Semisupervised Cross-Scale Graph Prototypical Network for Hyperspectral Image Classification

Bobo Xi, *Graduate Student Member, IEEE*, Jiaojiao Li, *Member, IEEE*, Yunsong Li, *Member, IEEE*, Rui Song, *Member, IEEE*, Yuchao Xiao, Qian Du, *Fellow, IEEE*, and Jocelyn Chanussot, *Fellow, IEEE*

***Abstract*— In practice, the acquirement of labeled samples for hyperspectral image (HSI) is time-consuming and labor-intensive. It frequently induces the trouble of model overfitting and performance degradation for the supervised methodologies in HSI classification (HSIC). Fortunately, semisupervised learning can alleviate this deficiency, and graph convolutional network (GCN) is one of the most effective semisupervised approaches, which propagates the node information from each other in a transductive manner. In this study, we propose a cross-scale graph prototypical network (X-GPN) to achieve semisupervised high-quality HSIC. Specifically, considering the multiscale appearance of the land covers in the same remotely captured scene, we involve the neighborhoods of different scales to construct the adjacency matrices and simultaneously design a multibranch framework to investigate the abundant spectral–spatial features through graph convolutions. Furthermore, to exploit the complementary information between different scales, we simply employ the standard 1-D convolution to excavate the dependence of the intranode and concatenate the output with the features generated from other scales. Intuitively, different branches for various samples should have different importance to predict their categories. Thus, we develop a self-branch attentional addition (SBAA) module to adaptively highlight the most critical features produced by multiple branches. In addition, different from previous GCN for HSIC, we devise an innovative prototypical layer comprising a distance-based cross-entropy (DCE) loss function and a novel temporal entropy-based regularizer (TER), which can enhance the discrimination and representativeness of the node features and prototypes actively. Extensive experiments demonstrate that the proposed X-GPN is superior to the classic and state-of-the-art (SOTA) methods in terms of the classification performance.**

***Index Terms*— Cross-scale, graph convolutional network (GCN), hyperspectral image (HSI) classification, semisupervised learning (SSL).**

Manuscript received January 31, 2021; revised July 23, 2021 and January 6, 2022; accepted March 2, 2022. This work was supported in part by the National Nature Science Foundation of China under Grant 61901343; in part by the Science and Technology on Space Intelligent Control Laboratory under Grant ZDSYS-2019-03; in part by the China Postdoctoral Science Special Foundation under Grant 2018T111019; in part by the Fundamental Research Funds for the Central Universities under Grant JB190107; in part by the National Nature Science Foundation of China under Grant 61571345, Grant 61671383, Grant 91538101, Grant 61501346, and Grant 61502367; in part by the 111 Project under Grant B08038; and in part by the Innovation Fund of Xidian University under Grant 5001-20109215456. *(Corresponding authors: Jiaojiao Li; Yunsong Li.)*

Bobo Xi and Jiaojiao Li are with the State Key Laboratory of Integrated Service Networks, School of Telecommunications Engineering, Xidian University, Xi'an 710071, China, and also with the CAS Key Laboratory of Spectral Imaging Technology, Xi'an 710119, China (e-mail: xibobo1301@foxmail.com; jjli@xidian.edu.cn).

Yunsong Li and Rui Song are with the State Key Laboratory of Integrated Service Networks, School of Telecommunications Engineering, Xidian University, Xi'an 710071, China (e-mail: ysli@mail.xidian.edu.cn; ruiscientific@gmail.com).

Yuchao Xiao is with the School of Artificial Intelligence, Beijing University of Posts and Telecommunications, Beijing 100876, China (e-mail: ycxiao@bupt.edu.cn).

Qian Du is with the Department of Electronic and Computer Engineering, Mississippi State University, Starkville, MS 39762 USA (e-mail: du@ece.msstate.edu).

Jocelyn Chanussot is with the University Grenoble Alpes, Inria, CNRS, Grenoble INP, LJK, 38000 Grenoble, France (e-mail: jocelyn.chanussot@grenoble-inp.fr).

Color versions of one or more figures in this article are available at https://doi.org/10.1109/TNNLS.2022.3158280.

Digital Object Identifier 10.1109/TNNLS.2022.3158280

I. Introduction

DIFFERENT from the red-green-blue (RGB) and multispectral image, hyperspectral image (HSI) provides more refined spectral information, which makes the land covers more distinguishable [1]. By virtue of its luxuriant spectral–spatial information, HSI has been applied to many significant applications, such as urban-area monitoring, precision agriculture, and mineral resource exploitation [2], to name a few. Similar to the semantic segmentation task for the RGB image, HSI classification (HSIC) intends to assign a predefined label to each individual pixel [3]. Obviously, HSIC is critical for a profound understanding of the captured scene, which attracts tremendous attention in practice. However, due to the intrinsic properties of HSI named high dimensionality versus insufficient annotated samples, high-accuracy HSIC is still a challenging subject and deserves further investigation [4]–[6].

Recently, semisupervised learning (SSL) has aroused broad concern, which can simultaneously harness the labeled and unlabeled data, fusing the advantages of the supervised and unsupervised algorithms to facilitate more accurate HSIC [7]–[9]. In specific, the SSL methods can be roughly divided into two categories: inductive learning and transductive learning. The former employs observed data out of the current task to build a prediction model. While in the latter, the adopted unlabeled samples are right from the test data, which can generally earn more promising performance. Most significantly, the graph-based learning imposes all concerned samples and their relationships to establish the topological structure [10], [11], and then propagates the labels and the node information until reaching a stable status. It has been validated by practical applications not only for the non-Euclidean domains but also for the regular statistics, such as the image dataset [12].

Noticeably, there have been several SSL studies for HSIC probing into the graph convolutional network (GCN) [13]–[18], which are differentiable and parameterized

by using Chebyshev polynomials to approximate the smooth filters in the spectral domain. In particular, Wan *et al.* [19] presented a multiscale dynamic GCN (MDGCN) leveraging the superpixel segmentation strategy, which achieved state-of-the-art (SOTA) recognition accuracy with high computational efficiency. However, the performance depends on the superpixel over-segmentation techniques to some degree. Besides, the features generated from multiple streams are finally aggregated without distinction. This is not reasonable in view that various scales for different samples should have different importance. Moreover, in the last layer of the networks, the softmax function is commonly employed to parse the extracted features and produce the probability vector reflecting the semantic information. The manner has shown a shortage of weakening the intraclass compactness and inducing less discriminative representations, which degrades the HSIC performance and deserves further studies [20]–[22].

To alleviate the above-mentioned issues, we propose a cross-scale graph prototypical network (X-GPN) for semi-supervised HSIC in an end-to-end transductive manner. In X-GPN, considering the multiscale characteristic of the objects in the remotely captured scene, the framework is designed in a multibranch fashion and multiscale neighborhoods are first selected to construct the adjacency matrices to directly involve the abundant spectral and precise spatial-context information. Second, for the sake of exploiting the complementary spatial information among diverse scales, we perform the lightweight 1-D convolution (1-D-Conv) on the middle-level graph feature maps to explore the intranode correlations and then concatenate them together to obtain more conducive expressions. Third, to adaptively assemble the high-level features produced by the multiple branches, we propose a self-branch attentional addition (SBAA) module to automatically assign varying importance on different samples of various scales. Hence, the most significant features can be accentuated, while the useless ones can be constrained. Finally, different from the widely used softmax paradigm, we utilize the distance-based cross-entropy (DCE) loss function and propose a novel temporal entropy-based regularizer (TER), which can synergistically derive more discriminative representations for achieving high-accuracy HSIC.

The main contributions of this study are specified as follows.

1) A novel X-GPN framework is proposed for high-accuracy HSIC, which can alleviate the overfitting dilemma with insufficient training samples in a semi-supervised mode.
2) To thoroughly investigate the abundant spectral–spatial information in the HSI, adjacency matrices determined by multiscale neighborhoods are involved in the multibranch graph convolutions. Moreover, to obtain more informative representations, the standard 1-D-Conv operation is applied to the graph nodes, which are concatenated afterward to exploit the complementary middle-level features across different scales.
3) An SBAA module is designed to merge the high-level features generated from different branches, which can adaptively assign different importance to various scales for each pixel. As a result, the expression ability of the extracted features is strengthened for accurately inferring the categories.
4) We devise an innovative prototypical layer comprising DCE loss function with a novel TER, which can simultaneously enhance the discrimination of the node features and the representativeness of the prototypes. Abundant experiments validate that the proposed X-GPN can achieve promising classification performance compared with the classic and SOTA methods.

The remains of this article are organized as follows. Section II introduces the related works, including a brief review of the HSIC, the formulations of GCN, and the convolutional prototype learning. Section III details the architecture of our proposed X-GPN. Section IV presents the experiment results and analysis. Section V draws the conclusions.

II. Related Works

A. Hyperspectral Image Classification

In the last two decades, plentiful canonical supervised classification models were successfully ameliorated for HSIC, such as random forest [23], K-nearest neighbors [24], extreme learning machine [25], [26], and so on. However, the HSIC performance dramatically degrades when faced with a lack of labeled samples, which is called the Hughes phenomenon. Then, support vector machine (SVM) [27] was presented to mitigate this problem. By mapping the original statistics into high-dimensional space, the separability of the features is virtually enhanced by kernel techniques [28]. Nevertheless, the approaches earlier are criticized for merely exploiting the shallow features that lack representativeness, which court suboptimal classification results.

To conquer this drawback, deep learning (DL)-based methods [29] have been extensively explored for HSIC nowadays [30], [31]. Specifically, the DL frameworks commonly learn from concrete to abstract hierarchical representations, and the parameters are iteratively optimized leveraging predefined penalty functions. The typical supervised networks for HSIC include deep belief networks [32], recurrent neural networks [33], and convolutional neural networks (CNNs) [34], and so on. Remarkably, 1-D-CNN [35], 2-D-CNN [36], and 3-D-CNN [37] were successively deployed to explore the spectral, spatial, and unified spectral–spatial features, respectively. In particular, Li *et al.* [38] proposed an excellent deep CNN using the pixel pair features (PPF-CNN), which achieved data augmentation through coupling the handful labeled samples. Then, a diverse region CNN (DR-CNN) [39] was presented to excavate the abundant spectral–spatial information, which acquired improved performance.

However, sufficient labeled samples are required for well training the above-mentioned CNN-related methods due to the considerable parameters. Unfortunately, the acquirement of annotated samples for HSI is time-consuming and labor-intensive in practical applications. Meanwhile, the above-mentioned supervised CNN models that are trained end-to-end cannot make full use of the vast number of unlabeled samples to improve the distinction among different classes. In contrast, by combining neighborhood structures with node features in the graph convolutional operations, the GCN can effectively aggregate the feature information across the neighbors of each graph node to obtain satisfactory results. Notably, the graph nodes include both the labeled and unlabeled instances. When faced with the small sample size regime, the GCN as a semisupervised approach can simultaneously exploit the information of the limited training data and a large number of unlabeled samples, which can facilitate extracting more expressive representations for achieving superior performance.

