Dynamic Spectral-Spatial Poisson Learning for Hyperspectral Image Classification with Extremely Scarce Labels

Shengwei Zhong, *Member, IEEE,* Tao Zhou, Sheng Wan, Jian Yang, *Member, IEEE,* and Chen Gong, *Member, IEEE,*

***Abstract*—Acquiring labeled training examples for hyperspectral images (HSI) is an expensive task, and even labeling one more pixel requires a real-time field survey of tens of square meters. Therefore, it is highly demanded to achieve satisfactory accuracy for an HSI classification method when the number of labeled examples is extremely limited. However, most of existing methods lack the ability to handle extremely sparse labeled data. To overcome this issue, we propose a novel graph-based framework for HSI classification, termed "Dynamic Spectral-Spatial Poisson Learning" (DSSPL). Specifically, three measures are utilized to enable the proposed model suitable for the situation of extremely limited labeled data. First, Poisson Learning (PL) is adopted for predicting labels on a graph, as it can prevent undesirable constant output labels of traditional label propagation methods and generate more informative label determinations. Second, spectral and spatial graphs are constructed from various features and fused to build a spectral-spatial graph, which exploits comprehensive connective relationships among pixels. Third, in each iteration, the fused graph is dynamically updated by feeding back the up-to-date label information generated by each iteration. The feedback strategy progressively refines the fused graph, and the propagation on updated graph in turn improves output labels iteratively. Intensive experimental results on three public datasets demonstrate that the proposed DSSPL significantly outperforms other state-of-the-art HSI classification methods when very few pixels (*e.g.*, 3, 5, or 10 of each class) are labeled.**

***Index Terms*—Hyperspectral image (HSI) classification, Semi-supervised learning, Poisson Learning, Graph updating.**

I. Introduction

HYPERSPECTRAL image (HSI) classification aims to classify every pixel (*i.e.*, example) in a given HSI into some pre-defined land-cover types (*i.e.*, classes) such as grassland, road, water, etc. To conduct HSI classification, one has to manually label part of examples for training classifiers, which requires expensive real-time survey and image interpretation by professionals. Up to now, lots of existing HSI classification methods can achieve more than 99% overall accuracy when there are sufficient labeled pixels [1]–[5].

Nevertheless, the performance of most methods will be largely degraded when the number of labeled pixels decreases. For example, deep neural networks have the ability to learn more complex models than shallow ones [6]. Several deep learning-based methods have been widely developed for HSI classification. However, when there are only limited labeled pixels, the deep learning-based models could easily be trained to zero training error by simply "memorizing" the training set without any generalization ability. This phenomenon has been theoretically studied in [7] and empirically verified in [8]. Therefore, it is important to develop classification methods with satisfactory performance when there are only limited labeled examples. Many methods have been proposed to address the abovementioned over-fitting problem. For instance, some methods improve the generalization ability via using different data augmentation strategies, including image patch transformation [1], adding regularization operators [9], and extending the pixel neighborhood information [10], [11]. In addition, some methods conduct the dimensionality reduction or feature extraction in the pre-processing phase to improve the classification accuracy [12]–[14]. However, as discussed in [10], these methods largely increase the computational complexity and still obtain unsatisfactory performance when the label rate is very low.

Except for the above supervised methods, some semi-supervised methods have also been proposed to utilize only a handful of labeled examples (usually $\sim$30 in each class) along with the remaining massive unlabeled examples for HSI classification [15]–[17]. In addition, few-shot learning [18] is adopted to construct a global representation from the labeled examples in base classes to aid the learning of scarce labeled examples in new classes. However, in some cases, due to the limitations in imaging such as bad atmospheric conditions and geographical factors (*e.g.*, border or other inaccessible regions), it is still difficult to collect $\sim$30 labeled examples for each class. In such extreme conditions, as reported in our experiments, a conventional semi-supervised method [19] could not achieve satisfactory classification performance due to the very limited supervisory information. Therefore, designing a new algorithm that still works with extremely scarce labeled pixels (*e.g.*, $3 \sim 10$ in each class) is highly demanded.

Manuscript received May 17, 2021; revised August 24, 2021; accepted October 31, 2021. This work was supported in part by the National Science Foundation (NSF) of China under Grant 62101261, Grant 61973162, Grant 62172228 and Grant U1713208; in part by the NSF of Jiangsu Province under Grant BK20210332 and BZ2021013; in part by the China Postdoctoral Science Foundation under Grant 2020M681605; and in part by the Fundamental Research Funds for the Central Universities under Grant 30920032202 and 30921013114. (*Corresponding authors: Chen Gong and Tao Zhou.*)

S. Zhong, T. Zhou, S. Wan, J. Yang, and C. Gong are with the PCA Lab, the Key Laboratory of Intelligent Perception and Systems for High-Dimensional Information of Ministry of Education, the Jiangsu Key Laboratory of Image and Video Understanding for Social Security, and the School of Computer Science and Engineering, Nanjing University of Science and Technology, Nanjing, 210094, P. R. China. (e-mail: zhongsw_91@foxmail.com; taozhou.ai@gmail.com; wansheng315@hotmail.com; csjyang@njust.edu.cn; chen.gong@njust.edu.cn).

