Joint Correlation Alignment-Based Graph Neural Network for Domain Adaptation of Multitemporal Hyperspectral Remote Sensing Images

Wenjin Wang, *Student Member, IEEE*, Li Ma, *Member, IEEE*, Min Chen, *Student Member, IEEE*, and Qian Du, *Fellow, IEEE*

***Abstract*—In this article, we propose a novel deep domain adaptation method based on graph neural network (GNN) for multitemporal hyperspectral remote sensing images. In GNN, graphs are constructed for source and target data, respectively. Then the graphs are utilized in each hidden layer to obtain features. GNN operates on graph structure and the relations between data samples can be exploited. It aggregates features and propagate information through graph nodes. Thus, the extracted features have an improved smoothness in each spectral neighborhood which is beneficial to classification. Furthermore, the domain-wise correlation alignment (CORAL) and class-wise CORAL are jointly embedded in GNN network to achieve a joint distribution adaptation performance. By introducing the joint CORAL strategy in GNN, the extracted features can not only be aligned between domains but also have a superior discriminability in each domain. This domain adaptation network is named as joint CORAL-based graph neural network. Experiments using multitemporal Hyperion and NSF-funded center for airborne laser mapping datasets demonstrate the effectiveness of the proposed method.**

***Index Terms*—Classification, domain adaptation, graph neural network (GNN), hyperspectral remote sensing.**

I. Introduction

MULTITEMPORAL hyperspectral remote sensing images are increasingly available for data analysis and applications [1]. For hyperspectral remote sensing image classification [2], it is cost and difficult to collect labels. If there are labeled data available from other related images and they can be reused to classify a new image, the classification can be achieved without further labeling cost. Multitemporal hyperspectral images are well related and thus it is appealing to reuse the labeled knowledge of a previously acquired image for classification of a new image. However, a classifier directly trained by the labeled data from one image would present poor performance on a new image, since there exists spectral shift between the two images. Spectral properties are affected by a lot of factors for hyperspectral images, such as the changed season, soil moisture, vegetation composition, topography, illumination, and the acquisition angle of the sun [3]–[5]. Domain adaptation is able to solve the problem. It aims to align distributions across domains and obtain an adaptative classifier for the target image by transferring knowledge from previous images. The previous image with abundant labeled data is called source domain, and the new image with few or no labeled data are called target domain.

Over the past few years, many shallow domain adaptation approaches have been proposed in the remote sensing community, and most of them can be classified into three major categories including instance-based methods, classifier-based methods and feature-based methods. The instance-based methods bridge source and target domains by reweighting data instances [6]–[8]. The classifier-based methods attempt to learn a classifier which has good performance on target domain by transferring labeled information from source domain [9]–[11]. The feature-based methods are the most popular strategy for shallow domain adaptation. They aim to learn a shared feature representation where spectral drift between domains can be minimized [12]–[16].

Recently, deep learning has been applied to domain adaptation because of its excellent feature representation ability. Compared with shallow domain adaptation approaches, end-to-end deep domain adaptation approaches transfer more effective knowledge by embedding adaptation modules in the network architecture. Maximum mean discrepancy (MMD) is a popular distribution alignment strategy, which is utilized in many deep domain adaptation methods. Tzeng *et al.* [17] proposed the deep domain confusion method, which firstly regularizes the single adaptation layer of deep neural network (DNN) using linear-kernel MMD. Similar to MMD, correlation alignment (CORAL) could also measure the distribution distance between domains, and it attempts to align the second-order statistics of the source and target features. Sun *et al*. [18] extended the CORAL to deep architectures and proposed the CORAL for deep domain adaptation (D-CORAL) approach, so that a nonlinear transformation that aligns the correlations of features between the two domains is obtained. Besides, adversarial learning is

Manuscript received December 7, 2020; revised January 14, 2021; accepted February 25, 2021. Date of publication March 3, 2021; date of current version March 25, 2021. This work was supported in part by National Natural Science Foundations of China under Grant 61771437, Grant 61102104, and Grant 91442201. *(Corresponding author: Li Ma.)*

Wenjin Wang, Li Ma, and Min Chen are with the School of Mechanical Engineering and Electronic Information, China University of Geosciences, Wuhan 430074, China (e-mail: crazyjin1996@cug.edu.cn; mali@cug.edu.cn; suprecm7@cug.edu.cn).

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

Digital Object Identifier 10.1109/JSTARS.2021.3063460

another increasingly popular deep domain adaptational strategy and it learns the domain-invariant features in a two-player game inspired by generative adversarial networks (GANs) [19]. Ganin *et al.* [20] first introduced adversarial learning into deep domain adaptation and proposed domain adversarial neural network (DANN), where a domain discriminator is used to obtain the domain-invariant features.

Deep learning has also been applied for domain adaptation of hyperspectral remote sensing images. Wang *et al*. [21] proposed a domain adaptation method by learning the manifold embedding and matching the discriminative distribution in source domain with neural network. Chen *et al*. [22] employed the criterion of cycle consistency to achieve features that are both domain-invariant and discriminative. Liu *et al.* [23] proposed a class-wise adversarial networks and achieved a superior feature-alignment performance. Gross *et al*. [24] proposed a nonlinear feature normalization alignment approach for domain adaptation of multitemporal hyperspectral images, which is able to mitigate nonlinear effects in hyperspectral data and transfer spectral features from one domain to another. Saha *et al*. [25] exploited a cycle-consistent GAN to learn deep transcoding between multisensory and multitemporal domains.

Deep domain adaptation methods usually utilize full connected network for feature extraction of hyperspectral data [21]–[23], [26], which cannot exploit the relations between data points. Some domain adaptation networks that utilize CNN for feature extraction are able to extract spatial spectral information [24], [25]. However, CNN is conducted on small patches, which are usually squared spatial windows, and thus the ability in modeling the relations between data points are still limited [27]. Moreover, the classification of pixels on the boundary between different classes may be affected. For the pixel on the boundary, the patch around this pixel contains signals from multiple classes. Thus, the CNN-extracted features cannot accurately represent the class of the central pixel, but are mixed features from multiple classes.

Recently, graph neural networks (GNNs) have been received significant attentions. GNN operates on graph structure and is able to aggregate features and propagate information through graph nodes. Compared to CNN that exploits the local spatial information, GNN utilizes graph to model the data relations in a larger range. The connected samples in the graph are spectral neighbors that have similar spectral properties and may not spatially adjacent, and thus GNN can model longer-range spatial relations than CNN, and the edge information can also be well preserved [27]–[29]. Moreover, GNN not only feeds pixelwise samples into network which is very suitable for pixel level hyperspectral image classification [28], but also feeds a graph into network to exploit the relations between data. The new feature of a node is a weighted sum of its neighboring nodes on the graph. Therefore, the nodes in a neighborhood are more likely to have similar features, and the smoothness of each spectral neighborhood is improved [29].

GNN was first proposed by Gori *et al*. [30], and the early study propagated the information of neighboring nodes via a recurrent neural architecture in an iterative manner, which is computationally expensive. To overcome these problems, many methods considered to re-define the GNN architecture by adopting the principle of convolution recently. Bruna *et al.* [31] proposed the definition of graph convolution based on spectral graph theory and updated the representation of graph by utilizing the graph Fourier transform. Defferrard *et al.* [32] adopted the Chebyshev polynomial to reduce the computational complexity of graph convolution in [31] and considered the K-localized spectral filter, which provided a faster forward propagation. Kipf *et al*. [33] proposed a first-order approximation of spectral graph filter and designed a simple layer-wise propagation rule, which achieved excellent performance in the semisupervised classification task of graph structure data.

Lately, some works applied GNN for hyperspectral remote sensing data analysis. Mou *et al.* [29] proposed a semisupervised nonlocal GNN for hyperspectral data classification. Tong *et al.* [35] introduced an attention-weighted graph into the GNN for hyperspectral image few-shot classification. Hong *et al*. [28] developed a minibatch GNN, and then combine the convolutional neural network and GNN with several fusion strategies. The proposed method allows us to train a large-scale network in a minibatch fashion and achieves out-of-sample extension without retraining network. Wan *et al*. [27] proposed a multiscale dynamic GNN, where the graph is dynamically updated during the training process. They further proposed a context-aware dynamic GNN that can capture the long-range contexture relations in hyperspectral data [36].

Due to the advantages of GNN [27]–[29], [34], [36], we applied GNN for feature extraction of multitemporal hyperspectral remote sensing images in this article, which is able to effectively exploit spectral relational information in images. However, GNN can extract features but is incapable of reducing domain shift or conducting domain adaptation. Therefore, domain adaptation strategy should be introduced to make the GNN network generate domain-invariant features. In the proposed domain adaptation network, we exploit GNN as a feature extractor to generate features for both source and target data, and employ the CORAL domain adaptation strategy for knowledge transfer.

It is worth noting that CORAL strategy only focuses on minimizing the domain-level distribution discrepancy, without considering the class-level information among source and target domain. For multitemporal hyperspectral remote sensing data, different land cover types may have different spectral shift. Adaptation of the domain-level distribution is not equivalent to aligning their class conditional distributions. Since the domain-wise CORAL cannot explore the class-level relations between the source and target domains, the class-wise CORAL is utilized to further reduce the distribution discrepancy on a per-class basis, which is able to adapt the conditional distributions across domains. Combining the two alignment strategies can achieve a joint distribution adaptation performance. The CORAL conducted on each class is called class-wise CORAL. We jointly conduct domain-wise CORAL and class-wise CORAL to obtain marginal and conditional distribution adaptation results. Our transfer network that exploits joint CORAL-based GNN is denoted as JCGNN in this article.

The main contributions of the proposed JCGNN method are summarized as follows.

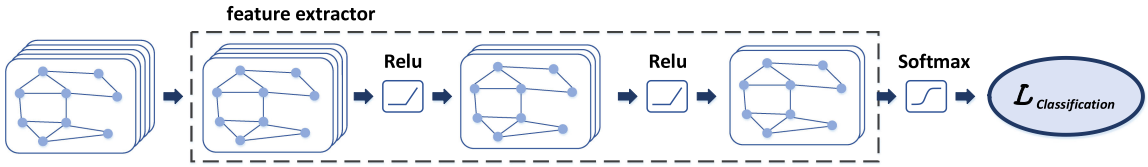

Fig. 1. Schematic depiction of the GNN.

1) To our best knowledge, this is the first attempt to introduce GNN and class-wise CORAL into unsupervised domain adaptation of multitemporal hyperspectral remote sensing images.
2) We apply GNN as the feature extractor for domain adaptation, which considers not only the information among spectral bands, but also the relations among neighboring points. It is able to obtain an improved smoothness in each spectral neighborhood and is beneficial to classification.
3) We introduce joint CORAL domain adaptation strategy to the GNN to achieve both domain-level and class-level domain adaptation performance.