B. Graph Convolutional Network

In specific, the GCN is one of the most typical spectral-based convolutional graph neural networks (ConvGNNs), which essentially distills the spatial characteristics of the correlative vertices and edges in the topological graph [40], [41]. Remarkably, Kipf and Welling [13] employed Chebyshev polynomial to fit the convolution kernels and derived an efficient layerwise propagation rule, which can encode both local graph structure and node features to reach a more stable point. Specifically, in the perspective of the Fourier domain for the graph Laplacian [42], the convolution operation can be defined as

$$g_\theta \star \mathbf{x}_g = \mathbf{U} g_\theta \mathbf{U}^\top \mathbf{x}_g \tag{1}$$

where $\mathbf{x}_g$ refers to the graph signal and g_θ represents a convolutional filter parameterized by θ. $\mathbf{U}$ denotes the eigenvector matrix of the normalized graph Laplacian expressed as $\mathbf{L} = \mathbf{I} - \mathbf{D}^{-(1/2)}\mathbf{A}\mathbf{D}^{-(1/2)} = \mathbf{U}\mathbf{\Lambda}\mathbf{U}^T$, where $\mathbf{I}$ represents the identity matrix with a suitable size. $\mathbf{D}$ is the degree matrix of the graph and $\mathbf{A}$ is the adjacency matrix. $\mathbf{\Lambda}$ is a diagonal matrix filled with the eigenvalues of $\mathbf{L}$. Then, [43] involved truncated shifted Chebyshev polynomial $\mathrm{T}_k(x)$ up to the Kth order to approximate g_θ, which is expressed as

$$\mathrm{g}_\theta \star \mathbf{x}_g \approx \sum_{k=0}^{K} \theta_k \mathrm{T}_k(\widehat{\mathbf{L}})\mathbf{x}_g \tag{2}$$

where θ_k is kth Chebyshev coefficient and shifted $\widehat{\mathbf{L}} = (2/\lambda_{\max})\mathbf{L} - \mathbf{I}$. $\lambda_{\max}$ is the maximum eigenvalue of $\mathbf{L}$. It is notable that this operation is K-localized since it uses the Kth order polynomial of the Laplacian. GCN [13] further approximates $\lambda_{\max} \approx 2$ and limits the layerwise convolution operation to $K = 1$, then the computation can be represented as

$$\mathrm{g}_{\theta'} \star \mathbf{x}_g \approx \theta'_0 \mathbf{x}_g + \theta'_1(\mathbf{L} - \mathbf{I})\mathbf{x}_g = \theta'_0 \mathbf{x}_g - \theta'_1 \mathbf{D}^{-\frac{1}{2}}\mathbf{A}\mathbf{D}^{-\frac{1}{2}}\mathbf{x}_g \tag{3}$$

in which θ'_0 and θ'_1 are two free parameters shared throughout the whole graph. After constraining that $\theta = \theta'_0 = -\theta'_1$, (3) can be simplified as

$$\mathrm{g}_\theta \star \mathbf{x}_g \approx \theta\left(\mathbf{I} + \mathbf{D}^{-\frac{1}{2}}\mathbf{A}\mathbf{D}^{-\frac{1}{2}}\right)\mathbf{x}_g. \tag{4}$$

In order to circumvent the numerical instabilities and exploding/vanishing gradients, renormalization is conducted by $\mathbf{I} + \mathbf{D}^{-(1/2)}\mathbf{A}\mathbf{D}^{-(1/2)} \rightarrow \widehat{\mathbf{D}}^{-(1/2)}\widehat{\mathbf{A}}\widehat{\mathbf{D}}^{-(1/2)}$ with $\hat{\mathbf{A}} = \mathbf{A} + \mathbf{I}$ and $\widehat{\mathbf{D}}_{ii} = \sum_j \hat{\mathbf{A}}_{ij}$. Finally, for the signal $\mathbf{X} \in \mathbb{R}^{N\times C}$ (N nodes), the graph convolution (G-Conv) can be denoted as

$$\mathbf{Y} = \widehat{\mathbf{D}}^{-\frac{1}{2}}\widehat{\mathbf{A}}\widehat{\mathbf{D}}^{-\frac{1}{2}}\mathbf{X}\mathbf{\Theta} \tag{5}$$

where $\mathbf{\Theta} \in \mathbb{R}^{C\times F}$ represents the trainable convolutional parameters, and F equals the kernel number. $\mathbf{Y} \in \mathbb{R}^{N\times F}$ refers to the output of the G-Conv.

As a matter of fact, the geometry of the hyperspectral data in the feature space is highly nonlinear, but structured (the data indeed lives on a low-dimensional manifold). The feature space is severely impacted by varying illumination conditions and other factors, while working on the graph may bring an enhanced robustness [44]. Actually, taking the characteristics of HSI into consideration, increasing efforts have been devoted to GCN for HSIC, which demonstrate impressive classification performance. For example, Qin *et al.* [14] proposed a spectral–spatial GCN (S^2GCN) to make use of the spatial information, which achieved significant improvement on the vanilla GCN. In this study, we make further exploration of the cross-scale complementary spatial information and the dependence of the hidden intranode by taking both advantages of the G-Conv and the standard convolutions. In specific, the G-Conv is utilized to propagate the information of the labeled samples into the unlabeled data, while the standard 1-D-Conv is adopted to investigate local correlations of the intranode feature. By means of the interaction of multiple scales and feature concatenation operation, more representative features can be obtained for achieving a high-accuracy final decision.

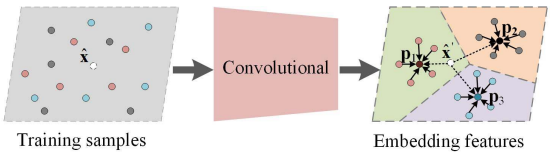


Fig. 1. Scheme of convolutional prototype learning.

C. Convolutional Prototype Learning

Typically, the DL-based multiclass classification frameworks employ the softmax function in the last layer to parse the comprehensive information derived from the initial sample and then output its probabilities belonging to each predefined class. However, it has been proven that the extracted representations in this manner are interclass separable but not intraclass compact. Consequently, the intraclass distance is larger than that of the alike samples between different categories and finally downgrades the classification result [20], [21]. To alleviate this problem, convolutional prototype learning (CPL) [22] is proposed and the scheme is depicted in Fig. 1.

Specifically, suppose that we have three categories drawn in different colors, and five training samples for each class. The prototypes $\mathbf{P} = \{\mathbf{p}_1, \mathbf{p}_2, \mathbf{p}_3\}$ representing each category and the embedding feature $f(\theta, \mathbf{x}_i)$ of the ith sample are simultaneous derived from the convolutional operations parameterized by θ. The utilized DCE objective function can be depicted as

$$L_{\mathrm{DCE}}(\theta, \mathbf{P}) = -\frac{1}{M \times C}\sum_{i=1}^{M}\sum_{j=1}^{C} \mathbf{1}\{j = y_i\} \log p\left(\mathbf{x}_i \in \mathbf{p}_j | \mathbf{x}_i\right) \tag{6}$$

where y_i is the true label of $\mathbf{x}_i$ and $\mathbf{1}\{j = y_i\}$ is an indicator function. M and C are the numbers of training samples and classes equaling 15 and 3 here, respectively. $p(\mathbf{x}_i \in \mathbf{p}_j|\mathbf{x}_i)$ means the probability that $\mathbf{x}_i$ belonging to the jth class, which is calculated by

$$p\left(\mathbf{x}_i \in \mathbf{p}_j | \mathbf{x}_i\right) = \frac{\mathrm{e}^{-\alpha d\left(f(\theta, \mathbf{x}_i), \mathbf{p}_j\right)}}{\sum_{k=1}^{C} \mathrm{e}^{-\alpha d\left(f(\theta, \mathbf{x}_i), \mathbf{p}_k\right)}} \tag{7}$$

where α is a steepness parameter and $d(f(\theta, \mathbf{x}_i), \mathbf{p}_j)$ denotes the distance between $f(\theta, \mathbf{x}_i)$ and the jth prototype.

In test, given a query sample $\hat{\mathbf{x}}$, the corresponding underlying feature is denoted as $\hat{\mathbf{x}}'$, then its category will be determined by computing the distance to the learned prototypes, which can be represented as

$$\hat{\mathrm{y}} = \arg\min_i \ \left(d(f(\theta, \hat{\mathbf{x}}), \mathbf{p}_i)\right) \tag{8}$$

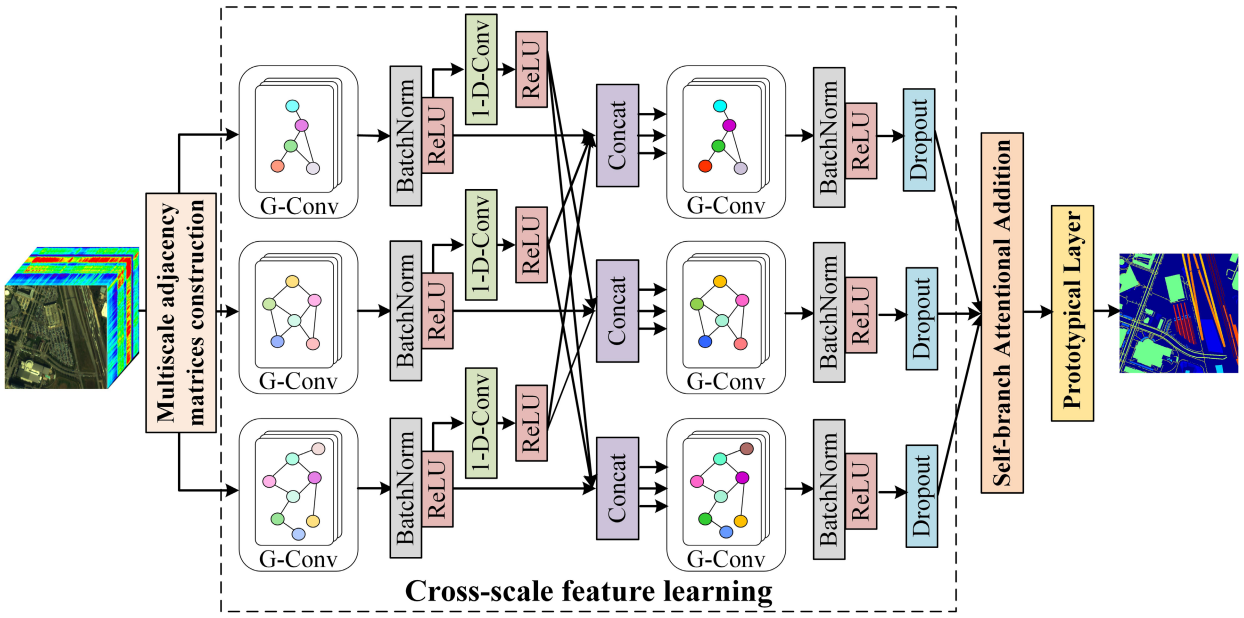


Fig. 2. Structure diagram of the proposed X-GPN for HSIC. It comprises four components: multiscale adjacency matrices construction, cross-scale feature learning, SBAA, and a novel prototypical layer.

where $\hat{y}$ is the predicted label to the test sample $\hat{\mathbf{x}}$. As in Fig. 1, it is clear to interpret that $\hat{\mathbf{x}}$ belongs to the first class since it is closest to $\mathbf{p}_1$. In this work, we propose a novel prototypical layer to integrate the CPL into GCN for enhancing the discrimination of the feature representations.

III. Proposed Method

A. Architecture of the Proposed X-GPN

The structure diagram of the proposed X-GPN is depicted in Fig. 2, which consists of four components, including multiscale adjacency matrices construction at the starting point, subsequent cross-scale feature learning, SBAA module, and the prototypical layer. First, taking multiscale contextual information into consideration, the multihop pixels of the central pixel are connected to construct the multiscale adjacency matrices, where the remaining positions are filled with zeros. Next, the 1-D-Convs followed by the G-Convs are performed on the graph signal with the assistance of the adjacency matrices to extract the expressive cross-scale node features. In this process, batch normalization (BatchNorm) is employed to scale the node feature, which can prominently accelerate the convergence process. Then, an SBAA module is devised to automatically aggregate the distilled features from multiple branches. Finally, in the designed prototypical layer, the assembled embedding features and the prototypes representing each class are interdependently optimized, which determines the precise classification through simple distance measurement. In the following, the accomplishments of the frameworks will be elaborated on in detail.

B. Multiscale Adjacency Matrices Construction

Given an HSI dataset, assuming that the labeled samples (i.e., pixels) are $\mathbf{X}_{\text{labeled}} \in \mathbb{R}^{N_1 \times L} = \{\mathbf{x}_l^1; \mathbf{x}_l^2; \ldots; \mathbf{x}_l^{N_1}\}$, where N_1 is the number of labeled samples and L denotes the length of each pixel. The corresponding labels are $\mathbf{Y} = \{y^1, y^2, \ldots, y^{N_1}\}$, where each element is a scalar belonging to $\{1, 2, \ldots, C\}$ and C is the number of the classes. Meanwhile, the unlabeled samples to-be-classified are denoted as $\mathbf{X}_{\text{unlabeled}} \in \mathbb{R}^{N_2 \times L} = \{\mathbf{x}_u^1; \mathbf{x}_u^2; \ldots; \mathbf{x}_u^{N_2}\}$, where N_2 is the number of the unlabeled samples, and $N_1 + N_2 = N$.

Taking all samples into account, the multiscale graph can be represented as $G^s = (\mathbf{X}, \mathbf{A}^s)$, where $\mathbf{X} \in \mathbb{R}^{N \times L}$ is the node feature of the graph. $\mathbf{A}^s$ denotes the sth adjacency matrix, and s is in the scope of {1, 2, 3} since three scales are involved into the calculation in this article. For the non-Euclidean dataset, e.g., citation networks, K nearest neighbors according to the node distance are selected to establish the connections in $\mathbf{A}$, while in the regular grid HSI dataset, we choose to utilize the physical neighborhoods to build the edges of the graphs. In this manner, the significant spatial correlations of the land covers can be preserved, which is prominent in the HSIC tasks. Precisely, $\mathbf{A}^s$ can be calculated by the radial basis function (RBF)

$$\mathbf{A}_{i,j}^s = \begin{cases} e^{-\tau \|\mathbf{x}^i - \mathbf{x}^j\|^2}, & \text{if } \mathbf{x}^i \in R_{\mathbf{x}^j}^s \text{ or } \mathbf{x}^j \in R_{\mathbf{x}^i}^s \\ 0, & \text{else} \end{cases} \tag{9}$$

where τ is a hyperparameter influencing the weight of the edges. $R_{\mathbf{x}^i}^s$ means the region with the size of $(2s+1) \times (2s+1)$ centered around $\mathbf{x}^i$, analogously for $R_{\mathbf{x}^j}^s$.