Currently, graph-based semi-supervised methods [15]–[17] have widely been applied in solving the issue of small-sized labeled data. Among them, the examples are represented by graph nodes, and the label information is propagated from the labeled nodes to the unlabeled ones via graph edges under the regularization of graph smoothness. However, traditional semi-supervised methods still face two main challenges at extremely low label rates. The first challenge is that traditional propagation algorithms degenerate at an extremely low label rate [20], [21]. The second one is that graph propagation is usually conducted on a single and fixed graph [15], [16]. To solve the above challenges, in this paper we propose a graph-based "Dynamic Spectral-Spatial Poisson Learning" (DSSPL) algorithm. Specifically, DSSPL includes three novel procedures to fully exploit the very limited supervisory information, which are:

1) Poisson Learning (PL) [22] is employed for propagating labels on graphs. In the situation of scarce labeled examples, traditional Laplace equation based label propagation methods [15], [17], [19] will produce an almost constant solution, so the "propagation power" of scarce labels is quite limited, leading to nearly constant label values on unlabeled examples with sharp spikes at the labeled examples. To address this issue, PL is utilized as it can remove average label values and center the labels of all examples at zero when conducting label propagation. Therefore, the propagation power is strengthened and unlabeled examples can receive proper label assignments.
2) Spectral and spatial graphs are constructed and then fused in the iterative label propagation procedure. Specifically, DSSPL constructs two different graphs according to the spectral and spatial feature similarities of HSI. As a sequel, the global and local connective relationships among pixels can be fully exploited, and then the two graphs are fused according to the label propagation results in each iteration.
3) A feedback strategy is proposed by our DSSPL to dynamically update the fused graph according to the label information generated by each iteration. Specifically, due to the extremely limited labeled examples, the similarities between pairwise examples revealed by the initial graph are inaccurate for classifying unlabeled examples. Consequently, DSSPL refines the graph by strengthening the correlation among examples with similar labels in a feedback way, and the updated graph in turn improves the label propagation results. Thus graph refinement and label determinations reinforce each other iteratively and make up for insufficient supervisory information.

In summary, our DSSPL adapts to the small-sized training example problem by employing the PL scheme, fusing the spatial and spectral graphs, and designing a feedback strategy for graph updating.

The remainder of this paper is organized as follows: The related works on HSI classification are briefly reviewed in Section II. Section III describes the details of the proposed DSSPL and explains the key procedures including graph construction, spectral-spatial graph fusion, and dynamic graph updating with feedback. Section IV computes the time complexity of the proposed DSSPL. In Section V, the experimental results of DSSPL are shown when compared with seven representative HSI classification methods on three public HSI datasets. Finally, Section VI concludes the entire paper.

II. Related Works

In this section, we briefly review existing methods for solving the small-sized labeled examples problem in HSI classification, and introduce the typical learning algorithms on graph.

A. Small-Sized Example Problem in HSI Classification

There are two main groups of methods proposed to handle the situation where only limited training examples are available for HSI classification, namely graph-based and pseudo labeling methods.

Graph-based methods usually follow the semi-supervised paradigm. Camp-Valls *et al.* [23] firstly introduced graph-based semi-supervised learning into HSI classification, which assigns relative importance values to the labeled examples by using a kernel-based graph. To avoid the inference of data noise, researchers constructed sparse graphs [24], [25] to select more discriminative neighbors in the feature space. In addition, graph construction was improved by graph fusion and hypergraph construction methods [26], [27]. For instance, Liao *et al.* [26] proposed a local graph-based fusion method to integrate spectral and spatial graphs, which yields a clear improvement over the prior approaches. Liu *et al.* [27] adopted low-rank representation to represent HSI data and constructed a hypergraph model to effectively capture high-order relationships among examples. However, most of the above methods focus on extracting effective features from HSIs based on fixed graphs, which do not contain a graph updating procedure to further improve HSI classification performance.

Pseudo labeling has also been used for addressing the small-sized labeled data problem. For example, considering the fact that deep learning methods generally require a large number of labeled examples, Wu *et al.* [28] proposed a deep convolutional recurrent neural networks by using the pseudo labels for HSI classification. To extract pseudo labels with high quality, a constrained Dirichlet process mixture model was proposed, and then the network was pre-trained using pseudo labels and fine-tuned on limited labeled examples. Besides, Sellars *et al.* [29] proposed a graph-based learning method with superpixel-based pseudo labeling for HSI classification with a low label rate. The main concept is that the pixels belonging to the same superpixel may share the same labels with a high probability, therefore pseudo labels were decided according to image segmentation results, and graph-based learning was then conducted for final classification. However, inaccurate pseudo labels will cause an accumulation of classification error, and thus the labels should be carefully decided.