The remainder of this article is organized as follows. Section II provides introduction of unsupervised domain adaptation and GNNs. Section III describes the proposed JCGNN in detail. Section IV discusses the related works. Experimental results are shown in Section V and the conclusion is drawn in Section VI.

II. Background

A. Unsupervised Domain Adaptation

Unsupervised domain adaptation approach assumes there are abundant labeled data in source domain, but no labeled data in target domain. Due to the domain shift across domains, a classifier that directly trained by source domain data would have an unsatisfactory classification performance for target domain data. In this article, we aim to align the data distribution across domains by jointly conducting domain-wise CORAL and class-wise CORAL to obtain marginal and conditional distribution adaptation performance. The joint CORAL domain adaptation strategy is introduced to GNN to achieve unsupervised domain adaptation.

Supposed the source domain data is denoted as $\mathbf{X}_s \in R^{N_s \times d}$ with labels $\mathbf{Y}_s \in R^{N_s \times 1}$ and the target domain data is denoted as $\mathbf{X}_t \in R^{N_t \times d}$ without labeled information, where N_s and N_t represent the number of source and target features, respectively, d is the feature dimensionality. $\mathbf{X}_s$ and $\mathbf{X}_t$ are distributed differently, and the domain adaptation aims to obtain the domain-invariant features of the two-domain data. Then the classifier trained on the source data in the comment feature space can be directly applied for classification of target data.

B. Graph Neural Networks

The network structure of GNN is illustrated in Fig. 1 [37], where three hidden layers are used. GNN employs graph to capture the relations between data points, and the graph is utilized in each hidden layer to obtain features. Softmax classifier is applied to the features from the last hidden layer to conduct classification.

In GNN, graph construction is first conducted before the network training [38]. Let $\mathbf{X} \in R^{N \times d}$ represent a dataset, where N denotes the number of data points and d represents the dimensionality. The graph constructed on $\mathbf{X}$ is denoted as G = ($\mathbf{X}$, $\mathbf{E}$, $\mathbf{A}$) with N nodes, where each node corresponds to a data point $\mathbf{x}_i \in \mathbf{X}$, $e_{ij} = (\mathbf{x}_i, \mathbf{x}_j) \in \mathbf{E}$ is the set of edges and $\mathrm{A} \in R^{N \times N}$ is the graph adjacency matrix.

The graph construction process generally contains two steps: determination of graph adjacency relationships and calculation of graph edge weights. For graph adjacency construction, we utilized k nearest neighbors method to obtain a sparse graph. For each data point $\mathbf{x}_i \in \mathbf{X}$, we calculated the Euclidean distances between $\mathbf{x}_i$ all the other data points in $\mathbf{X}$, and then selected its k nearest neighbors with the shortest Euclidean distances. In graph G, node $\mathbf{x}_i$ is only connected to its k nearest neighbors with edges and a sparse graph is thus achieved. For edge weight calculation, Gaussian kernel function is adopted to measure the similarity between two connected nodes. Suppose $\mathbf{x}_i$ and $\mathbf{x}_j$ are connected neighbors, and the edge weight between them is defined as [39]

$$\mathbf{A}_{ij} = \begin{cases} \exp^{-\frac{dist(\mathbf{x}_i, \mathbf{x}_j)}{\sigma^2}}, & \mathbf{x}_j \in N(\mathbf{x}_i) \text{ or } \mathbf{x}_i \in N(\mathbf{x}_j) \\ 0, & \text{otherwise} \end{cases} \tag{1}$$

where $N(\mathbf{x}_i)$ denotes the k nearest neighbors of $\mathbf{x}_i$, $\mathbf{A}_{ij}$ denotes the element of adjacency matrix $\mathbf{A}$ in row i and column j. σ is the Gaussian diffusion kernel parameter, and dist($\mathbf{x}_i$,$\mathbf{x}_j$) denotes the Euclidean distances between $\mathbf{x}_i$ and $\mathbf{x}_j$ which is calculated as

$$\mathrm{dist}(\mathbf{x}_i, \mathbf{x}_j) = \sqrt{\sum_{k=1}^{d} (\mathbf{x}_{ik} - \mathbf{x}_{jk})^2}. \tag{2}$$

It is worth noting that the adjacency matrix $\mathbf{A}$ represents the relational information of spectrum between pixels and the construction process do not need any labeled information.

Based on the constructed graph, the features extracted in the $(l+1)$th layer of GNN is denoted as [33]

$$\mathbf{H}^{(l+1)} = g\left(\hat{\mathbf{A}}\mathbf{H}^{(l)}\mathbf{\Theta}^{(l)}\right) \tag{3}$$

where $\mathbf{H}^{(l)} \in R^{N \times F_l}$ is the features in the lth layer and $\mathbf{H}^{(0)} = \mathbf{X}$, F_l is the number of nodes in the lth layer and $F_0 = d$, $\mathbf{\Theta}^{(l)} \in R^{F_l \times F_{l+1}}$ is the network parameters. $\mathrm{g}(\cdot)$ denotes the ReLU activation function. $\hat{\mathbf{A}}$ is the normalized graph adjacency matrix, which is defined as [33]

$$\hat{\mathbf{A}} = \tilde{\mathbf{D}}^{-\frac{1}{2}} (\mathbf{A} + \mathbf{I}_N) \tilde{\mathbf{D}}^{-\frac{1}{2}} \tag{4}$$

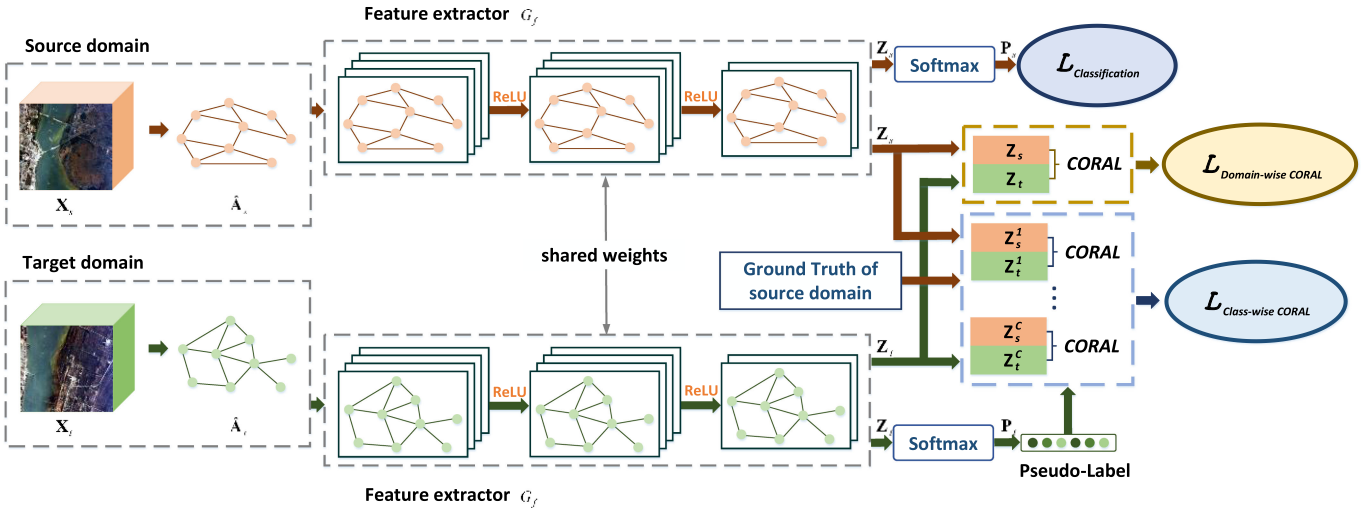

Fig. 2. Schematic depiction of the JCGNN.

where $\tilde{\mathbf{D}}$ is a diagonal matrix and $\tilde{\mathbf{D}}_{ii} = 1+\sum_j \mathbf{A}_{ij}$. $\mathbf{I}_N$ is the identity matrix.

It can be seen from (3) that each GNN layer firstly generates a new feature representation by performing $\hat{\mathbf{A}}\mathbf{H}^{(l)}$ and then feed the features to a fully connected layer characterized by $\mathbf{\Theta}^{(l)}$. The new feature is denoted by [29]

$$\mathbf{T}^{(l)} = \hat{\mathbf{A}}\mathbf{H}^{(l)}. \tag{5}$$

For the *i*th node, the *l*th layer feature and the new feature are denoted as $\mathbf{H}_i{}^{(l)}$ and $\mathbf{T}_i{}^{(l)}$ respectively, and the indices of the *i*th node and its *k* nearest neighbors are denoted as N_i. Its new feature $\mathbf{T}_i{}^{(l)}$ is calculated as

$$\mathbf{T}_i^{(l)} = \hat{\mathbf{A}}_i\mathbf{H}^{(l)} = \sum_{j\in N_i} \hat{\mathbf{A}}_{ij}\mathbf{H}_j^{(l)} \tag{6}$$

where $\hat{\mathbf{A}}_\mathrm{i}$ denotes the normalized weights between the *i*th node and all the nodes in the graph. The new feature of the *i*th node is a weighted sum of its neighboring nodes on the graph. Therefore, the nodes in a neighborhood are more likely to have similar features, and then classified as the same category, which can reduce the within-class differences and thus improve the feature discriminability [29].

Supposing the features of the *i*th data point from the last hidden layer is denoted as $\mathbf{z}^i$, the softmax activation function is applied for classification [33]

$$p^{ic} = \mathrm{softmax}\left(z^{ic}\right) = \frac{\exp\left(z^{ic}\right)}{\sum_{j=1}^{C}\exp\left(z^{ij}\right)} \tag{7}$$

where $\mathbf{z}^{ic}$ represents the *c*th feature of $\mathbf{z}^i$, $i \in \{1,\ldots,N\}$, and p^{ic} is the probability of z^i that belongs to the *c*th class.