From the construction process of the adjacency matrices, it can be found that compared with the standard convolutions that investigate the spectral–spatial information of locally fixed regions due to the local receptive field, the GCN can directly utilize the correlations between adjacent pixels to conduct flexible convolution on irregular image regions. Specifically, this advantage is reflected in two aspects: 1) the samples out of the training, validation, and test sets will not participate in the subsequent calculation, which can avoid the interference from the background pixels, especially for boundary regions and 2) the edge weights between the nodes are calculated by directly using the spectral distance of the training, validation,

and test pixels in the scope of $(2s+1)\times(2s+1)$. That indicates, if the central pixel and its surrounding pixels belong to different classes, their connection weights will be relatively small. Then, the information flow between these nodes is naturally decreased, which explicitly facilitates the networks to divide the samples into different categories. Hence, the spatial-context structure of the HSI can be better modeled by using the G-Convs [18], [41].

C. Cross-Scale Feature Learning

In virtue of the multiscale adjacency matrices, the feature matrix is deeply investigated by the G-Conv and 1-D-Convs, exploring the effective representations of each node. As [14], we set K to 2 in the G-Convs, which are conducted on the graph signal involving the neighbor nodes of the current node. Then, (3) should be modified as:

$$\begin{aligned} \mathrm{g}_{\theta'} \star \mathbf{x}_g &\approx \theta'_0\mathbf{x}_g + \theta'_1(\mathbf{L}-\mathbf{I})\mathbf{x}_g + \theta'_2\big(2(\mathbf{L}-\mathbf{I})^2-1\big)\mathbf{x}_g \\ &= \big(\theta'_0-\theta'_2\big)\mathbf{x}_g - \theta'_1\mathbf{D}^{-\frac{1}{2}}\mathbf{A}\mathbf{D}^{-\frac{1}{2}}\mathbf{x}_g + 2\theta'_2\Big(\mathbf{D}^{-\frac{1}{2}}\mathbf{A}\mathbf{D}^{-\frac{1}{2}}\Big)^2 \\ &\quad \times \mathbf{x}_g. \end{aligned} \tag{10}$$

Noticeably, different from one arbitrary node feature $\mathbf{x}$ (one-pixel vector in the first G-Conv), the graph signal $\mathbf{x}_g$ is the feature in the same dimension overall the graph. As (4), restricting $\theta = 2\theta'_2 = -\theta'_1 = \theta'_0 - \theta'_2$, (10) is simplified to

$$\mathrm{g}_\theta \star \mathbf{x}_g \approx \theta\left(\mathbf{I} + \mathbf{D}^{-\frac{1}{2}}\mathbf{A}\mathbf{D}^{-\frac{1}{2}} + \left(\mathbf{D}^{-\frac{1}{2}}\mathbf{A}\mathbf{D}^{-\frac{1}{2}}\right)^2\right)\mathbf{x}_g \tag{11}$$

Then, the output feature maps of the sth scale can be obtained by operating the first G-Conv on the graph signal $\mathbf{X}\in\mathbb{R}^{N\times L}$

$$\mathbf{H}_1^s = \left(\mathbf{I} + \mathbf{D}^{-\frac{1}{2}}\mathbf{A}\mathbf{D}^{-\frac{1}{2}} + \left(\mathbf{D}^{-\frac{1}{2}}\mathbf{A}\mathbf{D}^{-\frac{1}{2}}\right)^2\right)\mathbf{X}\Theta_1^s \tag{12}$$

where $\mathbf{\Theta}_1^s \in \mathbb{R}^{L\times F_1}$ refers to the learnable filter parameters in the sth scale of the first G-Conv, and $\mathbf{H}_1^s \in \mathbb{R}^{N\times F_1}$.

Inspired by the effective BatchNorm technique in CNNs [45], [46], we perform normalization over the nodes of the graph to reduce the internal covariance shift, in order to boost the convergence and accelerate the training process. Concretely, denote one graph signal after the first G-Conv as $\mathbf{h}_{1g}^s \in \mathbb{R}^{N\times 1}$, the BatchNorm can be mathematically described as

$$\begin{aligned} \hat{\mathbf{h}}_{1g}^s &= \frac{\mathbf{h}_{1g}^s - \mu_{1g}^s}{\sigma_{1g}^s} \\ \mu_{1g}^s &= \frac{1}{N}\sum_{i=1}^{N}\mathbf{h}_{1g}^{s(i)} \\ \sigma_{1g}^s &= \sqrt{\frac{1}{N}\sum_{i=1}^{N}\left(\mathbf{h}_{1g}^{s(i)} - \mu_{1g}^s\right)^2} \end{aligned} \tag{13}$$

where $\mathbf{h}_{1g}^{s(i)}$ represents the graph signal element of the ith node. Next, ReLU nonlinear activation function [47] is performed to obtain the hidden feature $\hat{\mathbf{H}}_1^s \in \mathbb{R}^{N\times F_1}$.

In fact, the G-Conv mainly focuses on exploring the internode correlations. To explicitly exploit the intranode dependence between the node feature maps, we perform the standard 1-D-Conv on $\hat{\mathbf{H}}_1^s$, which can be expressed by

$$\mathbf{H}_2^s = \Phi\left(\mathbf{W}^s\hat{\mathbf{H}}_1^s + b^s\right) \tag{14}$$

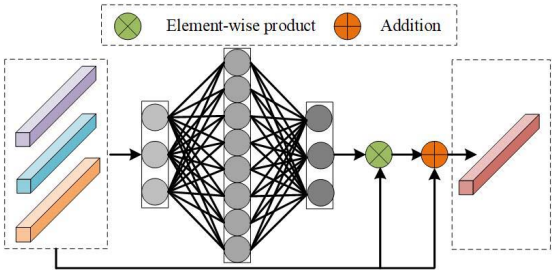


Fig. 3. Implementation of the SBAA.

where $\mathbf{H}_2^s \in \mathbb{R}^{N\times F_1}$ denotes the output of the ReLU function $\Phi(\cdot)$ following the 1-D-Conv in the sth scale. $\mathbf{W}^s$ and b^s represent the weight and bias that independently impose on each node feature of the $\hat{\mathbf{H}}_1^s$. To achieve information exchange between different scales and take advantage of the features from both the upper and lower levels, we concatenate $\hat{\mathbf{H}}_1^s$ and $\mathbf{H}_2^k$ across scales to acquire $\mathbf{H}_3^s$, which can be depicted as

$$\mathbf{H}_3^s = \hat{\mathbf{H}}_1^s \big\| \mathbf{H}_2^k, \quad k = \{1,2,3\} \text{ and } k \neq s \tag{15}$$

where $||$ denotes the concatenation operation. Subsequently, $\mathbf{H}_3^s$ is fed into the next G-Conv layer to obtain higher level abstract representations $\mathbf{H}_4^s \in \mathbb{R}^{N\times F_2}$

$$\mathbf{H}_4^s = \left(\mathbf{I} + \mathbf{D}^{-\frac{1}{2}}\mathbf{A}\mathbf{D}^{-\frac{1}{2}} + \left(\mathbf{D}^{-\frac{1}{2}}\mathbf{A}\mathbf{D}^{-\frac{1}{2}}\right)^2\right)\mathbf{H}_3^s\mathbf{\Theta}_2^s \tag{16}$$

where $\mathbf{\Theta}_2^s \in \mathbb{R}^{F_1\times F_2}$ represents the trainable parameters of the second G-Conv. Similarly, the BatchNorm and ReLU function are successively imposed on $\mathbf{H}_4^s$ to generate the refined hidden feature $\hat{\mathbf{H}}_4^s$. Next, dropout [48] regularization is employed to alleviate the overfitting problems and promote the generalization capability of the model. As a result, the output of the dropout layer from each channel can be acquired as $\mathbf{H}_5^s \in \mathbb{R}^{N\times F_2}$.

D. Self-Branch Attentional Addition

In a given scene, the appearance of the same material has various scales, which have different difficulty levels for identifying its category. Intuitively, the distilled features from each scale of each sample should have different importance for reasoning the final category, too. To this end, we devise an SBAA module to automatically integrate the deep node features $\mathbf{H}_5^s$ from multiple branches. Visually, the implementation is delineated in Fig. 3.

Suppose $\mathbf{h}_5 \in \mathbb{R}^{F_2\times 3}$ attributes one node feature of three scales. We first squeeze it into a node descriptor $\mathbf{D}\in\mathbb{R}^{1\times 3}$ along the node feature dimension through using 1-D global average pooling. Then, the parameters are learned to explicitly model the correlation between the features from different branches. Specifically, $\mathbf{D}$ is fed into two consecutive fully connected (FC) layers to obtain the attention weight vector $\mathbf{\Omega}\in\mathbf{R}^{1\times 3}$, which can be expressed as

$$\mathbf{\Omega} = \Psi(\mathbf{W}_\mathrm{D}\Phi(\mathbf{W}_\mathrm{U}\mathbf{Z})) \tag{17}$$

where Φ is the ReLU activation function. $\mathbf{W}_\mathrm{U}$ is the weight of the first FC layer, which acts as upscaling with r hidden nodes. $\mathbf{W}_\mathrm{D}$ is the weight of the second FC layer. Compared with using only one FC layer, the two FCs structure has more nonlinearity

for better fitting the complex correlation between channels. The sigmoid function $\Psi(\cdot)$ is used as a gating mechanism to control the value of attention weight ranging from 0 to 1. To avoid unexpected information loss, the output node feature from each scale is reused through residual connections. Then, we have the output of the SBAA represented by

$$\mathbf{h}_6 = \sum_{s=1}^{3} \mathbf{h}_5^s \otimes (1 + \mathbf{\Omega}^s) \tag{18}$$

where $\mathbf{h}_5^s$ represents one node feature of the sth scale, and $\mathbf{\Omega}^s$ refers to the sth attention score of $\mathbf{\Omega}$. $\mathbf{h}_6$ is one of the output node feature denoted as $\mathbf{H}_6$ of the integrated graph. Notably, from the above-mentioned cross-scale feature learning and SBAA module, it is observed that the multiscale spectral–spatial features can not only interact in the middle level but also flexibly aggregate in the higher level, which can boost the representativeness of the extracted in-depth features.

E. Prototypical Layer

At the end of the networks, the attentional merged node feature $\mathbf{H}_6$ is fed into a prototypical layer to anatomize their attribution information. As introduced in Section II-C, in the training process, the optimal prototypes in the prototypical layer are cooperatively learned with the node features, while in the test procedure, categories of the unlabeled samples can be determined by (8). To further enhance the discrimination of the extracted features, we devise an ER involving both the labeled and unlabeled samples, which can be depicted as

$$L_{\mathrm{ER}}(\mathbf{Y}) = -\frac{1}{N \times C} \sum_{i=1}^{N} \sum_{j=1}^{C} \hat{y}_j \log \hat{y}_j$$

$$\hat{y}_j = p\left(\mathbf{x}_i \in \mathbf{p}_j \mid \mathbf{x}_i\right) = \frac{e^{-\alpha d\left(\mathbf{h}_6, \mathbf{p}_j\right)}}{\sum_{k=1}^{C} e^{-\alpha d\left(\mathbf{h}_6, \mathbf{p}_k\right)}} \tag{19}$$

where $\hat{y}_j$ denotes the predict probability that the ith sample $\mathbf{x}_i$ belongs to the jth class, i.e., the normalized distance between the corresponding node feature $\mathbf{h}_6$ of $\mathbf{x}_i$ and the jth prototype $\mathbf{p}_j$. Followed [22], we empirically assign the Euclidean distance as the measurement in L_{DCE} and L_{ER}. It is noteworthy that, benefited from the ER, the produced prototypes are not only determined by the labeled samples but also affected by the unlabeled data. Namely, the prototypes are optimized using both the training and test samples during the training procedure, and their representativeness can be further enhanced for more precise classification under this semisupervised generative strategy.

