Remote Sens*, doi: 10.1109/TGRS.2021.3049372.
[49] P. R. Lorenzo, L. Tulczyjew, M. Marcinkiewicz, and J. Nalepa, "Hyperspectral band selection using attention-based convolutional neural networks," *IEEE Access*, vol. 8, pp. 42384-42403, 2020.
[50] F. Scarselli, M. Gori, A. C. Tsoi, M. Hagenbuchner, and G. Monfardini, "The Graph Neural Network Model," *IEEE Trans. Neural Networks*, vol. 20, no. 1, pp. 61-80, Jan. 2009.
[51] J. Gilmer, S. S. Schoenholz, P. F. Riley, O. Vinyals, and G. E. Dahl, "Neural message passing for quantum chemistry," *Proc. of ICML*, 2017, pp. 1263–1272.
[52] J. Qiu, J. Tang, H. Ma, Y. Dong, K. Wang, and J. Tang, "Deepinf: Social influence prediction with deep learning," *in Proc. of KDD. ACM*, 2018, pp. 2110–2119.
[53] A. Sandryhaila and J. M. Moura, "Discrete signal processing on graphs," *IEEE transactions on signal processing*, vol. 61, no. 7, pp. 1644–1656, 2013
[54] Z. Wu, S. Pan, F. Chen, G. Long, C. Zhang and P. S. Yu, "A comprehensive survey on graph neural networks," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 32, no. 1, pp. 4-24, Jan. 2021.
[55] M. Defferrard, X. Bresson, and P. Vandergheynst, "Convolutional neural networks on graphs with fast localized spectral filtering," in *NIPS*, 2016, pp. 3844–3852.
[56] Q. Li, Z. Han, and X.-M. Wu. "Deeper insights into graph convolutional networks for semi-supervised learning". In *Thirty-Second AAAI Conference on Artificial Intelligence*, 2018.
[57] K. He, X. Zhang, S. Ren, and J. Sun, "Deep Residual Learning for Image Recognition," in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016, pp. 770-778.
F. Melgani and L. Bruzzone, "Classification of hyperspectral remote sensing images with support vector machines," *IEEE Trans. Geosci.Remote Sens.*, vol. 42, no. 8, pp. 1778–1790, Aug. 2004.