III. Proposed Method

The architecture of the proposed JCGNN is shown in Fig. 2, where three-layer GNN is employed for feature extraction. G_f denotes the feature extractor. GNN conducts full-batch network learning. The input of GNN for domain adaptation contains the source labeled data $\mathbf{X}_s \in R^{N_s\times d}$ with its normalized graph adjacent matrix $\hat{\mathbf{A}}_s \in R^{N_s\times N_s}$ and the target data $\mathbf{X}_t \in R^{N_t\times d}$ with its normalized adjacent matrix $\hat{\mathbf{A}}_t \in R^{N_t\times \mathrm{N}_t}$. The source labeled data $\mathbf{X}_s$ is used to train the classifier, and both $\mathbf{X}_s$ and $\mathbf{X}_t$ are employed to obtain the domain-invariant features. Two feature extractors with shared weights are applied to source data and target data, respectively, which can also be regarded as one feature extractor with two inputs. All the four graphs in the first row denote the source graph, where the orange nodes represent source data points and the spectrally similar points are connected by edges. Similarly, the four graphs with green nodes in the second row denote the target graph. The produced domain invariant features are denoted as $\mathbf{Z}_s \in R^{N_s\times C}$ and $\mathbf{Z}_t \in R^{N_t\times C}$ for source data and target data, respectively. $\mathbf{Z}_s$= $[\mathbf{Z}_s{}^1,\mathbf{Z}_s{}^2,\ldots,\mathbf{Z}_s{}^C]$, where $\mathbf{Z}_s{}^i$ denotes the source data of the *i*th class in the feature space. Target data can also be divided into *C* classes according to their pseudo labels. $\mathbf{Z}_t$= $[\mathbf{Z}_t{}^1,\mathbf{Z}_t{}^2,\ldots,\mathbf{Z}_t{}^C]$, where $\mathbf{Z}_t{}^i$ denotes the target data that are predicted as the *i*th class. With the softmax classification, $\mathbf{P}_s \in R^{N_s\times C}$ denote the predicted probability results of source domain and $\mathbf{P}_t \in R^{N_t\times C}$ denote the predicted probability results of target domain. The domain-wise CORAL strategy and class-wise CORAL strategy are combined to constrain the output features between domains to be well aligned.

A. Feature Extraction and Classification by GNN

We apply GNN for feature extraction and classification of source data and target data respectively. For source data $\mathbf{X}_s$ and target data $\mathbf{X}_t$, the graphs G_s and G_t are first constructed. Then two feature extractors with the shard weights are applied to the two-domain data. The three-layer-GNN-generated features are denoted as

$$\mathbf{Z}_s = \hat{\mathbf{A}}_s\mathrm{ReLU}(\hat{\mathbf{A}}_s\mathrm{ReLU}(\hat{\mathbf{A}}_s\mathbf{X}_s\mathbf{\Theta}^{(0)})\mathbf{\Theta}^{(1)})\mathbf{\Theta}^{(2)} \tag{8}$$

$$\mathbf{Z}_t = \hat{\mathbf{A}}_t\mathrm{ReLU}(\hat{\mathbf{A}}_t\mathrm{ReLU}(\hat{\mathbf{A}}_t\mathbf{X}_t\mathbf{\Theta}^{(0)})\mathbf{\Theta}^{(1)})\mathbf{\Theta}^{(2)} \tag{9}$$

Algorithm 1: JCGNN Approach

Input: Source domain data $\mathbf{X}_s \in R^{N_s \times d}$ with labels $\mathbf{Y}_s \in R^{N_s \times 1}$
target domain data $\mathbf{X}_t \in R^{N_t \times d}$.
epochs_1 and *epoch_2* are the numbers of iteration.
Output: The target probability prediction results $\mathbf{P}_t$.
Initialization:
Randomly initializing the parameters of GNN.
Calculating the normalized adjacency matrix $\mathbf{A}_s$ and $\mathbf{A}_t$ by (1).
Training:
stage 1: pre-train and domain-level alignment:
for $i = 1 \rightarrow$ *epochs_1* **do**
Generating embedding of both domains: $\mathbf{Z}_s \leftarrow$ (8) $\mathbf{Z}_t \leftarrow$ (9).
Calculating loss function $\mathrm{L_{DCGNN}}$ using (15).
Updating the parameters of feature extractor G_f by backpropagating through $\mathrm{L_{DCGNN}}$.
end for
stage 2: class-level alignment:
for $i = 1 \rightarrow$ *epochs_2* **do**
Generating embedding of both domains: $\mathbf{Z}_s \leftarrow$ (8) $\mathbf{Z}_t \leftarrow$ (9).
using predicted probability result $\mathbf{P}_t$ to assign pseudo labels for target domain data.
Calculating loss function L_{JCGNN} using (16).
Updating the parameters of feature extractor G_f by backpropagating through L_{JCGNN}.
end for

where $\mathbf{\Theta}^{(0)} \in R^{d \times F_0}$, $\mathbf{\Theta}^{(1)} \in R^{F_0 \times F_1}$ and $\mathbf{\Theta}^{(2)} \in R^{F_1 \times C}$ are the parameters of the three GNN layers, F_0, F_1 are the number of nodes in the first two hidden layers, and C is the number of classes. The softmax activation function in (7) is finally applied to the features $\mathbf{Z}_s$ and $\mathbf{Z}_t$ respectively for classification [33].

Source labeled data are used to calculate the classification loss in this article, which is obtained by the cross-entropy error over all source labeled data

$$L_{\mathrm{Classification}} = -\frac{1}{N_s}\sum_{i=0}^{N_s}\sum_{c=1}^{C} y_s^{ic} \cdot \log p_s^{ic} \tag{10}$$

where y_s^{ic} is the one-hot encoding of the class label for the ith source data point, and p_s^{ic} is the cth output features of softmax activation function.

B. Feature Alignment by Joint CORAL

The domain-wise CORAL reduces the distribution discrepancy among domains by aligning the covariance of the source and target features [18]. The loss measured by domain-wise covariances is defined as

$$L_{\mathrm{Domain-wiseCORAL}} = \frac{1}{4d^2}||\mathbf{C}_s - \mathbf{C}_t||_F^2 \tag{11}$$

where $||\cdot||_{\mathrm{F}}^2$ is the squared matrix Frobenius norm, and $\mathbf{C}_s$ and $\mathbf{C}_t$ denote the covariance matrices of the two-domain data which can be estimated as

$$\mathbf{C}_s = \frac{1}{N_s - 1}\left(\mathbf{Z}_s^T\mathbf{Z}_s - \frac{1}{N_s}\left(\mathbf{1}^T\mathbf{Z}_s\right)^T\left(\mathbf{1}^T\mathbf{Z}_s\right)\right) \tag{12}$$

$$\mathbf{C}_t = \frac{1}{N_t - 1}\left(\mathbf{Z}_t^T\mathbf{Z}_t - \frac{1}{N_t}\left(\mathbf{1}^T\mathbf{Z}_t\right)^T\left(\mathbf{1}^T\mathbf{Z}_t\right)\right) \tag{13}$$

where $\mathbf{1}$ is an all-ones column vector.

Since different classes may have different spectral drifts, the marginal distribution adaptation cannot guarantee the distribution adaptation of each specific class. To achieve better alignment, CORAL is generalized on a per-class basis, which is called class-wise CORAL. The class-wise CORAL loss is defined as

$$L_{\mathrm{Class-wiseCORAL}} = \frac{1}{4d^2C}\sum_{i=1}^{C}\left\|\mathbf{C}_s^i - \mathbf{C}_t^i\right\|_F^2 \tag{14}$$

where C denotes the number of class. $\mathbf{C}_s^i$ and $\mathbf{C}_t^i$ denote the covariance of ith class in the source domain and target domain, respectively, which can be estimated similarly with (11) using class-specific samples in the corresponding domains.

C. Two-Stage Training Procedure

The class-wise CORAL requires the labeled information to calculate the covariance matrix of each class. However, there is no labels in target domain for unsupervised domain adaptation. Therefore, we utilized pseudo labels of target data to obtain covariance matrix of each class. Due to spectral drift, the network trained on source domain data without any domain adaptation would have a poor performance on target domain data, resulting in inferior predicted labels of target data. Since domain-wise CORAL does not require labels and is able to reduce the global distribution difference, it is used to obtain a more accurate pseudo labels of target data than the prediction method without using any domain adaptation strategy. Then the source data with labels and target data with pseudo-labels are used to compute the covariance matrix for each class, respectively.

In this article, we adopt a two-stage training procedure: Firstly, we train a domain-wise CORAL-based graph neural network (DCGNN), by defining the loss function as follows:

$$L_{\mathrm{DCGNN}} = L_{\mathrm{Classification}} + \lambda_1 L_{\mathrm{Domain-wiseCORAL}}. \tag{15}$$

Using DCGNN, the target data are predicted and the pseudo labels are utilized to obtain the covariance matrix of each target class. Second, with the initial pseudo labels obtained by DCGNN, the joint CORAL-based graph neural network (JCGNN) is trained with the loss function composed of joint CORAL loss and classification loss, which is expressed as

$$L_{\mathrm{JCGNN}} = L_{\mathrm{Classification}} + \lambda_1 L_{\mathrm{Domain-wiseCORAL}} + \lambda_2 L_{\mathrm{Class-wiseCORAL}} \tag{16}$$

where the first term denotes the classification loss on source labeled data, the second term denotes the domain-wise CORAL loss and the third term denotes the class-wise CORAL loss. λ_1 and λ_2 are trade-off parameters to balance the contributions of different losses.

Since the joint CORAL can further improve the prediction accuracy, the pseudo labels of target data are updated during each iteration of the training process. The training procedure of JCGNN is summarized in Algorithm 1.

IV. Related Work and Discussion

The relationships between the proposed JCGNN and several related works are described as follows.

1) Deep Adaptation Network (DAN) [40]: The DAN achieves domain adaptation by exploring multikernel MMD for matching different distributions. In the proposed JCGNN, we achieve domain adaptation by using domain-wise CORAL strategy and class-wise CORAL strategy to extract domain invariant features. The MMD strategy and CORAL strategy are all distribution measurement methods of the data, where MMD utilizes first-order statistic and CORAL exploits the second-order statistic. Moreover, DAN only considers the domain-level distribution discrepancy without exploring the class-level information. Our proposed JCGNN jointly takes domain-level and class-level distribution discrepancy into account.

2) Multiple Adversarial Domain Adaptation (MADA) [23]: The MADA is a domain adaptation approach based on adversarial learning and it considers class-level information by using multiple domain discriminators. Our proposed JCGNN jointly embeds domain-wise CORAL strategy and class-wise CORAL strategy in GNN. Both the domain-level information and class-level information are considered.

3) Moving Semantic Transfer Network (MSTN) [41]: Both the MSTN and the proposed JCGNN consider the class-level information. However, MSTN aims to align exponential moving average centroids with the same label but different domains, while our proposed JCGNN adopts CORAL strategy and aims to align the per-class covariance matrix of features.

4) Correlation Alignment for Deep Domain Adaptation (D-CORAL) [18]: The D-CORAL embeds CORAL strategy in the deep architecture and aims to align the covariance matrix of features that extracted by DNN. The CORAL adopts the second-order statistics as in our approach. However, it calculates the covariance matrix by utilizing all the input data in a domain, without considering the distribution discrepancy between classes. Moreover, The D-CORAL ignores the interdependence among the spectrum during feature extraction. The proposed JCGNN applies GNN for feature extraction, and adopts CORAL strategy on both domain-level and class-level.