In addition, in the early training phase of the framework, the prediction probability is suspect since the network has not been well convergent. If we directly apply $L_{\mathrm{ER}}(\mathbf{Y})$ from the beginning, the classifier may be led to an unexpected bias. Therefore, we employ a sigmoid function by using the current epoch number to restrict $L_{\mathrm{ER}}(\mathbf{Y})$. The temporal coefficient can be calculated as

$$T = \frac{1}{1 + e^{-\beta t}}, \quad t = \frac{\text{epoch\_ct} - \frac{\text{Epoch\_num}}{2}}{\text{Epoch\_num}} \tag{20}$$

where β is a parameter affecting the warm-up period of the regularizer. epoch_ct and epoch_num denote the current and the total epoch number during the optimization process, respectively. Comprehensively, the objective function employed in the prototypical layer comprises the DCE loss and the TER, which can be summarized as

$$L = L_{\mathrm{DCE}} + L_{\mathrm{TER}} = L_{\mathrm{DCE}} + \mathrm{TL}_{\mathrm{ER}} \tag{21}$$

In particular, the effectiveness of the proposed objective function will be analyzed in Section IV-H.

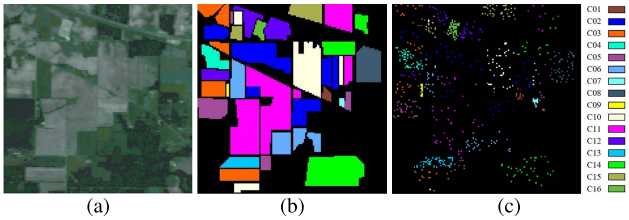


Fig. 4. InP dataset. (a) False-color synthetic image (30-R, 25-G, and 18-B bands). (b) Ground truth. (c) Standard training samples. Best in zoomed-in view.

IV. Experiments and Analysis

In order to evaluate the validity of our proposed networks, abundant experiments are carried out on three real-world HSIC benchmarks, i.e., Indian Pines (InP), University of Pavia (UP), and Kennedy Space Center (KSC) scenes. The computing power is provided by an NVIDIA GeForce GTX 1080Ti GPU. The architecture of the networks is constructed by the familiar TensorFlow DL frameworks. In addition, three widely used criteria as average accuracy (AA), overall accuracy (OA), and Kappa coefficient (k) are employed to quantitatively evaluate the performance of the proposed and compared algorithms.

A. Experiment Datasets

1) Indian Pines: This classic InP dataset was collected over Northwestern Indiana by the Airborne Visible/Infrared Imaging Spectrometer (AVIRIS) facility. The spatial dimension comprises 145 × 145 pixels and the geographic resolution is 20 m. In the spectral dimension, reflection values of 220 bands were recorded from 0.4 to 2.5 μm. We preserve 200 bands in our experiments after abandoning the noisy and water-absorbed channels. The false-color synthetic image and the ground truth are shown in Fig. 4(a) and (b), respectively. We can observe that there are 16 labeled categories distributed in the scenario. Since this image contains much noise and most land-cover types are vegetation with minor spectral differences, it is a challenging HSIC benchmark that is widely used to assess the fresh algorithms.

2) University of Pavia: The UP dataset was captured by the Reflective Optics System Imaging Spectrometer (ROSIS) sensor over an urban field, which consists of 610 × 340 pixels in 1.3-m spatial resolution. This image has 103 channels in the wavelength ranging from 0.43 to 0.86 μm after removing the noisy bands. Nine labeled land-cover classes exist in the scene. The false-color composite image and ground truth are displayed in Fig. 5(a) and (b), respectively.

3) Kennedy Space Center: The KSC dataset was acquired over the Kennedy Space Center, FL, USA, by the AVIRIS sensor. This image is composed of 224 bands with spectral coverage from 0.4 to 2.5 μm. After discarding the low SNR and water absorption bands, 176 channels are preserved in the experiments. In the spatial dimension, there are 614 × 512 pixels with a resolution of 18 m. There are sixteen different annotated land-cover types, such as "Scrub," "Willow swamp,"

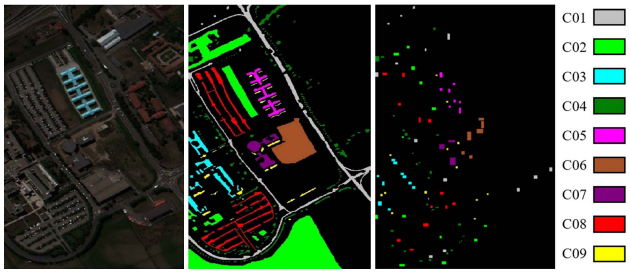


Fig. 5. UP dataset. (a) False-color synthetic image (30-R, 25-G, and 18-B bands). (b) Ground truth. (c) Standard training samples. Best in zoomed-in view.

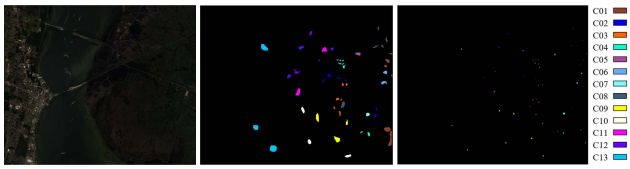


Fig. 6. KSC dataset. (a) False-color synthetic image (50-R, 30-G, and 20-B bands). (b) Ground truth. (c) Controlled random sampling training samples. Best in zoomed-in view.

and so on. Fig. 6(a) and (b) demonstrate its false-color image and ground truth map, respectively.

For the first two datasets, their standard training and test split scheme have been wildly utilized as benchmarks in the recent literature [4], [15], [49]. Thus, we adopt this partitioning mode in this study. The training locations are visualized in Figs. 4(c) and 5(c), respectively. For the KSC dataset, we employ the classic controlled random sampling strategy (CRSS) [50]–[53] to choose 30 spatially separated labeled samples from each class to constitute the training set, which is shown in Fig. 6(c). Unlike the widely used random sampling strategy, we can observe that the training set generated by CRSS is concentrated locally and dispersed globally, which is similar to the distribution of the standard training set of UP. This training/test assignment is closer to the practical application, which can ensure the realistic performance of the proposed spectral–spatial classifier [31], and, in turn, convincingly demonstrate the superior performance of our proposed frameworks. Moreover, among these training sets, 10% samples are randomly selected as the validation data to develop the networks. Namely, during the training process, 90% of the training sets are with available labels for optimizing the learnable weights, and the remainder is utilized to fine-tune the hyperparameters, such as the learning rate, epoch number, and so on. In specific, the numbers of the train, validation, and test samples of the experiment datasets are detailed in Tables I–III, respectively.

B. Experiment Setups

As shown in Fig. 2, we utilize three scales to construct the adjacency matrices, which are subsequently involved in the graph convolutional calculation to investigate the spectral–spatial information. Specifically, spatial neighborhoods as 3×3, 5×5, and 7×7 of the central target pixel are taken into account to fill the connections. Here, the parameter τ in (9) affects the edge strength in the adjacency matrices. We investigate the performance sensitivity of X-GPN with different values of τ on the three datasets. In particular, the grid search strategy is chosen to find the optimal value of τ, which varies in the collection of {0.001, 0.01, 0.1, 1, 10}. The overall classification accuracies are displayed in Fig. 7.

TABLE I

LABELED LAND-COVER TYPES AND NUMBERS OF TRAIN, VALIDATION, AND TEST SAMPLES FOR INP DATASET

ID	Land-Cover Type	Train	Vali.	Test
C01	Alfalfa	13	2	39
C02	Corn-notill	45	5	1384
C03	Corn-mintill	45	5	784
C04	Corn	45	5	184
C05	Grass-pasture	45	5	447
C06	Grass-trees	45	5	697
C07	Grass-pasture-mowed	13	2	11
C08	Hay-windrowed	45	5	439
C09	Oats	13	2	5
C10	Soybean-notill	45	5	918
C11	Soybean-mintill	45	5	2418
C12	Soybean-clean	45	5	564
C13	Wheat	45	5	162
C14	Woods	45	5	1244
C15	Buildings-Grass-Trees-Drives	45	5	330
C16	Stone-Steel-Towers	45	5	45
	Total	624	71	9671

TABLE II

LABELED LAND-COVER TYPES AND NUMBERS OF TRAIN, VALIDATION, AND TEST SAMPLES FOR UP DATASET

ID	Land-Cover Type	Train	Vali.	Test
C01	Asphalt	493	55	6304
C02	Meadows	486	54	18146
C03	Gravel	353	39	1815
C04	Trees	472	52	2912
C05	Painted metal sheets	238	27	1113
C06	Bare Soil	479	53	4572
C07	Bitumen	337	38	981
C08	Self-Blocking Bricks	463	51	3364
C09	Shadows	208	23	795
	Total	3529	392	40002

TABLE III

LABELED LAND-COVER TYPES AND NUMBERS OF TRAIN, VALIDATION, AND TEST SAMPLES FOR KSC DATASET

ID	Land-Cover Type	Train	Vali.	Test
C01	Scrub	27	3	731
C02	Willow swamp	27	3	213
C03	CP hammock	27	3	226
C04	CP/Oak	27	3	222
C05	Slash pine	27	3	131
C06	Oak/Broadleaf	27	3	199
C07	Hardwood swamp	27	3	75
C08	Graminoid marsh	27	3	401
C09	Spartina marsh	27	3	490
C10	Cattail marsh	27	3	374
C11	Salt marsh	27	3	389
C12	Mud flats	27	3	473
C13	Water	27	3	897
	Total	351	39	4821

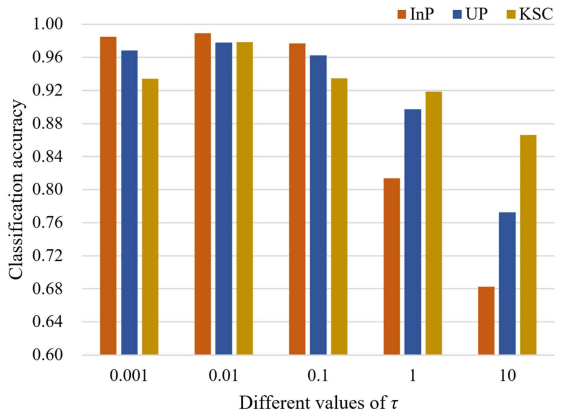


Fig. 7. Classification performance with different values of τ in the multiscale adjacency matrices construction.

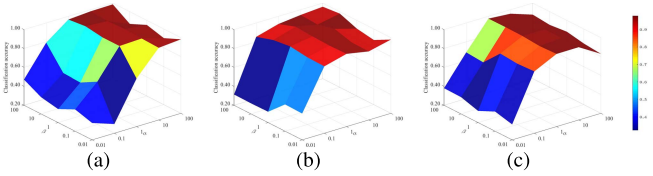


Fig. 8. Influence of the parameters of α and β in the proposed prototypical layer. (a) InP. (b) UP. (c) KSC.

As shown in Fig. 7, the OA reaches an optimal point for all datasets when $\tau = 0.01$. It implies that the connection of the graph nodes cannot be too tight or too loose for satisfactory results. Thus, we set τ to 0.01 in our experiments.

For the cross-scale feature learning procedure, since the GCN commonly has no need for very deep networks to obtain satisfied performance [19], all experiment datasets share the same network architecture comprising two G-Conv layers. The hidden units of the first and second G-Conv are experimentally assigned to 32 and 16, respectively. Besides, the standard 1-D-Convs are configured with the generally used 1×3 size kernels. In addition, in the SBAA module, the number of the hidden nodes in the first FC layer is empirically set to 10. To search the optimal network parameters, we employ the Adam optimization algorithm with the full-batch gradient descent method, where the learning rate and the epoch number are 0.01 and 1000, respectively. The relatively small epoch number is benefited from the use of the BatchNorm in our networks, which will be elaborated in Section IV-I. Moreover, in order to mitigate the overfitting phenomenon, early stopping, and dropout techniques are adopted, and the dropout rate is set to 0.2 for the InP and UP, and 0.3 for KSC datasets.

In the prototypical layer, the prototypes belonging to each class are initialized to be orthogonal to promote their discrimination. There are two critical parameters that need to be determined, i.e., the steepness parameters α and β in TER. We analyze their influence on them by using the three experiment datasets. Keeping other settings unchanged, the tradeoff parameters α and β both vary in the set of {0.01, 0.1, 1, 10, 100}. The overall classification accuracies are shown in Fig. 8. As can be observed, the OAs will collapse if the α and β are set inappropriately, which indicates their importance. And the performance achieves the best when $\alpha = 1$ and $\beta = 10$ for the three datasets. Therefore, we set them to be 1 and 10, respectively. For the sake of brevity, the above-mentioned parameter settings are summarized according to different components of the network, which are listed in Table IV.