V. Experiments

A. Datasets Description

The performance of the proposed JCGNN is evaluated by comparing with several state-of-art domain adaptation methods on two real-world multitemporal hyperspectral remote sensing datasets, including Botswana (BOT) multitemporal Hyperion dataset and the Houston multitemporal dataset.

1) Botswana Multitemporal Hyperspectral Dataset: The BOT dataset consists of three multitemporal images that were acquired by NASA EO-1 Hyperion instrument in May, June and July 2001 over the Okavango Delta, Botswana. All images contain 242-band data at a 30-m spatial resolution and cover the 357–2576 nm spectral portion in 10 nm spectral resolution. Removal of uncalibrated and noisy bands leaves 145 bands for experiments. The three images include nine identified classes and any two of them can be selected as the source and target data, and thus six data pairs are available. The images and labeled information are shown in Fig. 3. The class names and the number of samples of the image are given in Table I.

2) Houston Multitemporal Hyperspectral Dataset: The Houston images were collected by NSF-funded Center for Airborne Laser Mapping (NCALM). The dataset consists of two multitemporal images that were acquired over the University of Houston campus and its neighboring area in 2012 and 2017, respectively. The 2012 Houston data has 144 bands but the 2017 Houston data only has 48 bands. The spectral range of both images is 380–1050 nm. Every three bands in 2012 Houston data were averaged as a new band [45], and then the number of the spectral band in the 2012 Houston data becomes 48, which is consistent with the dimensionality of the 2017 Houston data. We used the 2012 Houston data and 2017 Houston data with four classes as source and target data respectively. The images and labeled information are shown in Fig. 4. The class names and the number of samples of the image are given in Table I.

B. Setup

We compared the proposed method with several state-of-the-art deep learning domain adaptation methods, including DAN [40], DANN [20], MADA [23], MSTN [41], and D-CORAL [18]. In addition, the performance of the DNN, the GNN that were trained on source data without embedding domain adaptation strategy were also employed for comparison. JCGNN achieves distribution alignment by applying the joint CORAL to the GNN architecture. For a better understanding of the proposed JCGNN, we conducted experiments using DNN with joint CORAL (JCDNN), and GNN with only domain-wise CORAL (DCGNN).

The proposed JCGNN was implemented on pytorch framework and composed of two hidden layers (350 units for BOT and 128, 32 units for Houston dataset). The ReLU activation function was employed in the hidden layers and the softmax activation function is employed in the output layer. The dropout strategy was also used in each hidden layer to prevent overfitting. For a fair comparison, all the compared networks adopted the full-connected network with the same hidden units and activation function as JCGNN. GNN and DCGNN adopted the GNN with the same hidden units and activation function as JCGNN.

In the proposed JCGNN, we used Adam to optimize the network and the weight decay was set to be 5e-4. The learning rate was annealed by $\eta = \frac{\eta_0}{(1+\alpha^{\mathrm{i}})^{\mathrm{p}}}$, where η_0 was the initial learning rate and i was the training step updating from 0 to 1. Decay parameters α and p were set to be 10 and 0.75, respectively. The dropout rate was set to be 0.9. We set the initial learning rate as 0.0008 in BOT "May-June," "June-May," 0.005 in BOT

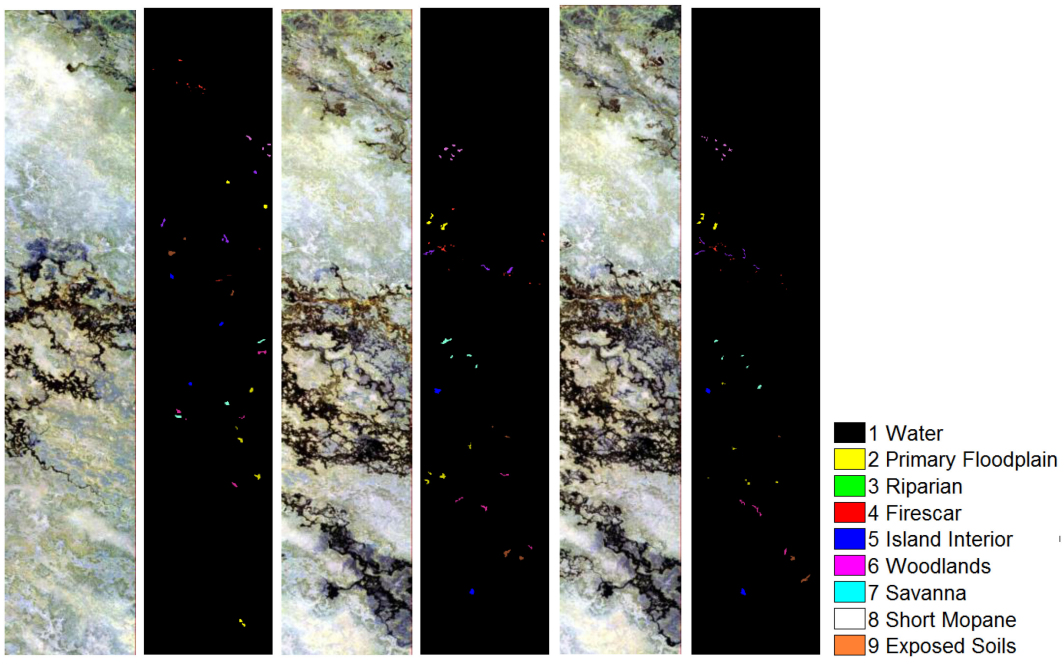

Fig. 3. BOT images in May, June, and July. (a) BOT image in May. (b) Ground truth of image in May. (c) BOT image in June. (d) Ground truth of image in June. (e) BOT image in July. (f) Ground truth of image in July. (g) Class legend.

TABLE I
CLASS NAME AND NUMBER OF SAMPLES OF BOT AND KSC IMAGES

BOT					HOUSTON DATASET			
ID	Class Name	May	June	July	ID	Class Name	2012	2017
1	Water	297	361	185	1	Healthy Grass	875	980
2	Primary Floodplain	437	308	96	2	Stressed Grass	906	3250
3	Riparian	448	303	164	3	Highway	710	986
4	Firescar	354	335	186	4	Railway	941	694
5	Island Interior	337	370	131				
6	Woodlands	357	324	169				
7	Savanna	330	342	171				
8	Short Mopane	239	299	152				
9	Exposed Soils	215	229	96				
Total		3014	2871	1350			3432	5910

"May-July," "July-May," "June-July," "July-June," and 0.001 in Houston dataset. We adopted a two-stage training procedure. In the first stage, the number of iterations was set to be 1000 for BOT data and 500 for Houston dataset. In the second stage, the number of iterations was set to be 4000 for BOT and 2000 for Houston dataset. It is worth noting that all the source and target data were directly forwarded into the network without mini-batch strategy in GNN, DCGNN and JCGNN. Besides, the number of adjacency nodes k was fixed as eight for all data pairs. More analysis about parameters would be provided in the sensitivity analysis section.

There are two parameters in the loss function of the proposed JCGNN approach, where λ_1 controls the weight of domain-wise CORAL loss and λ_2 controls the weight of class-wise CORAL loss. JCGNN also has the parameter σ of the Gaussian diffusion kernel. In the experiments, parameter λ_1 were tested with ten different values (0.1, 0.5, 1, 1.5, 2, 2.5, 3, 3.5, 4, 4.5, 5), parameter λ_2 were selected from (0.5, 1, 1.5, 2, 2.5, 3, 3.5, 4, 4.5, 5), and the parameter σ were chosen from (0.1, 0.5, 0.8, 1, 3, 5, 7, 9, 15, 20). We selected the best value of each parameter on the condition that the other parameters were fixed. The detailed parameter analysis is described in the sensitivity

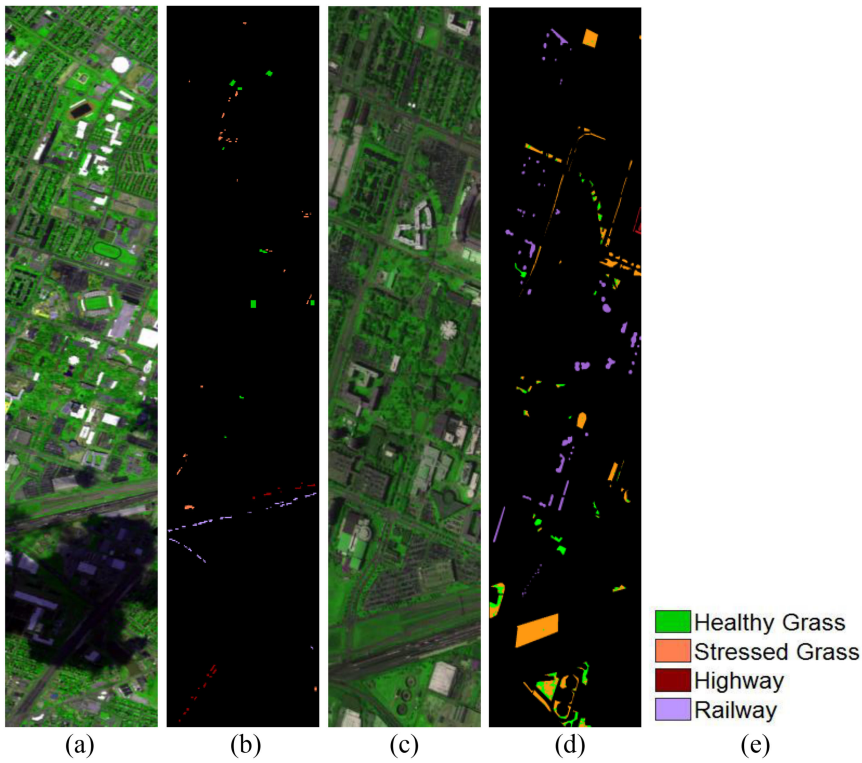

Fig. 4. Houston images. (a) 2012 Houston image. (b) Ground truth of 2012 Houston image. (c) 2017 Houston image. (d) Ground truth of 2017 Houston image. (g) Class legend.

TABLE II
OA% OF DIFFERENT DOMAIN ADAPTATION METHODS

Data Name	Data Pair		DNN	GNN	DAN	DANN	MADA	MSTN	D-CORAL	JCDNN	DCGNN	JCGNN
	Source image	Target image										
BOT	May	June	88.46	92.23	86.72	88.23	91.51	92.84	90.30	92.33	93.35	**94.11**
	June	May	81.30	83.44	86.28	78.15	82.12	85.60	87.13	90.10	89.87	**92.91**
	May	July	85.78	87.68	87.51	86.79	90.29	90.86	89.79	89.72	90.69	**91.11**
	July	May	71.18	72.43	78.47	68.69	73.10	82.84	76.08	79.29	80.40	**83.54**
	June	July	94.26	94.73	91.16	94.47	94.76	94.27	94.31	94.23	95.05	**95.85**
	July	June	90.47	91.37	90.19	89.40	91.54	92.83	91.71	92.49	93.63	**94.67**
DFC	2012	2017	73.91	74.77	75.45	77.39	78.05	75.82	78.58	78.88	81.25	**82.89**
	2017	2012	90.12	90.43	89.51	89.66	91.33	91.58	90.88	91.92	92.33	**92.89**

analysis section. For the comparative methods, DAN, DANN, MADA, MSTN, D-CORAL, JCDNN, and DCGNN have the weight parameter for the domain adaptation loss. The weight of domain-wise CORAL loss and class-wise CORAL loss in D-CORAL, JCDNN, and DCGNN adopted the same parameter as in the proposed JCGNN for a fair comparison, and other parameters were chosen as the recommended values in [20], [23], [40], and [41].

Before training the JCGNN, each spectral band of both source and target data was normalized to have a standard normal distribution $N(01)$. Such data preprocessing method was also applied to all the compared methods. Since the initial network parameters were chosen randomly, all experiments were conducted ten times and the average classification was used for evaluation.

C. Results of JCGNN

The overall accuracy (OA) and kappa coefficients of all the compared methods and proposed JCGNN are given in Tables II and III. The DNN without any adaptation strategy can achieve satisfactory performances on these data pairs which have a small spectral drift, such as BOT "June-July" and "July-June." On the contrary, if the data pairs have a big spectral drift, such as

TABLE III
KAPPA COEFFICIENT OF DIFFERENT DOMAIN ADAPTATION METHODS

Data Name	Data Pair		DNN	GNN	DAN	DANN	MADA	MSTN	D-CORAL	JCDNN	DCGNN	JCGNN
	Source image	Target image										
BOT	May	June	0.869	0.912	0.850	0.869	0.904	0.920	0.891	0.912	0.925	**0.934**
	June	May	0.789	0.813	0.845	0.748	0.798	0.838	0.855	0.888	0.885	**0.919**
	May	July	0.839	0.860	0.859	0.850	0.890	0.897	0.884	0.884	0.895	**0.899**
	July	May	0.675	0.689	0.758	0.647	0.696	0.800	0.730	0.764	0.779	**0.814**
	June	July	0.935	0.940	0.900	0.937	0.941	0.940	0.936	0.935	0.944	**0.953**
	July	June	0.873	0.903	0.889	0.881	0.908	0.914	0.907	0.915	0.928	**0.939**
DFC	2012	2017	0.638	0.647	0.659	0.679	0.684	0.662	0.690	0.697	0.723	**0.743**
	2017	2012	0.867	0.871	0.858	0.860	0.876	0.883	0.871	0.893	0.897	**0.904**

BOT "July-May", Houston dataset "2012–2017," the accuracies of DNN are low. Almost all the domain adaptation methods obtain higher accuracies than DNN, demonstrating their positive transfer learning ability. The GNN outperforms DNN on almost all the data pairs, which demonstrates that GNN could achieve more effective feature extraction ability by utilizing spectral neighborhood information. The D-CORAL can obtain higher accuracies than DAN, indicating the advantage of CORAL strategy compared to MMD. The MSTN and MADA performed better than DAN, DANN and D-CORAL, which demonstrates that the exploration of class-level knowledge would facilitate producing superior alignment performance. Compared to all the compared methods, JCGNN yields the best performances on all the data pairs. The JCGNN performed better than MSTN and MADA, which demonstrates the advantage of embedding class-wise CORAL strategy and domain-wise CORAL in GNN architecture. It is worth noting that the JCGNN improves the classification performance significantly on the "difficult" data pairs like BOT "July-May" and Houston dataset "2012–2017," and it also achieves comparable performance on the "easy" data pairs. Moreover, the accuracies of the JCGNN obtained almost 5%–10% improvement with respect to all the compared methods on BOT dataset. These observations demonstrate that the GNN, domain-wise CORAL and class-wise CORAL can cooperate well for domain adaptation of remote sensing images.

The JCGNN achieves distribution alignment by applying the domain-wise CORAL loss and class-wise CORAL loss to the GNN architecture, where domain-wise CORAL loss aims to reduce the global distribution discrepancy across domains and class-wise CORAL loss aims to align the distribution on a per-class basis. The GNN architecture is adopted rather than the DNN because the GNN makes use of spectral neighborhood information for better classification performance. For a better understanding of the proposed JCGNN, we also compared the results of JCGNN with JCDNN and DCGNN.

As given in Tables II and III, JCGNN performed better than JCDNN on almost all data pairs, which demonstrates GNN has a better performance than DNN in domain adaptation task. JCGNN outperformed DCGNN, indicating the necessity of distribution alignment on a class-level basis.

D. Analysis of JCGNN

The performances of the five algorithms were also reported for comparison on a per-class basis. As given in Table IV about the results of BOT "June-May," the JCGNN obtained the highest accuracies for almost all the classes. For difficult classes such as class 2 (DNN: 55.47%, GNN: 56.34%, and DCGNN: 81.8%), JCGNN achieved remarkable performance (JCGNN:96.25%). Moreover, JCGNN also achieved comparable performance on these classes which are easy to be well classified, such as class 1 (DNN: 99.76%, JCGNN: 100.00%) and class 8 (DNN:97.28%, JCGNN:99.16%).

E. Alignment Performance of JCGNN

To illustrate the effectiveness of the proposed JCGNN, we employed t-SNE embeddings [42] to set the high-dimensional features as two-dimensional (2-D) for visualization. Fig. 5, shows classes in "May-June" "June-May," and "July-June" data pair of the BOT dataset in different colors, and represents scatters from source and target domain by hollow circles and pentagrams, respectively. Compared with the visualizations obtained by DNN without any adaptation in Fig. 5(a), (d), and (h), the visualizations obtained by GNN without any adaptation in Fig. 5(b), (e), and (i) become more compact and better separated. For instance, the features of class 35,9 from "July-June" are gravely overlapped in Fig. 5(h), but they are tightly clustered in Fig. 5(i). This observation demonstrates that the GNN outperforms DNN in classification task. However, GNN is incapable of reducing the domain shift and conducting domain adaptation. Therefore, it is necessary to embed alignment strategy in the architecture of GNN to conduce domain-invariance features. Fig. 5(c), (f), and (j) illustrates the performance of the JCGNN, where features from different domains are aligned well after adopting domain-wise CORAL loss and class-wise CORAL loss in GNN architecture.

To further illustrate the alignment performance of each class to verify the effect of domain-wise CORAL and class-wise CORAL. Fig. 6 plots the features from second and ninth output units of GNN for "June-May" in BOT datasets, where the red scatters represent one class data from source domain and blue

TABLE IV
CLASSIFICATION ACCURACY OF EACH CLASS OF BOT "JUNE-MAY" DATA PAIR

ID	Class	DNN	GNN	JCDNN	DCGNN	JCGNN
1	Water	99.76	99.12	100.00	100.00	**100.00**
2	Primary Floodplain	55.47	56.34	95.42	81.08	**96.25**
3	Riparian	73.06	84.96	85.49	90.08	**91.03**
4	Firescar	95.73	97.18	97.01	95.93	**97.46**
5	Island Interior	84.30	83.65	98.74	96.49	**99.08**
6	Woodlands	83.50	**84.99**	68.62	74.03	73.73
7	Savanna	73.48	71.69	80.91	85.91	**92.94**
8	Short Mopane	97.28	98.79	98.74	98.70	**99.16**
9	Exposed Soils	87.53	89.12	89.37	89.30	**90.00**
	OA%	81.30	83.44	90.14	89.88	**92.91**
	AA%	83.35	85.09	90.56	90.48	**93.09**
	Kappa	0.789	0.813	0.889	0.885	**0.919**

scatters represent the data from same class of target domain. Classes 12 and 8 are selected for illustration. From Fig. 6(a), (d), and (h), it can be seen that the distributions between source and target domain features are different due to spectral drift. Fig. 6(b), (e), and (i) show the effectivity of domain-wise CORAL, where the features of two domains are closer than the previous features after introducing the domain-wise CORAL loss. The visualization of JCGNN were shown in Fig. 6(c), (f), and (j). With the refined adaptation of class-wise CORAL, the covariance of same class from different domains have been well aligned. Class-wise CORAL can achieve more effective alignment performance than domain-wise CORAL due to considering the class-level information.

F. Classification Results of the Whole Image by the JCGNN

To classify the whole target image, we firstly divided all the target data into subsets, where each subset is with 5000 pixels. Then each subset and its graph adjacency matrix are fed into the network. By combining the classification results of all the subsets, the classification map of the whole target image can be obtained. Fig. 7 shows the classification results of the BOT images, where "June-May" and "July-June" data pairs are chosen for demonstration. Fig. 7(a) and (d) shows the DNN classification results without any adaptation for "June-May" and "July-June," respectively. Fig. 7(b) and (e) denotes the classification results of proposed JCGNN. Due to lack of ground truth on the whole images, the classification results of classifier that trained on the target labeled data are used as "reference," and the whole images of "reference" are shown in Fig. 7(c) and (f). In Fig. 7(a) and (c), we could observe that the results of DNN without any adaptation is significantly different from the results of "reference." After adopting the adaptation strategy, the results of JCGNN are highly similar to "reference," which could demonstrate the advantage of our proposed domain adaptation approach.

To better demonstrate the effect of our proposed methods, we chose two local regions for further analysis. All local regions are shown in the pink windows in Fig. 7 and enlarged in Fig. 8. The wetland and upland define the two major ecosystems included in BOT dataset [43]. For BOT "June-May" data pair, a wetland area is selected. The selected wetland local region mainly contains class 1 (water, black), class 2 (primary floodplain, yellow), and class 5 (island interior, dark blue). As shown in Fig. 8(a)–(c), we can clearly see that class 2 (primary floodplain, yellow) and class 5 (island interior, dark blue) are easily misclassified as class 6 (woodlands, purple) in the results of DNN without any adaptation. Fortunately, our proposed JCGNN could effectively prevent the occurrence of misclassification, so that the results of JCGNN is much similar to "reference". For BOT "July-June" data pair, an upland area is selected, which mainly contains class 6(woodlands, purple), class 7 (savanna, light blue) and class 9 (exposed soils, orange). As shown in Fig. 8(d)–(f), the results of DNN without any adaptation assigned many false predictions, and many pixels of class 3(Riparian, light green) were misclassified as class 2 (primary Floodplain, yellow) and class 3(riparian, light green). After using the JCGNN, most pixels belonging to class 2 (primary Floodplain, yellow) and class 3(riparian, light green) are correctly classified.