TABLE IV
PARAMETER SETTINGS OF THE NETWORK

Components	Parameter	Value
Adjacency matrices	scales	3×3, 5×5, 7×7
	τ	0.01
G-Conv	hidden units	32, 16
1-D-Conv	kernel size	1×3
SSBA	hidden nodes	10
Dropout	dropout rate	0.2 (0.3 for KSC)
Optimization	learning rate	0.01
	epoch number	1000
Prototypical layer	α	1
	β	10

C. Network Ablations

In this section, we perform ablation studies to determine whether each component of the proposed frameworks is necessary and effective. Specifically, X-GPN w/o CS and w/o SBAA and w/o both of them are tested on the three experiment datasets. The first variant means that X-GPN removes the 1-D-Convs and cross-scale feature concatenate operations, while the third variant is denoted as GPN. Table V shows the influence of the ablations, where the results are mean accuracies and the standard deviations over ten independent runs. The highest values are highlighted in bold. It can be seen that X-GPN achieves the optimal results and the performance degrades if CS and SBAA are cut off. In specific, the OA of X-GPN surpasses X-GPN w/o CS, X-GPN w/o SBAA, and GPN by 0.31%, 0.53%, and 1.02% on the InP dataset, respectively; by 0.57%, 0.54%, and 0.72% on the UP dataset, respectively; by 1.16%, 1.58%, and 1.99% on the KSC dataset, respectively.

Besides, we employ McNemar's test to validate the significance of classification accuracies produced by the X-GPN and other variants. In detail, McNemar's test can be performed by

$$z_{ij} = \frac{f_{ij} - f_{ji}}{\sqrt{f_{ij} + f_{ji}}} \tag{22}$$

where f_{ij} is the number of the samples that correctly categorized by algorithm i but incorrectly categorized by algorithm j. |z| denotes the absolute value of z. For 5% level of significance, the |z| value is 1.96. That means, if the |z| value is greater than this quantity, the two classification maps have significant discrepancy. Concretely, Table V presents the average |z| values achieved from the InP, UP, and KSC datasets of the proposed X-GPN against other variants. A "True" here denotes the two classification methods in McNemar's test have significant performance discrepancy. Evidently, the proposed classification framework is statistically different from its counterparts with 5% significance level ($|z| > 1.96$).

As for the computational complexity, the three lightweight 1-D-Convs just have $4 \times 3 = 12$ parameters and the SBAA module with two FCs only has $3 \times 10 \times 2 = 60$ parameters, which are negligible compared with that of the principal G-Convs of the networks. In conclusion, it can be inferred

TABLE V
CLASSIFICATION RESULTS OF THE NETWORK ABLATIONS ON THE THREE EXPERIMENT DATASETS

DataSet	Method	CS	SBAA	AA (%)	OA (%)	$k\times100$	$\|z\|$ value/significant?
InP	X-GPN	✓	✓	**99.28±0.05**	**98.68±0.15**	**98.48±0.09**	-
	w/o CS		✓	99.11±0.09	98.37±0.08	98.12±0.14	3.94/True
	w/o SBAA	✓		99.22±0.05	98.15±0.16	97.88±0.13	3.39/True
	GPN			99.02±0.09	97.66±0.09	97.32±0.12	6.96/True
UP	X-GPN	✓	✓	**97.19±0.15**	**98.21±0.05**	**97.57±0.03**	-
	w/o CS		✓	96.71±0.17	97.64±0.13	96.81±0.10	11.38/True
	w/o SBAA	✓		96.42±0.23	97.67±0.11	96.84±0.14	9.82/True
	GPN			95.64±0.21	97.49±0.12	96.61±0.09	11.18/True
KSC	X-GPN	✓	✓	**96.80±0.05**	**97.78±0.16**	**97.52±0.07**	-
	w/o CS		✓	94.91±0.19	96.62±0.21	96.22±0.09	5.44/True
	w/o SBAA	✓		93.91±0.22	96.20±0.17	95.75±0.13	6.94/True
	GPN			93.55±0.08	95.79±0.28	95.29±0.17	7.84/True

TABLE VI
CLASSIFICATION ACCURACIES OBTAINED BY DIFFERENT ALGORITHMS FOR INP DATASET

NO.	Land-Cover Type	RBF-SVM	2-D-CNN	3-D-CNN	PPF-CNN	DR-CNN	GCN	S^2GCN	MDGCN	X-GPN
01	Alfalfa	79.49	92.31	**100.00**	97.44	**100.00**	**100.00**	**100.00**	84.90	**100.00**
02	Corn-notill	67.99	79.12	66.11	84.25	82.17	82.95	81.29	**98.09**	95.30
03	Corn-mintill	66.84	76.66	79.59	88.78	84.36	80.61	87.76	**100.00**	99.87
04	Corn	85.87	98.91	**100.00**	98.37	85.65	95.65	**100.00**	95.53	**100.00**
05	Grass-pasture	90.38	88.37	**97.99**	95.30	91.60	93.29	97.09	96.99	97.09
06	Grass-trees	86.66	80.49	90.24	93.69	**100.00**	92.25	91.68	99.09	**100.00**
07	Grass-pasture-mowed	90.91	**100.00**	**100.00**	90.91	91.67	92.59	**100.00**	**100.00**	**100.00**
08	Hay-windrowed	94.31	98.86	96.81	99.09	**100.00**	97.95	99.32	97.85	**100.00**
09	Oats	**100.00**	**100.00**	**100.00**	**100.00**	**100.00**	**100.00**	**100.00**	95.74	**100.00**
10	Soybean-notill	76.25	80.72	72.22	93.14	82.42	89.11	93.46	98.77	**99.89**
11	Soybean-mintill	60.22	68.07	75.43	74.73	95.70	79.20	85.03	**100.00**	98.51
12	Soybean-clean	82.98	81.56	79.08	84.22	80.32	86.35	84.22	**100.00**	97.87
13	Wheat	96.91	**100.00**	**100.00**	99.38	98.18	**100.00**	**100.00**	**100.00**	**100.00**
14	Woods	80.47	94.13	94.61	90.84	99.66	95.98	97.11	**100.00**	**100.00**
15	Buildings-Grass-Trees-Drives	73.03	97.58	94.24	92.12	77.62	92.73	93.94	**100.00**	**100.00**
16	Stone-Steel-Towers	97.78	**100.00**	**100.00**	97.78	78.95	**100.00**	**100.00**	**100.00**	**100.00**
AA (%)	-	83.13	89.80	90.40	92.50	90.52	92.88	94.43	97.47	**99.28**
OA (%)	-	74.02	81.35	81.67	86.71	90.01	87.15	89.70	95.71	**98.68**
$k\times100$	-	70.66	78.87	79.15	84.90	88.61	85.35	88.24	95.07	**98.48**
Params	-	-	913.56K	445.01K	60.95K	9.15M	16.24K	16.24K	13.07K	71.90K

that, while the proposed framework seems complex, each partition of it is productive and efficient for achieving better classification.

D. Classification Performance

To demonstrate the superiority of our proposed algorithm, we compare X-GPN with some prevalent and SOTA HSIC approaches, where the parameter configurations are kept the same with the corresponding publications. Specifically, the RBF-SVM [27] denotes the SVM with the RBF kernel, which acts as a milestone in traditional HSIC. The popular CNN-based methods, i.e., 2-D-CNN [36], 3-D-CNN [37], PPF-CNN [38], and DR-CNN [39], are highly cited as spectral–spatial HSIC baselines. Hence, we employed them to compare with X-GPN. In addition, GCN [13] introduces DL paradigm into graph learning. S^2GCN [14] further takes advantage of both spectral and spatial features in HSI. MDGCN [19] is one of the recently proposed SOTA methods, which also utilizes multiscale spatial information. Thus, the three graph-related SSL methods are also taken as comparisons. All algorithms are executed up to ten times in our experiments. In detail, the obtained mean per-class accuracy, AA, OA, k, and the number of the parameters of the DL-based networks are displayed from Tables VI–VIII. The highest accuracies of each row are highlighted in bold.

From Table VI, we can observe that the presented X-GPN obtains the best performance with respect to AA, OA, and k. As expected, the traditional RBF-SVM is inferior to the following DL-based approaches. Regarding AA, it is shown that the SSL methods based on graph convolution exceed the supervised CNN-based networks, which demonstrates its potential in HSIC to a certain degree. What is more, with a moderate number of parameters, most class-specific accuracies of X-GPN rank first place among these methods, and ten classes even reach 100% perfect prediction. Especially, for the tenth class named Soybean-notill, X-GPN achieves 99.89% accuracy, while it is difficult for other methods to recognize. It can be speculated that the cross-scale graphs have a strong capability to investigate the abundant spectral–spatial features contained in the HSI. Besides, the SBAA module and prototypical layer can produce more discriminative representations, which boosts the classification performance.

Table VII shows the classification results of various algorithms on the UP dataset. Due to the scattered gathered distribution of the training samples, the high-accuracy classification is challenging for most of the methods. Thus,

TABLE VII
CLASSIFICATION ACCURACIES OBTAINED BY DIFFERENT ALGORITHMS FOR UP DATASET

NO.	Land-Cover Type	RBF-SVM	2-D-CNN	3-D-CNN	PPF-CNN	DR-CNN	GCN	S^2GCN	MDGCN	X-GPN
01	Asphalt	84.23	85.73	82.61	94.15	**98.62**	84.38	82.22	90.74	95.67
02	Meadows	66.43	88.34	81.54	83.66	**99.87**	76.20	82.60	97.17	99.86
03	Gravel	68.43	62.85	57.19	52.95	84.51	83.58	79.06	**88.37**	86.28
04	Trees	97.87	84.44	**98.73**	94.88	96.40	94.40	97.70	96.36	97.63
05	Painted metal sheets	99.37	88.80	99.55	99.82	98.92	99.91	**100.00**	96.86	**100.00**
06	Bare Soil	92.43	86.03	92.98	97.48	67.99	94.73	95.32	86.20	**99.15**
07	Bitumen	89.91	61.10	81.86	93.58	91.16	95.11	**98.78**	82.57	97.15
08	Self-Blocking Bricks	92.60	78.31	97.24	**99.41**	71.44	95.01	94.59	96.46	98.99
09	Shadows	97.36	93.89	94.21	**100.00**	97.17	96.60	**100.00**	97.36	**100.00**
AA (%)	-	87.63	81.05	87.32	90.66	89.56	91.10	92.25	92.45	**97.19**
OA (%)	-	78.89	84.83	85.25	88.66	89.95	84.38	87.17	94.02	**98.21**
$k\times100$	-	73.41	79.12	80.77	85.15	86.78	79.93	83.31	91.91	**97.57**
Params	-	-	242.32K	53.03K	33.10K	7.47M	8.43K	8.43K	6.81K	43.86K

TABLE VIII
CLASSIFICATION ACCURACIES OBTAINED BY DIFFERENT ALGORITHMS FOR KSC DATASET

NO.	Land-Cover Type	RBF-SVM	2-D-CNN	3-D-CNN	PPF-CNN	DR-CNN	GCN	S^2GCN	MDGCN	X-GPN
01	Scrub	84.68	88.65	97.95	**100.00**	**100.00**	97.26	95.21	**100.00**	**100.00**
02	Willow swamp	80.75	73.24	87.32	**100.00**	**100.00**	91.08	92.02	**100.00**	99.53
03	CP hammock	86.73	95.13	89.38	85.61	98.23	96.02	99.56	**100.00**	**100.00**
04	Slash pine	68.33	75.57	44.34	84.95	**95.71**	77.83	88.24	83.71	85.07
05	Oak/Broadleaf	56.49	**97.71**	48.85	39.38	62.33	83.21	72.52	46.56	94.66
06	Hardwood	64.82	44.22	83.42	**100.00**	**100.00**	74.87	84.42	98.49	91.46
07	Swamp	53.33	62.67	66.67	68.81	60.00	92.00	89.33	**100.00**	**100.00**
08	Graminoid	75.06	63.09	88.53	77.71	82.85	96.01	96.76	**100.00**	**100.00**
09	Spartina marsh	87.76	98.57	84.90	**100.00**	**100.00**	78.37	91.22	**100.00**	98.78
10	Cattail marsh	79.36	81.23	81.23	**100.00**	**100.00**	89.28	95.44	98.12	99.73
11	Salt marsh	98.71	82.01	87.92	**100.00**	**100.00**	**100.00**	88.95	**100.00**	89.20
12	Mud flats	90.05	93.23	76.11	97.93	97.53	95.35	98.94	99.79	**100.00**
13	Water	97.88	**100.00**	99.55	**100.00**	**100.00**	95.88	99.22	**100.00**	**100.00**
AA (%)	-	78.77	81.18	79.71	88.80	92.05	89.78	91.68	94.36	**96.80**
OA (%)	-	85.00	86.01	86.14	92.78	95.64	91.78	94.15	97.57	**97.78**
$k\times100$	-	83.26	84.31	84.47	91.92	95.12	90.82	93.46	97.28	**97.52**
Params	-	-	913.26K	92.29K	55.82K	8.69M	14.21K	14.21K	11.44K	64.95K

their accuracies are comparatively low. Taking both spectral and spatial information into account, 3-D-CNN and S^2GCN acquire improved performance than 2-D-CNN and GCN, respectively, which verifies the significance of spectral–spatial merged characteristics in HSI. Furthermore, DR-CNN also acquired remarkable results mainly attributed to the exploited diverse spatial information. Most significantly, we can observe that the proposed X-GPN outperforms the SOTA MDGCN by a substantial margin although with more parameters. In concrete, the AA, OA, and k gain encouraging 4.74%, 4.19%, and 5.66% improvements, respectively. It is reasonable to infer that our proposed framework has a better ability to handle the dataset with many irregular boundaries.