G. Parameter Sensitivity

We conducted sensitivity analysis for the three parameters in the JCGNN: λ_1 , λ_2, σ. The parameters λ_1 and λ_2 are the trade-off parameters where λ_1 controls the weight of domain-wise CORAL loss and λ_2 controls the weight of class-wise CORAL loss. The parameter σ is the Gaussian diffusion kernel. We used six BOT data pairs to show the results, and similar trends could also be obtained on the other data pairs. To find a best domain adaptation performance, each parameter was adjusted on the condition that the other parameters are fixed. For parameter λ_1, ten different values (0.1, 0.5, 1, 1.5, 2, 2.5, 3, 3.5, 4, 4.5, 5) were tested. The optimal parameter values of λ_2 and σ were fixed as (1, 3, 1, 0.5, 1, 1) and (1, 1, 0.7, 1, 0.5, 1) for the six data pairs, respectively. The classification results were shown in Fig. 9(a). We can observe that the proposed approach was not sensitive to this parameter in this range, suggesting that this parameter should be set in this range. Fig. 9(b) shows the classification

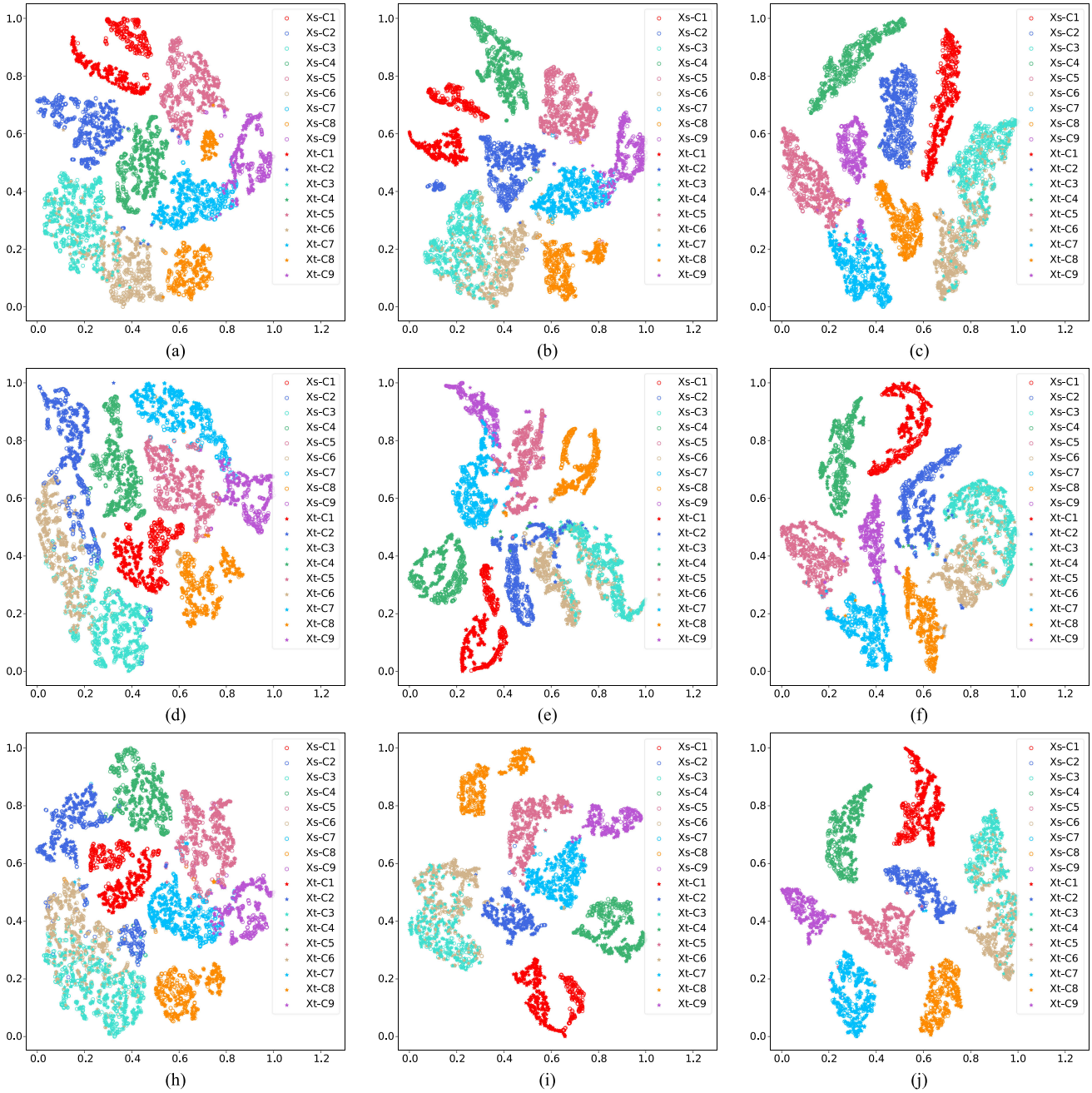

Fig. 5. The t-SNE visualization of BOT dataset. (a) DNN for "May-June" data pair (b) GNN for "May-June" data pair. (c) JCGNN for "May-June" data pair. (d) DNN for "June-May" data pair (e) GNN for "June-May" data pair. (f) JCGNN for "June-May" data pair. (h) DNN for "July-June" data pair (i) GNN for "July-June" data pair. (j) JCGNN for "July-June" data pair.

results with parameter λ_2 from (0.5, 1, 1.5, 2, 2.5, 3, 3.5, 4, 4.5, 5). The optimal parameter values of λ_1 and σ were fixed as (1.5, 1.5, 1, 0.5, 1.5, 1) and (1, 1, 0.7, 1, 0.5, 1) for the six data pairs, respectively. For BOT "May-June," "June-May," "May-July," "June-July," "July-June," different values of this parameter yield similar classification performance. In addition, we can observe that almost all data pairs can achieve satisfactory performance when λ_2 was equal to 1. We suggested that the value of λ_2 should be set around 1. In Fig. 9(c), we tested different values of parameter σ from (0.1, 0.5, 0.8, 1, 3, 5, 7, 9, 15, 20), with the parameter values of λ_1 and λ_2 being fixed as (1.5, 1.5, 1, 0.5, 1.5, 1) and (1, 3, 1, 0.5, 1, 1) for the six data pairs, respectively. When the value of σ increases from 0 to 1, the accuracies on these "easy" data pairs increase and reach the best at $\sigma = 1$. However, when the σ was too large, the accuracy would drop significantly. For these "difficult" data pairs, such as BOT "July-May," the accuracy would grow with an increasing value of σ. We suggested that the parameter σ should not more than 1 when data pairs with little spectral drifts. When there was large spectral drift across domains, the parameter σ should be greater than 10.

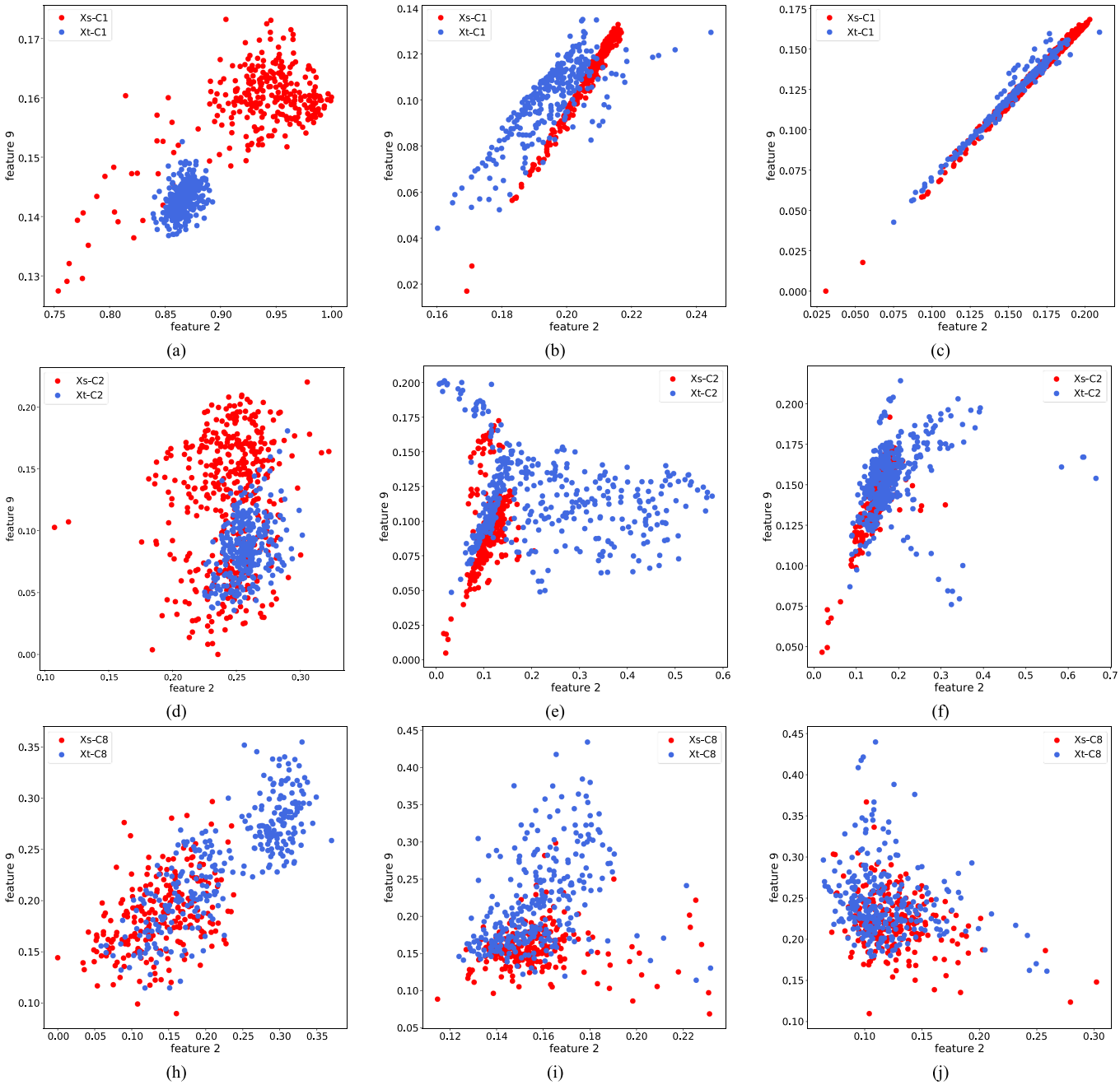

Fig. 6. Alignment performance for BOT "June-May" data pair (a) Class 1 of DNN results. (b) Class 1 of GNN with only domain-wise CORAL. (c) Class 1 of JCGNN (d) Class 2 of DNN results. (e) Class 2 of GNN with only domain-wise CORAL. (f) Class 2 of JCGNN (h) Class 5 of DNN results. (i) Class 5 of GNN with only domain-wise CORAL. (j) Class 5 of JCGNN.