In Table VIII, some similar conclusions can be summarized on the KSC dataset. The classification result of the X-GPN method again reaches the top-level among different algorithms regarding the measurements of AA, OA, and k. Moreover, profited by the data augmentation strategy using pixel-pair features and diverse regions, PPF-CNN and DR-CNN also obtain satisfactory performance as achieving 100% right prediction for up to seven land-cover categories. Owing to the preprocessing of the superpixel segmentation, the SOTA MDGCN acquires competitive performance with the least parameters. Nevertheless, X-GPN slightly outperforms MDGCN by 2.44%, 0.21%, and 0.24% in terms of AA, OA, and k, respectively. It indicates that the cross-scale architectures, SBAA module, and prototypical layer together make the proposed network an advanced HSIC model.

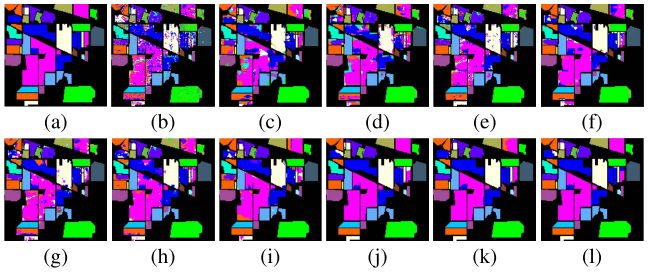


Fig. 9. Classification maps produced by different methods using the InP dataset. (a) Ground truth. (b) RBF-SVM. (c) 2-D-CNN. (d) 3-D-CNN. (e) PPF-CNN. (f) DR-CNN. (g) GCN. (h) S^2GCN. (i) MDGCN. (j) X-GPN w/o CS. (k) X-GPN w/o SBAA. (l) X-GPN. Best in zoomed-in view.

For visually subjective evaluation, the classification maps along with the ground truth for the InP, UP, and KSC datasets are delineated from Figs. 9–11, respectively. Besides, the classification maps obtained by the X-GPN w/o CS and X-GPN w/o SBAA are also depicted to demonstrate their significant discrepancy. Figs. 9–11 are in consistent with Tables V–VIII for various algorithms. In general, it is evident the classification maps obtained by X-GPN have the highest quality since they are closest to the ground truth with the least misclassification. Due to exclusively exploiting the spectral information without the spatial profiles, there exist many scattered salt-and-pepper noises in the maps of RBF-SVM

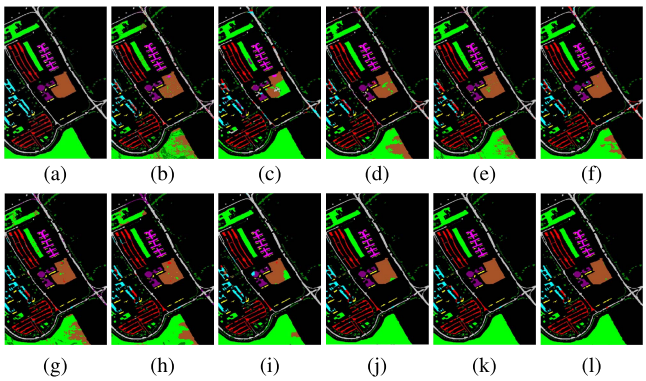


Fig. 10. Classification maps produced by different methods using UP dataset. (a) Ground truth. (b) RBF-SVM. (c) 2-D-CNN. (d) 3-D-CNN. (e) PPF-CNN. (f) DR-CNN. (g) GCN. (h) S^{2}GCN. (i) MDGCN. (j) X-GPN w/o CS. (k) X-GPN w/o SBAA. (l) X-GPN. Best in zoomed-in view.

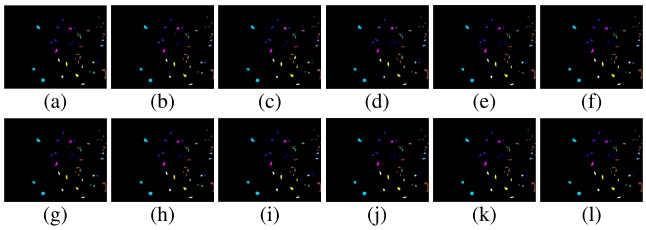


Fig. 11. Classification maps produced by different methods using KSC dataset. (a) Ground truth. (b) RBF-SVM. (c) 2-D-CNN. (d) 3-D-CNN. (e) PPF-CNN. (f) DR-CNN. (g) GCN. (h) S^{2}GCN. (i) MDGCN. (j) X-GPN w/o CS. (k) X-GPN w/o SBAA. (l) X-GPN. Best in zoomed-in view.

and GCN methods. In contrast, the classification maps derived from the spectral–spatial classifiers are more compact and smoother. In particular, due to the consideration of multiscale spatial information, the SOTA MDGCN and our proposed X-GPN can acquire classification results with clearer edges, where the cross-boundary samples are hard to be accurately classified. This implies that the proposed method can precisely extract the distinguishing features to help recognize different materials even with similar characteristics.

E. Performance Under Various Number of Training Samples

In this part, we validate the performance of the proposed X-GPN under various percentages of the training data. To guarantee a reliable conclusion, the competitive methods, i.e., PPF-CNN, DR-CNN, GCN, S^{2}GCN, and MDGCN are also executed under the same training partitions to serve as comparisons. In specific, while keeping the test data as the same as in Tables I–III, 20%–80% training data of each class are randomly chosen from Figs. 4(c)–6(c). The average OA of ten individual tests on the three datasets are delineated in Fig. 12, where the 100% exactly refers to all training samples shown in Figs. 4(c)–6(c). From Fig. 12, it can be observed that our proposed X-GPN always ranks first on the InP and UP datasets as the training set varies. However, it works poorer than the SOTA MDGCN on the KSC dataset when 20%–80% of the training set is used. The possible reason is that the labeled samples of KSC are more intensive than the other two datasets. In this context, the superpixel-level training samples in MDGCN will not reduce that much as the pixel-level training samples decrease, which ensures the stability of the MDGCN on KSC dataset under fewer training samples. Besides, the DR-CNN and S^{2}GCN also demonstrate encouraging performance profited from the diverse scale contextual information and semisupervised graph learning process, respectively. In practice, the proposed X-GPN is capable of combining both advantages to obtain more satisfactory results.

TABLE IX

CLASSIFICATION PERFORMANCE ON THE HIGHER RESOLUTION SAMPLES OF HU2018 DATASET

No.	Train	Test	DR-CNN	S^{2}GCN	MDGCN	X-GPN
01	500	9299	84.13	93.57	92.24	**96.96**
02	500	32002	**96.96**	85.77	82.99	92.94
03	68	616	**100**	**100**	99.84	99.51
04	500	13088	90.38	96.96	95.83	**98.99**
05	500	4548	66.39	89.95	**95.84**	95.49
06	451	4065	94.82	98.03	**100**	99.70
07	26	240	**100**	**100**	97.50	98.75
08	500	39262	79.37	84.92	**95.61**	92.28
09	500	223184	**98.85**	72.21	85.63	94.62
10	500	45310	73.57	45.34	68.63	**73.88**
11	500	33502	63.23	47.29	55.95	**69.59**
12	151	1365	22.34	21.68	**69.16**	52.09
13	500	45858	83.80	58.86	81.63	**90.28**
14	500	9349	80.14	95.21	96.80	**97.08**
15	500	6437	95.66	99.57	98.23	**99.91**
16	500	10975	88.77	90.56	91.84	**97.36**
17	14	135	79.50	**100**	84.44	91.11
18	500	6078	58.54	91.51	84.50	**98.60**
19	500	4865	86.36	96.36	97.90	**98.58**
20	500	6324	72.51	97.96	**100**	99.94
Total	8210	496502	-	-	-	-
AA	-	-	80.77	83.29	88.73	**91.88**
OA	-	-	86.64	72.06	83.71	**90.77**
$k\times100$	-	-	83.02	65.60	79.34	**88.02**

F. Performance on HSI With Higher Resolution Samples

To give a more comprehensive evaluation, the experiments on another higher resolution dataset, i.e., Houston University 2018 (HU2018) with 1-m spatial resolution, are presented. The HU2018 dataset was captured on February 16, 2017, by the CASI-1500 over the area of the Houston University, which is available from the 2018 IEEE GRSS data fusion contest. The large-scale image has 601 $\times$ 2384 pixels with 50 channels sampling the wavelength of between 0.38 and 1.05 μm. The ground truth contains 20 land-cover types and Table IX gives the specific training and test division as [54]. Table IX also reports the classification results of the proposed X-GPN and the strongest competitors, i.e., DR-CNN, S^{2}GCN, and the SOTA MDGCN. It can be observed that the proposed X-GPN works the best on this higher resolution dataset as well. For instance, the X-GPN promotes the accuracies of about 4.13%, 18.71%, and 7.06% in OA compared with DR-CNN, S^{2}GCN, and MDGCN, respectively. Furthermore, Fig. 13 presents the classification maps corresponding to Table IX. The false-color synthetic image, ground truth, and the distribution of the training samples are also depicted for convenient comparison. It can be seen that the X-GPN achieves the optimal consistency with the ground truth, especially for the eighth category Residential, which demonstrates the best classification performance.

G. Influence of Different Number of Scales and Their Sizes

This section analyzes the influence of different number of scales and their sizes on the classification performance, which

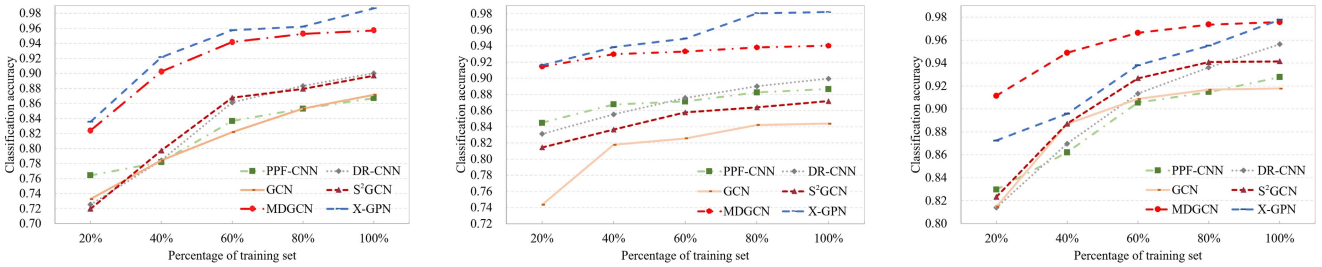


Fig. 12. Classification performance of PPF-CNN, DR-CNN, GCN, S^{2}GCN, MDGCN, and the proposed X-GPN with various number of training samples. (a) InP. (b) UP. (c) KSC.