H. Computational Time

The computational time of all the compared domain adaptation approaches was given in Table V. All the experiments were implemented with the deep learning framework and were executed on NVIDIA GeForce RTX 2080 GPU with 16-GB memory. The GPU was used to accelerate the training process. As given in Table V, the training time of GNN is longer than DNN because the training process of GNN did not adapt mini-batch strategy. The MSTN and MADA were slower than DAN, DANN D-CORAL, and DCGNN, which indicated that achieve class-level distribution alignment was more time-consuming than achieve domain-level distribution alignment. The JCGNN that combined both class-level and domain-level distribution alignment costed more time than most compared domain adaptation approach, but the computational time was still within acceptable limits.

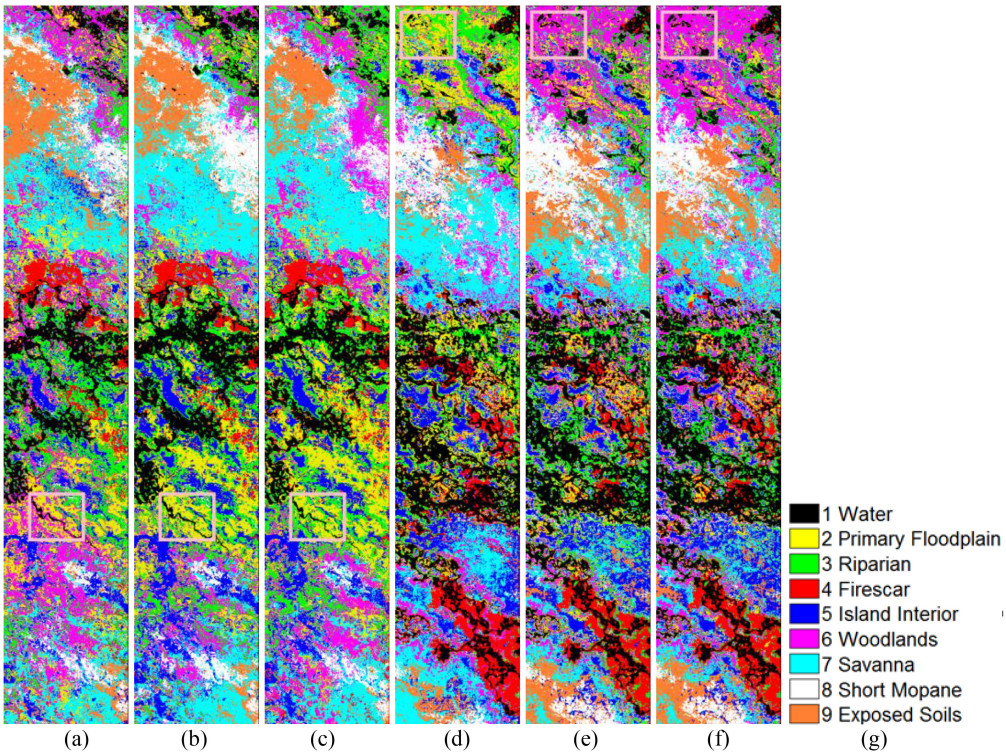

Fig. 7. Classification results of the target image in BOT "June-May" and "June-July" data pairs. (a) DNN result without any adaptation for "June-May" data pair (b)JCGNN result for "June-May" data pair. (c) Reference obtained by using target labeled data as training data for "June-May" data pair. (d) DNN result without any adaptation for "July-June" data pair. (e) JCGNN result for "July-June" data pair. (f) Reference obtained by using target labeled data as training data for "July-June" data pair. (g) Class legend.

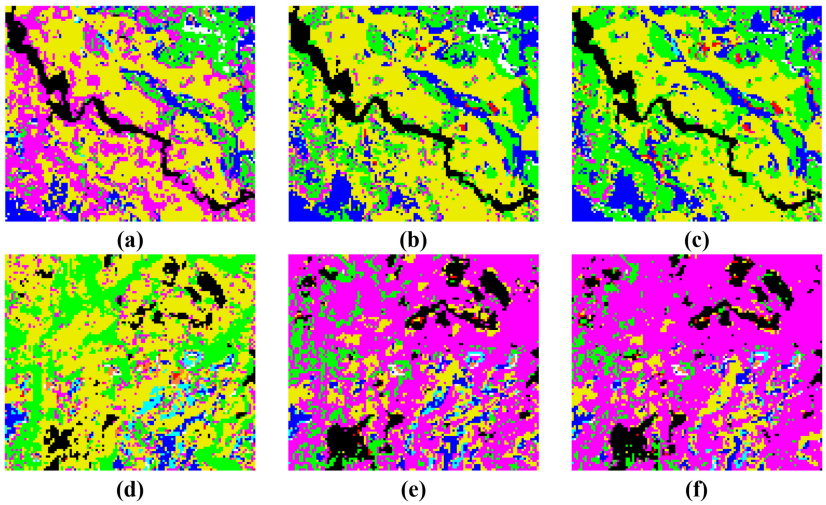

Fig. 8. Classification results of local regions. (a) DNN result without any adaptation for "June-May" wetland area. (b) JCGNN result for "June-May" wetland area. (c) Reference obtained by using target labeled data as training data for "June-May" wetland area. (d) DNN result without any adaptation for "July-June" upland area. (e) JCGNN result for "July-June" upland area. (f) Reference obtained by using target labeled data as training data for "July-June" upland area.

TABLE V
COMPUTATIONAL TIME(S) OF DIFFERENT METHODS

Data sets	DNN	GNN	DAN	DANN	MADA	MSTN	D-CORAL	DCGNN	JCDNN	JCGNN
BOT	62.86	132.98	94.82	108.24	240.63	813.76	105.70	244.14	374.06	535.34
DFC	45.13	105.35	75.27	81.69	159.87	581.19	69.49	183.62	217.53	336.13

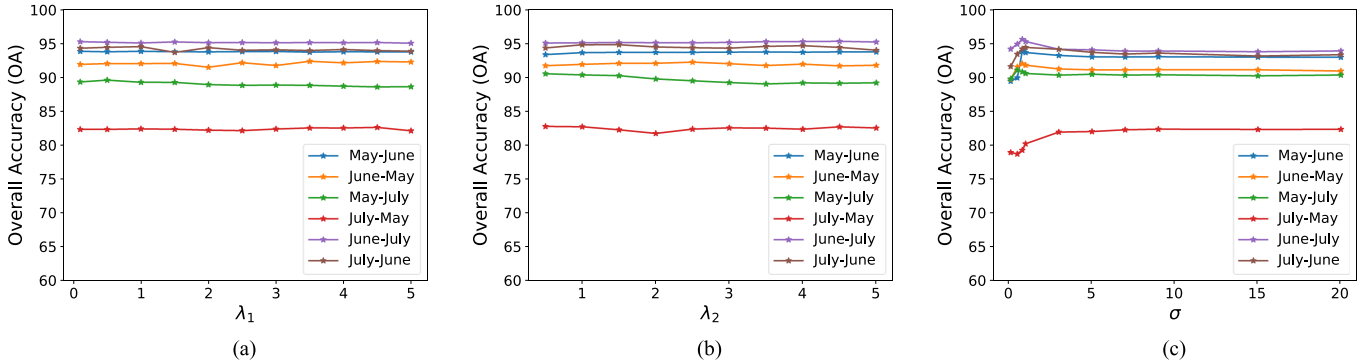

Fig. 9. Sensitivity analysis of parameters in our adaptation using BOT data pairs. (a) Parameter λ_1. (b) parameter λ_2, and (c) Parameter σ.

VI. Conclusion

In this article, we proposed a new unsupervised domain adaptation method for multitemporal remote sensing images, which adopted GNN for feature extraction to utilize spectral neighborhood information. Features from two domains are then aligned by jointly exploiting CORAL on both domain-level and class-level. Experiments on two multitemporal hyperspectral remote sensing images datasets demonstrated that the proposed JCGNN method offers the classification performance superior to other state-of-the-art deep domain adaptation methods for multitemporal images, because GNN can be used to extract representative features from spectral neighbors in multitemporal images. Moreover, the JCGNN outperformed DCGNN, indicating the advantage of the class-level distribution alignment. Since the construction of adjacent matrix plays an important part in GNN [31] and [44], we will consider to update the adjacent matrix in the feature extracting process to learn more domain-invariant features in our future work.

Acknowledgment

The authors would like to thank Professor Melba Crawford at Purdue University for providing the BOT data and thank the Hyperspectral Image Analysis Group and the National Center for Airborne Laser Mapping, University of Houston, Houston, TX, USA, for providing the Houston data used in this study.

References

[1] C. M. Bachmann, T. L. Ainsworth, and R. A. Fusina, "Exploiting manifold geometry in hyperspectral imagery," *IEEE Trans. Geosci. Remote Sens.*, vol. 43, no. 3, pp. 441–454, Mar. 2005.

[2] J. Peng, W. Sun, and Q. Du, "Self-paced joint sparse representation for the classification of hyperspectral images," *IEEE Trans. Geosci. Remote Sens.*, vol. 57, no. 2, pp. 1183–1194, Feb. 2019.

[3] C. Persello and L. Bruzzone, "Kernel-based domain-invariant feature selection in hyperspectral images for transfer learning," *IEEE Trans. Geosci. Remote Sens.*, vol. 54, no. 5, pp. 2615–2626, May 2016.

[4] W. Kim and M. M. Crawford, "Adaptive classification for hyperspectral image data using manifold regularization kernel machines." *IEEE Trans. Geosci. Remote Sens.*, vol. 48, no. 11, pp. 4110–4121, Nov. 2010.

[5] D. Tuia, C. Persello, and L. Bruzzone, "Domain adaptation for the classification of remote sensing data: An overview of recent advances," *IEEE Geosci. Remote Sens. Mag.*, vol. 4, no. 2, pp. 41–57, Jun. 2016.

[6] E. Zhong, K. Zhang, J. Ren, D. Turaga, and O. Verscheure, "Cross domain distribution adaptation via kernel mapping," in *Proc. ACM Sigkdd Int. Conf. Knowl. Discovery Data Mining*, 2009, pp. 1027–1036.

[7] M. Long, J. Wang, G. Ding, J. Sun, and P. S. Yu, "Transfer joint matching for unsupervised domain adaptation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2014, pp. 1410–1417.

[8] B. Gong, K. Grauman, and F. Sha, "Connecting the dots with landmarks: Discriminatively learning domain-invariant features for unsupervised domain adaptation," in *Proc. Int. Conf. Int. Conf. Mach. Learn.*, 2013, pp. 16–21.