TABLE X
CLASSIFICATION PERFORMANCE INFLUENCED BY DIFFERENT NUMBER OF SCALES AND SIZES

DataSet		*Num. of Scales* = 1				*Num. of Scales* = 2			*Num. of Scales* = 3		*Num. of Scales* = 4
		3×3	5×5	7×7	9×9	3×3 → 5×5	5×5 → 7×7	7×7 → 9×9	3×3 → 7×7	5×5 → 9×9	3×3 → 9×9
InP	AA	84.43	85.58	91.71	79.76	98.34	98.85	97.86	**99.28**	99.08	98.91
	OA	77.16	77.51	84.90	88.30	97.48	98.03	95.39	**98.68**	98.07	97.05
	k×100	73.08	73.31	82.27	86.90	97.01	97.73	94.73	**98.48**	97.79	96.62
UP	AA	91.29	92.61	91.25	92.47	96.41	95.18	91.95	**97.19**	93.04	96.47
	OA	92.51	93.72	92.16	93.16	97.28	91.76	90.43	**98.21**	94.76	97.10
	k×100	89.64	91.35	89.12	90.53	96.59	89.14	87.19	**97.57**	92.84	96.06
KSC	AA	95.05	95.44	92.97	95.96	96.55	96.28	95.75	**96.80**	94.47	85.68
	OA	96.87	96.00	95.93	96.40	97.17	96.77	96.80	**97.78**	97.16	77.41
	k×100	96.49	95.77	95.44	96.33	96.95	96.63	97.54	**97.52**	96.82	75.16

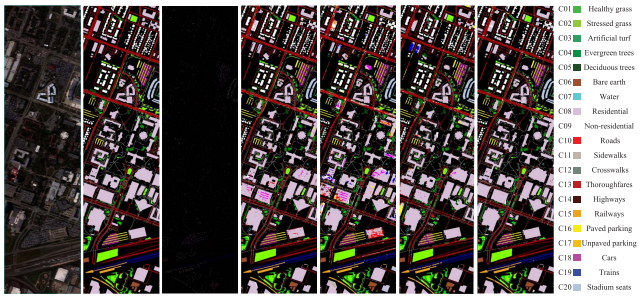


Fig. 13. HU2018 dataset. (a) False-color synthetic image (21-R, 12-G, and 5-B bands). (b) Ground truth. (c) Specific training samples. (d) DR-CNN. (e) S^{2}GCN. (f) MDGCN. (g) X-GPN. Best in zoomed-in view.

plays an important role as the beginning of the frameworks. Specifically, the scales of 3 × 3, 5 × 5, 7 × 7, and 9 × 9 are selected to carry out the experiments. The AA, OA, and k obtained by their sequential combinations from one to four scales are shown in Table X. It can be seen that the utilization of multiscale adjacency matrices achieves the best result when the number of scales is set to three and their sizes are 3 × 3, 5 × 5, and 7 × 7 for all datasets. It indicates that fewer scales (i.e., one or two scales) cannot adequately exploit the rich spectral–spatial information, while more scales and large size may bring side effects to the classifier. For example, the overfitting phenomenon occurs when four scales are employed for the KSC dataset.

H. Feature Visualization With Different Loss Functions

To assess the superiority of the proposed prototypical layer, we test the backbone framework as in Fig. 2 with: 1) softmax-based cross-entropy loss; 2) DCE loss; and

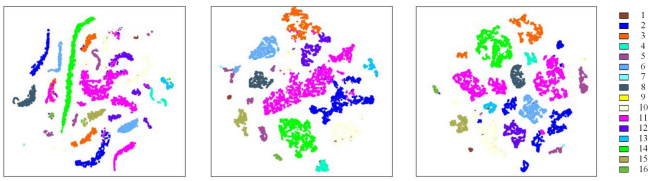


Fig. 14. Feature visualization with different loss functions for InP dataset. (a) Softmax (OA = 90.98%). (b) DCE (OA = 95.89%). (c) DCE + TER (OA = 98.68%).

3) DCE + TER loss functions, respectively. For convenient analysis, the generated embedding features of InP datasets before the softmax function or in the prototypical layer are visualized in Fig. 14 by using the t-SNE method [55]. Correspondingly, the obtained OAs are 90.98%, 95.89%, and 98.68%, respectively. From Fig. 14, the different classification performances can be interpreted. First, we can find that the feature acquired by the softmax function are entangled with each other, and the intraclass distance is larger than that of the interclass for most categories. It will apparently bring difficulties for accurate classification. In contrast, as shown in Fig. 14(b), the utilized DCE loss can make the intraclass features more compact and interclass more separable, which facilitates distinguishing different land covers precisely. Moreover, from Fig. 14(c), it is observed that the devised TER can further force the intraclass features to be closer. As a result, the boundaries between different land-cover classes are clearer to achieve more accurate identification.

I. Analysis of the BatchNorm and Computational Cost

Taking inspiration from the widely used BatchNorm in the standard CNNs, we conduct the BatchNorm overall graph

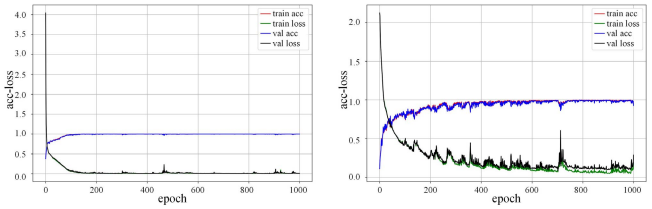


Fig. 15. Train/validation loss and accuracy of X-GPN with BatchNorm and w/o BatchNorm for UP dataset. (a) X-GPN with BatchNorm, (b) X-GPN w/o BatchNorm.

TABLE XI
COMPUTATIONAL COST OF THE TRAINING AND TEST PROCEDURE OF VARIOUS METHODS

Methods		InP	UP	KSC
PPF-CNN	Training time (s)	840.74	3019.45	1111.42
	Test time (s)	2.86	9.36	2.11
DR-CNN	Training time (s)	1756.15	5219.96	954.56
	Test time (s)	323.30	122.49	208.81
GCN	Training time (s)	192.16	329.45	42.57
	Test time (s)	0.02	0.05	0.01
S^2GCN	Training time (s)	251.61	595.09	50.98
	Test time (s)	0.03	0.10	0.02
MDGCN	Training time (s)	62.54	350.47	24.75
	Test time (s)	1.11	5.84	0.13
X-GPN	Training time (s)	289.22	1090.46	493.94
	Test time (s)	0.12	0.51	0.21

nodes in our X-GPN frameworks. In order to verify its effectiveness, we examine the X-GPN without BatchNorm for comparison. In detail, the train/validation loss and accuracy on the UP dataset during the training procedure are shown in Fig. 15. Comparing Fig. 15(a) with (b), we can find that the BatchNorm can evidently facilitate quicker convergence. Concretely, X-GPN with BatchNorm is convergent at around the 150th epoch. By contrast, the train/validation loss curves of X-GPN without BatchNorm have much vibration and slightly stabilize until around the 600th epoch. For the frameworks without BatchNorm, even a bigger epoch number can only achieve inferior capability.

Moreover, the computational cost of X-GPN and the competitive compared algorithms PPF-CNN, DR-CNN, and the graph-based GCN, S^2GCN, and MDGCN on the InP, UP, and KSC datasets are reported in Table XI. From Table XI, it can be observed that DR-CNN expends much time due to pretraining of the subnetworks in the training phase and diverse region calculation in the test stage. PPF-CNN is also time-consuming resulting from a large number of augmented samples. On the contrary, GCN and S^2GCN are efficiently benefited to their relatively simple network architectures. And the SOTA MDGCN is the fastest method due to superpixels squeezing the data volume. Overall, our proposed X-GPN is faster than PPF-CNN and DR-CNN, and although X-GPN takes more time than GCN, S^2GCN, and MDGCN, it is acceptable to acquire more extraordinary performance.

V. CONCLUSION

In this article, a novel semisupervised X-GPN is presented for HSI classification. In the proposed X-GPN, we take the neighborhoods of three scales into account to construct the adjacency matrices and then investigate the abundant spectral–spatial features through graph convolutions in a transductive manner. In particular, in order to exploit the complementary information between different scales, we utilize the lightweight standard 1-D-Conv to explore the intranode dependence and then concatenate the output with the features produced by the other two scales. Furthermore, we design an SBAA module to automatically aggregate the multistream deep node features. In the above-mentioned two ways, the multiscale information can not only interact in the middle-level but also be flexibly integrated into the relatively high level, which is more favorable to generate comprehensive representations. Moreover, we devise a new prototypical layer, including the DCE loss and TER to enhance the discrimination of the extracted features, i.e., intraclass compact and interclass isolated, which can further promote the classification accuracy. Overall, abundant experiment results demonstrate that our proposed frameworks can achieve superior performance than the classic and SOTA algorithms. In future work, we will develop lightweight networks to reduce the memory usage and model complexity while maintaining the classification capability.

REFERENCES

[1] X. X. Zhu *et al.*, "Deep learning in remote sensing: A comprehensive review and list of resources," *IEEE Geosci. Remote Sens. Mag.*, vol. 5, no. 4, pp. 8–36, Dec. 2017.
[2] G. Camps-Valls, D. Tuia, L. Bruzzone, and J. A. Benediktsson, "Advances in hyperspectral image classification: Earth monitoring with statistical learning methods," *IEEE Signal Process. Mag.*, vol. 31, no. 1, pp. 45–54, Jan. 2014.
[3] M. Fauvel, Y. Tarabalka, J. A. Benediktsson, J. Chanussot, and J. C. Tilton, "Advances in spectral-spatial classification of hyperspectral images," *Proc. IEEE*, vol. 101, no. 3, pp. 652–675, Mar. 2013.
[4] P. Ghamisi *et al.*, "New frontiers in spectral-spatial hyperspectral image classification: The latest advances based on mathematical morphology, Markov random fields, segmentation, sparse representation, and deep learning," *IEEE Geosci. Remote Sens. Mag.*, vol. 6, no. 3, pp. 10–43, Sep. 2018.
[5] L. He, J. Li, C. Liu, and S. Li, "Recent advances on spectral-spatial hyperspectral image classification: An overview and new guidelines," *IEEE Trans. Geosci. Remote Sens.*, vol. 56, no. 3, pp. 1579–1597, Mar. 2018.
[6] J. Li, Q. Du, Y. Li, and W. Li, "Hyperspectral image classification with imbalanced data based on orthogonal complement subspace projection," *IEEE Trans. Geosci. Remote Sens.*, vol. 56, no. 7, pp. 3838–3851, Jul. 2018.
[7] H. Wu and S. Prasad, "Semi-supervised deep learning using pseudo labels for hyperspectral image classification," *IEEE Trans. Image Process.*, vol. 27, no. 3, pp. 1259–1270, Mar. 2018.
[8] K. Ronald, R. Luu, and C. Kanan, "Low-shot learning for the semantic segmentation of remote sensing imagery," *IEEE Trans. Geosci. Remote Sens.*, vol. 56, no. 10, pp. 6214–6223, Oct. 2018.
[9] D. Hong, N. Yokoya, N. Ge, J. Chanussot, and X. X. Zhu, "Learnable manifold alignment (LeMA): A semi-supervised cross-modality learning framework for land cover and land use classification," *ISPRS J. Photogramm. Remote Sens.*, vol. 147, pp. 193–205, Jan. 2019.
[10] S. S. Sawant and M. Prabukumar, "A review on graph-based semi-supervised learning methods for hyperspectral image classification," *Egyptian J. Remote Sens. Space Sci.*, vol. 23, no. 2, pp. 243–248, Aug. 2020.
[11] J. Jiang, J. Ma, and X. Liu, "Multilayer spectral–spatial graphs for label noisy robust hyperspectral image classification," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 33, no. 2, pp. 839–852, Feb. 2022.
[12] K. Cheng, Y. Zhang, X. He, W. Chen, J. Cheng, and H. Lu, "Skeleton-based action recognition with shift graph convolutional network," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 183–192.
[13] T. N. Kipf and M. Welling, "Semi-supervised classification with graph convolutional networks," 2016, *arXiv:1609.02907*.

[14] A. Qin, Z. Shang, J. Tian, Y. Wang, T. Zhang, and Y. Y. Tang, "Spectral-spatial graph convolutional networks for semisupervised hyperspectral image classification," *IEEE Geosci. Remote Sens. Lett.*, vol. 16, no. 2, pp. 241–245, Sep. 2019.

[15] D. Hong, L. Gao, J. Yao, B. Zhang, A. Plaza, and J. Chanussot, "Graph convolutional networks for hyperspectral image classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 59, no. 7, pp. 5966–5978, Jul. 2021.

[16] X. He, Y. Chen, and P. Ghamisi, "Dual graph convolutional network for hyperspectral image classification with limited training samples," *IEEE Trans. Geosci. Remote Sens.*, vol. 60, pp. 1–18, 2022.

[17] J. Bai, B. Ding, Z. Xiao, L. Jiao, H. Chen, and A. C. Regan, "Hyperspectral image classification based on deep attention graph convolutional network," *IEEE Trans. Geosci. Remote Sens.*, vol. 60, pp. 1–16, 2022.

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

[19] 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 2019.

[20] X. Yang, T. Wu, N. Wang, Y. Huang, B. Song, and X. Gao, "HCNN-PSI: A hybrid CNN with partial semantic information for space target recognition," *Pattern Recognit.*, vol. 108, Dec. 2020, Art. no. 107531.

[21] B. Xi *et al.*, "Deep prototypical networks with hybrid residual attention for hyperspectral image classification," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 13, pp. 3683–3700, 2020.

[22] H.-M. Yang, X.-Y. Zhang, F. Yin, and C.-L. Liu, "Robust classification with convolutional prototype learning," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 3474–3482.