[9] L. Bruzzone and D. F. Prieto, "Unsupervised retraining of a maximum likelihood classifier for the analysis of multitemporal remote sensing images," *IEEE Trans. Geosci. Remote Sens.,* vol. 39, no. 2, pp. 456–460, Feb. 2001.

[10] L. Bruzzone and M. Marconcini, "Domain adaptation problems: A DASVM classification technique and a circular validation strategy," *IEEE Trans. Pattern Anal. Mach. Intelligencel*, vol. 32, no. 5, pp. 770–787, May 2010.

[11] K. Bahirat, F. Bovolo, L. Bruzzone, and S. Chaudhuri, "A novel domain adaptation Bayesian classifier for updating land-cover maps with class differences in source and target domains," *IEEE Trans. Geosci. Remote Sens.,* vol. 50, no. 7, pp. 2810–2826, Jul. 2012.

[12] S. Rajan, J. Ghosh, and M. M. Crawford, "Exploiting class hierarchies for knowledge transfer in hyperspectral data," *IEEE Trans. Geosci. Remote Sens.*, vol. 44, no. 11, pp. 3408–3417, Nov. 2006.

[13] W. Kim and M. M. Crawford, "Adaptive classification for hyperspectral image data using manifold regularization kernel machines," *IEEE Trans. Geosci. Remote Sens.,* vol. 48, no. 11, pp. 4110–4121, Nov. 2010.

[14] L. Zhu and L. Ma, "Class centroid alignment based domain adaptation for classification of remote sensing images," *Pattern Recognit. Lett*., vol. 83, no. 2, pp. 124–132, Jan.2016.

[15] L. Ma, M. M. Crawford, L. Zhu, and Y. Liu, "Centroid and covariance alignment-based domain adaptation for unsupervised classification of remote sensing images," *IEEE Trans. Geosci. Remote Sens.*, vol. 57, no. 4, pp. 2305–2323, Apr. 2019.

[16] L. Song *et al.* "Unsupervised domain adaptive re-identification: Theory and practice," 2018. [Online]. Available: *arxiv.org/abs/1807.11334v1*.

[17] E. Tzeng, J. Hoffman, N. Zhang, K. Saenko, and T. Darrel, "Deep domain confusion: Maximizing for domain invariance," 2014, *arXiv:1412.3474*.

[18] B. Sun and K. Saenko, "Deep CORAL: Correlation alignment for deep domain adaptation," 2016, *arXiv:1607.01719v1*.

[19] I. Goodfellow *et al.*, "Generative adversarial nets," 2014, *arXiv:1406.2661*.

[20] Y. Ganin, E. Ustinova, H. Ajakan, P. Germain, H. Larochelle, and F. Laviolette. "Domain-adversarial training of neural networks," *J. Mach. Learn. Res.*, vol. 17, no. 59, pp. 1–35, May. 2016.

[21] Z. Wang, B. Du, Q. Shi, and W. Tu, "Domain adaptation with discriminative distribution and manifold embedding for hyperspectral image classification," *IEEE Geosci. Remote Sens. Lett.*, vol. 16, no. 7, pp. 1155–1159, Jul. 2019.

[22] M. Chen, L. Ma, W. J. Wang, and Q. Du, "Augmented associative learning-based domain adaptation for classification of hyperspectral remote sensing images," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 13, pp. 6236–6248, Oct, 2020.

[23] Z. Liu, L. Ma, and Q. Du, "Classwise distribution adaptation for unsupervised classification of hyperspectral remote sensing images," *IEEE Trans. Geosci. Remote Sens.*, vol. 59, no. 1, pp. 508–521, Jan. 2021.
[24] W. Gross, D. Tuia, U. Soergel, and W. Middelmann, "Nonlinear feature normalization for hyperspectral domain adaptation and mitigation of nonlinear effects," *IEEE Trans. Geosci. Remote Sens.*, vol. 57, no. 8, pp. 5975–5990, Aug. 2019.
[25] S. Saha, F. Bovolo, and L. Bruzzone, "Unsupervised multiple-change detection in VHR multisensor images via deep-learning based adaptation," in *Proc. IEEE Int. Geosci. Remote Sens. Symp.*, Jul. 2019, pp. 5033–5036.
[26] W. Li, W. Wei, L. Zhang, C. Wang, and Y. Zhang, "Unsupervised deep domain adaptation for hyperspectral image classification," in *Proc. IEEE Int. Geosci. Remote Sens. Symp.*, Jul. 2019, pp. 1–4.
[27] S. Wan, C. Gong, P. Zhong, B. Du, L. Zhang, and J. Yang, "Multiscale dynamic graph convolutional network for hyperspectral image classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 58, no. 5, pp. 3162–3177, May, 2020.
[28] D. Hong, L. Gao, J. Yao, A. Plaza, and J. Chanussot, "Graph convolutional networks for hyperspectral image classification," *IEEE Trans. Geosci. Remote Sens.*, to be published, doi: 10.1109/TGRS.2020.3015157.
[29] L. Mou, X. Lu, X. Li, and X. Zhu, "Nonlocal graph convolutional networks for hyperspectral image classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 58, no. 12, pp. 8246–8257, Dec. 2020.
[30] M. Gori, G. Monfardini, and F. Scarselli, "A new model for learning in graph domains," in *Proc. Int. Joint Conf. Neural Netw.*, 2005, pp. 729–734.
[31] J. Bruna, W. Zaremba, A. Szlam, and Y. LeCun, "Spectral networks and deep locally connected networks on graphs," 2014, *arXiv:1312.6203*.
[32] M. Defferrard, X. Bresson, and P. Vandergheynst, "Convolutional neural networks on graphs with fast localized spectral filtering," 2017, *arXiv:1606.09375v2*.
[33] T. Kipf and M. Welling, "Semi-supervised classification with graph convolutional networks," in *Proc. Int. Conf. Learn. Representations*, 2017, pp. 1–14.
[34] 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.
[35] X. Tong, J. Yin, B. Han, and H. Qv, "Few-shot learning with attention-weighted graph convolutional networks for hyperspectral image classification," in *Proc. IEEE Int. Conf. Image Process.*, 2020, pp. 1686–1690.
[36] S. Wan, C. Gong, P. Zhong, S. Pan, G. Li, and J. Yang, "Hyperspectral image classification with context-aware dynamic graph convolutional network," *IEEE Trans. Geosci. Remote Sens.*, vol. 59, no. 1, pp. 597–612, May 2020, doi: 10.1109/TGRS.2020.2994205.
[37] Z. Wu, S. Pan, F. Chen, G. Long, C. Zhang, and P. S. Yu, "A comprehensive survey on graph neural networks," 2019, *arXiv:1901.00596v2*.
[38] M. Henaff, J. Bruna, and Y. LeCun, "Deep convolutional networks on graph-structured data," 2015, *arXiv:1506.05163v1*.
[39] M. Belkin and P. Niyogi, "Laplacian eigenmaps and spectral techniques for embedding and clustering," in *Proc. 14th Int. Conf. Neural Inf. Process. Syst. Natural and Synthetic*, 2001, vol. 14, pp. 585–591.
[40] M. Long, Y. Cao, J. Wang, and M. I. Jordan, "Learning transferable features with deep adaptation networks," 2015, *arXiv:1502.02791*.
[41] S. Xie, Z. Zheng, L. Chen, and C. Chen, "Learning semantic representations for unsupervised domain adaptation," in *Proc. Int. Conf. Mach. Learn.*, 2018, pp. 1–10.
[42] J. Donahue *et al.*, "DeCAF: A deep convolutional activation feature for generic visual recognition," 2013, *arXiv:1310.1531*.
[43] A. L. Neuenschwander, "Remote sensing of vegetation dynamics in response to flooding and fire in the okavango delta, Botswana," Ph.D. dissertation, Dept. Aerosp. Eng., Univ. Texas, Austin, Austin, TX, USA, 2007.
[44] B. Jiang, Z. Zhang, D. Lin, and J. Tang, "Graph-learning-convolutional networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2018, pp. 1–10.
[45] C. Ge, Q. Du, Y. Li, and J. Li, "Multitemporal hyperspectral image classification using collaborative representation-based classification with tikhonov regularization," in *Proc. 10th Int. Workshop Anal. Multitemporal Remote Sens. Images*, Oct., 2019, pp. 1–4.

Wenjin Wang received the B.S. degree from the School of Communication Engineering in 2018, China University of Geosciences, Wuhan, China, where she is currently working toward the M.S. degree at the School of Mechanical Engineering and Electronic Information.

Her research interests include pattern recognition, computer vision and hyperspectral data analysis.

Li Ma (Member, IEEE) received the B.S. and M.S. degrees from Shandong University, Jinan, China, in 2004 and 2006, respectively, and the Ph.D. degree in pattern recognition and intelligent system from Huazhong University of Science and Technology, Wuhan, China, in 2011.

During 2008–2010, she was a Visiting Scholar with Purdue University, West Lafayette, IN, USA. She also visited Mississippi State University, for five months in 2018. She is currently an Associate Professor with School of Mechanical Engineering and Electronic Information, China University of Geosciences, Wuhan, China. Her research interests include hyperspectral data analysis, pattern recognition, and remote sensing applications.

Min Chen (Student Member, IEEE) received the B.S. degree from the School of Information Engineering, Wuhan University of Technology, Wuhan, China, in 2018. He is currently working toward the M.S degree at the School of Mechanical Engineering and Electronic Information from China University of Geosciences, Wuhan, China.

His research interests include pattern recognition, computer vision and hyperspectral data analysis.

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 the Bobby Shackouls Professor with the Department of Electrical and Computer Engineering, Mississippi State University, Starkville, MS, USA, and also an Adjunct Professor with the College of Surveying and Geo-informatics, Tongji University, Shanghai, China. 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 SPIC International Society for Optics and Photonics. She was a recipient of the 2010 Best Reviewer Award from the IEEE Geoscience and Remote Sensing Society (GRSS). She was the Chair of the Remote Sensing and Mapping Technical Committee of the International Association for Pattern Recognition from 2010 to 2014. She served as the Co-Chair for the Data Fusion Technical Committee of the IEEE GRSS from 2009 to 2013. She was the General Chair for the fourth IEEE GRSS Workshop on Hyperspectral Image and Signal Processing: Evolution in Remote Sensing held at Shanghai, in 2012. She was an Associate Editor for the IEEE JOURNAL OF TOPICS IN APPLIED EARTH OBSERVATIONS AND REMOTE SENSING (JSTARS), *the Journal of Applied Remote Sensing,* and THE IEEE SIGNAL PROCESSING LETTERS. She has served as the Editor-in-Chief of the IEEE JSTARS (2016–2020). She also organized several international workshops and journal special issues on remote sensing image processing and analysis.