[23] J. Ham, Y. Chen, M. M. Crawford, and J. Ghosh, "Investigation of the random forest framework for classification of hyperspectral data," *IEEE Trans. Geosci. Remote Sens.*, vol. 43, no. 3, pp. 492–501, Mar. 2005.

[24] C. Bo, H. Lu, and D. Wang, "Spectral-spatial K-nearest neighbor approach for hyperspectral image classification," *Multimedia Tools Appl.*, vol. 77, no. 9, pp. 10419–10436, 2018.

[25] J. Li, B. Xi, Q. Du, R. Song, Y. Li, and G. Ren, "Deep kernel extreme-learning machine for the spectral–spatial classification of hyperspectral imagery," *Remote Sens.*, vol. 10, no. 12, p. 2036, Dec. 2018.

[26] B. Xi, J. Li, Y. Li, R. Song, W. Sun, and Q. Du, "Multiscale context-aware ensemble deep KELM for efficient hyperspectral image classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 59, no. 6, pp. 5114–5130, Jun. 2021.

[27] 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.

[28] G. Camps-Valls and L. Bruzzone, "Kernel-based methods for hyperspectral image classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 43, no. 6, pp. 1351–1362, Jun. 2005.

[29] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, pp. 436–444, Feb. 2015.

[30] N. Audebert, B. L. Saux, and S. Lefevre, "Deep learning for classification of hyperspectral data: A comparative review," *IEEE Geosci. Remote Sens. Mag.*, vol. 7, no. 2, pp. 159–173, Jun. 2019.

[31] M. E. Paoletti, J. M. Haut, J. Plaza, and A. Plaza, "Deep learning classifiers for hyperspectral imaging: A review," *ISPRS J. Photogramm. Remote Sens.*, vol. 158, pp. 279–317, Dec. 2019.

[32] J. Li, B. Xi, Y. Li, Q. Du, and K. Wang, "Hyperspectral classification based on texture feature enhancement and deep belief networks," *Remote Sens.*, vol. 10, no. 3, p. 396, Mar. 2018.

[33] L. Mou, P. Ghamisi, and X. X. Zhu, "Deep recurrent neural networks for hyperspectral image classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 55, no. 7, pp. 3639–3655, Jul. 2016.

[34] B. Xi *et al.*, "Multi-direction networks with attentional spectral prior for hyperspectral image classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 60, pp. 1–15, 2022.

[35] W. Hu, Y. Huang, L. Wei, F. Zhang, and H. Li, "Deep convolutional neural networks for hyperspectral image classification," *J. Sensors*, vol. 2015, pp. 1–12, Jan. 2015.

[36] K. Makantasis, K. Karantzalos, A. Doulamis, and N. Doulamis, "Deep supervised learning for hyperspectral data classification through convolutional neural networks," in *Proc. IEEE Geosci. Remote Sens. Symp.*, Jul. 2015, pp. 4959–4962.

[37] Y. Li, H. Zhang, and Q. Shen, "Spectral–spatial classification of hyperspectral imagery with 3D convolutional neural network," *Remote Sens.*, vol. 9, no. 1, p. 67, Jan. 2017.

[38] W. Li, G. Wu, F. Zhang, and Q. Du, "Hyperspectral image classification using deep pixel-pair features," *IEEE Trans. Geosci. Remote Sens.*, vol. 55, no. 2, pp. 844–853, Feb. 2017.

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

[40] J. Zhou *et al.*, "Graph neural networks: A review of methods and applications," 2018, *arXiv:1812.08434*.

[41] Z. Wu, S. Pan, F. Chen, G. Long, C. Zhang, and P. S. Yu, "A comprehensive survey on graph neural networks," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 32, no. 1, pp. 4–24, Jan. 2020.

[42] J. Bruna, W. Zaremba, A. Szlam, and Y. LeCun, "Spectral networks and locally connected networks on graphs," 2013, *arXiv:1312.6203*.

[43] D. K. Hammond, P. Vandergheynst, and R. Gribonval, "Wavelets on graphs via spectral graph theory," *Appl. Comput. Harmon. Anal.*, vol. 30, no. 2, pp. 129–150, Mar. 2011.

[44] J. Chen, L. Jiao, X. Liu, L. Li, F. Liu, and S. Yang, "Automatic graph learning convolutional networks for hyperspectral image classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 60, pp. 1–16, 2022.

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

[46] K. Xu, W. Hu, J. Leskovec, and S. Jegelka, "How powerful are graph neural networks?" 2018, *arXiv:1810.00826*.

[47] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," *Commun. ACM*, vol. 60, no. 6, pp. 84–90, May 2017.

[48] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: A simple way to prevent neural networks from overfitting," *J. Mach. Learn. Res.*, vol. 15, no. 1, pp. 1929–1958, Jan. 2014.

[49] L. Mou, X. Lu, X. Li, and X. X. Zhu, "Nonlocal graph convolutional networks for hyperspectral image classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 58, no. 12, pp. 8246–8257, Dec. 2020.

[50] J. Liang, J. Zhou, Y. Qian, L. Wen, X. Bai, and Y. Gao, "On the sampling strategy for evaluation of spectral-spatial methods in hyperspectral image classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 55, no. 2, pp. 862–880, Feb. 2017.

[51] C. Shi and C.-M. Pan, "Superpixel-based 3D deep neural networks for hyperspectral image classification," *Pattern Recognit.*, vol. 74, pp. 600–616, Feb. 2018.

[52] S. Zeng, Z. Wang, C. Gao, Z. Kang, and D. Feng, "Hyperspectral image classification with global–local discriminant analysis and spatial–spectral context," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 11, no. 12, pp. 5005–5018, Dec. 2018.

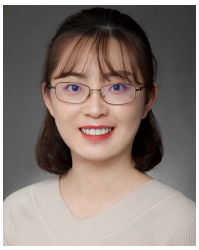
[53] X. He, Y. Chen, and P. Ghamisi, "Heterogeneous transfer learning for hyperspectral image classification based on convolutional neural network," *IEEE Trans. Geosci. Remote Sens.*, vol. 58, no. 5, pp. 3246–3263, May 2019.

[54] D. Hong, X. Wu, P. Ghamisi, J. Chanussot, N. Yokoya, and X. X. Zhu, "Invariant attribute profiles: A spatial-frequency joint feature extractor for hyperspectral image classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 58, no. 6, pp. 3791–3808, Jun. 2020.

[55] L. van der Maaten and G. Hinton, "Visualizing data using t-SNE," *J. Mach. Learn. Res.*, vol. 9, pp. 2579–2605, Nov. 2008.

Bobo Xi (Graduate Student Member, IEEE) received the B.S. degree in information engineering from the School of Telecommunications Engineering, Xidian University, Xi'an, China, in 2017, where he is currently pursuing the Ph.D. degree with the State Key Laboratory of Integrated Service Networks.

His research interests include hyperspectral image processing, machine learning, and deep learning.



Jiaojiao Li (Member, IEEE) received the B.E. degree in computer science and technology, the M.S. degree in software engineering, and the Ph.D. degree in communication and information systems from Xidian University, Xi'an, China, in 2009, 2012, and 2016, respectively.

She was an exchange Ph.D. Student with Mississippi State University, Starkville, MS, USA, under the supervision of Dr. Qian Du. She is currently an Associate Professor with the State Key Laboratory of Integrated Service Networks, School of Telecommunications, Xidian University. Her research interests include hyperspectral remote sensing image analysis and processing, pattern recognition, and data compression.

Yunsong Li (Member, IEEE) received the M.S. degree in telecommunication and information systems and the Ph.D. degree in signal and information processing from Xidian University, Xi'an, China, in 1999 and 2002, respectively.

In 1999, he joined the School of Telecommunications Engineering, Xidian University, where he is currently a Professor, and also the Director of the State Key Laboratory of Integrated Service Networks, Image Coding and Processing Center. His research interests include image and video processing, hyperspectral image processing, and high-performance computing.

Rui Song (Member, IEEE) received the M.S. and Ph.D. degrees in signal and information processing from Xidian University, Xi'an, China, in 2006 and 2009, respectively.

He is currently a Professor with the State Key Laboratory of Integrated Service Networks, School of Telecommunications, Xidian University. His research interests include video coding algorithms, VLSI architecture design, and 3-D reconstruction.

Yuchao Xiao received the B.E. degree in communication engineering from the School of Telecommunications Engineering, Xidian University, Xi'an, China, in 2021. He is currently pursuing the M.D. degree in information and communication engineering with the School of Artificial Intelligence, Beijing University of Posts and Telecommunications, Beijing, China.

Since 2019, he has been a Research Assistant with the State Key Laboratory of Integrated Service Network, Image Coding and Processing Center, Xidian University. His research interests include hyperspectral image processing, target identification, and deep learning.

Qian Du (Fellow, IEEE) received the Ph.D. degree in electrical engineering from the University of Maryland at Baltimore, Baltimore, MD, USA, in 2000.

She is currently the Bobby Shackouls Professor with the Department of Electrical and Computer Engineering, Mississippi State University, Starkville, MS, USA. Her research interests include hyperspectral remote sensing image analysis and applications, pattern classification, data compression, and neural networks.

Dr. Du is a fellow of the International Society for Optics and Photonics (SPIE). She is currently a member of the IEEE Periodicals Review and Advisory Committee and the SPIE Publications Committee. She was a recipient of the 2010 Best Reviewer Award from the IEEE Geoscience and Remote Sensing (GRSS) Society. She was the Co-Chair of the Data Fusion Technical Committee of the IEEE GRSS from 2009 to 2013, the Chair of the Remote Sensing and Mapping Technical Committee of the International Association for Pattern Recognition from 2010 to 2014, and the General Chair of the Fourth IEEE GRSS Workshop on Hyperspectral Image and Signal Processing: Evolution in Remote Sensing, Shanghai, China, in 2012. She was an Associate Editor of the IEEE JOURNAL OF SELECTED TOPICS IN APPLIED EARTH OBSERVATION AND REMOTE SENSING, the *Journal of Applied Remote Sensing*, and the IEEE SIGNAL PROCESSING LETTERS. From 2016 to 2020, she was the Editor-in-Chief of the IEEE *Journal of Selected Topics in Applied Earth Observation and Remote Sensing*.

Jocelyn Chanussot (Fellow, IEEE) received the M.Sc. degree in electrical engineering from the Grenoble Institute of Technology (Grenoble INP), Grenoble, France, in 1995, and the Ph.D. degree in electrical engineering from the Universite de Savoie, Annecy, France, in 1998.

Since 1999, he has been with Grenoble INP, where he is currently a Professor of signal and image processing. He was a Visiting Scholar with Stanford University, Stanford, CA, USA, the KTH Royal Institute of Technology, Stockholm, Sweden, and the National University of Singapore, Singapore. Since 2013, he has been an Adjunct Professor with the University of Iceland, Reykjavík, Iceland, and the Aerospace Information Research Institute, Chinese Academy of Sciences, Beijing, China. From 2015 to 2017, he was a Visiting Professor with the University of California at Los Angeles, Los Angeles, CA, USA. His research interests include image analysis, hyperspectral remote sensing, data fusion, machine learning, and artificial intelligence.

Prof. Chanussot was a member of the Institut Universitaire de France from 2012 to 2017. He holds the AXA Chair in remote sensing with the Aerospace Information Research Institute, Chinese Academy of Sciences. He was the founding President of the IEEE Geoscience and Remote Sensing French chapter from 2007 to 2010, which received the 2010 IEEE GRSS Chapter Excellence Award. He was a member of the Machine Learning for Signal Processing Technical Committee of the IEEE Signal Processing Society from 2006 to 2008. He was the Vice-President of the IEEE Geoscience and Remote Sensing Society, in charge of meetings and symposia, from 2017 to 2019. He has received multiple outstanding paper awards. He was the General Chair of the first IEEE GRSS Workshop on Hyperspectral Image and Signal Processing, Evolution in Remote Sensing. He was the Chair and the Co-Chair of the GRS Data Fusion Technical Committee from 2009 to 2011 and from 2005 to 2008, respectively. He was the Program Chair of the IEEE International Workshop on Machine Learning for Signal Processing in 2009. He was the Editor-in-Chief of the IEEE JOURNAL OF SELECTED TOPICS IN APPLIED EARTH OBSERVATIONS AND REMOTE SENSING from 2011 to 2015. In 2014, he served as a Guest Editor for the *IEEE Signal Processing Magazine*. He is an Associate Editor of the IEEE TRANSACTIONS ON GEOSCIENCE AND REMOTE SENSING, the IEEE TRANSACTIONS ON IMAGE PROCESSING, and the PROCEEDINGS OF THE IEEE. He was a Highly Cited Researcher (Clarivate Analytics/Thomson Reuters).