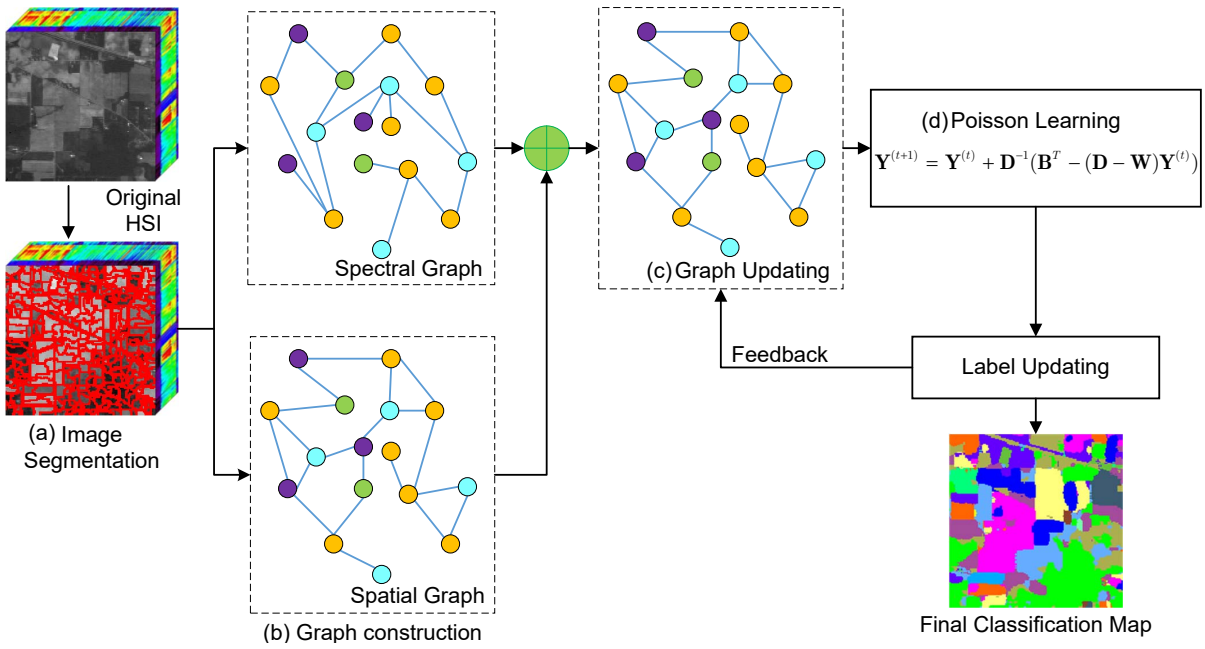


Fig. 1. The framework of our proposed DSSPL. (a) represents superpixels segmented on the original HSI. (b) denotes the construction of spectral and spatial graphs based on three features extracted from superpixels. In the iteration process, (c) and (d) constitute an iterative process, where (c) updates the fused graph by combining spectral and spatial graphs according to the updated labels in a feedback way, and (d) represents the Poisson Learning process for label propagation on the fused graph.

B. Learning on Graphs

Graph theory has been utilized to solve various machine learning problems, such as classification [30]–[32], clustering [33], and dimension reduction [34]. Due to the ability to represent data manifold and encoding pairwise similarities between examples, graph-based label propagation is a common strategy to solve semi-supervised learning problems. In this case, labels from scarce labeled data are propagated to unlabeled ones over a graph [35].

There are several algorithms proposed for single graph input such as Gaussian Field and Harmonic Functions (GFHF) [19] and Flexible Manifold Embedding (FME) [36]. To handle ambiguous examples, Gong *et al.* [37] introduced a local smoothness term called deformed graph Laplacian, which effectively prevented erroneous label propagation by suppressing the labels of ambiguous examples. A Teaching-to-Learn and Learning-to-Teach label propagation strategy was then proposed in [38], where label propagation was conducted from simple examples to more difficult ones to avoid tackling ambiguous examples in the primary propagation stage. In addition, lots of graph-based neural networks were also proposed for classification on graph nodes. For example, Gori *et al.* [39] first introduced the concept of Graph Neural Network (GNN), which can aggregate the node features in a graph and properly embed the entire graph in a new discriminative space. Bruna *et al.* [40] proposed graph-based analogues of convolutional architectures, called Graph Convolutional Network (GCN), which reduced the number of parameters in GNN without worsening the testing error. Afterwards, a lot of modification methods of GCN were proposed, such as [41]–[43]. In [44], a GCN model is constructed using the multi-layer convolutional features extracted by a pre-trained CNN, which can mine the CNN-GCN contextual information from hyperspectral image patches.

Moreover, due to the fact that a single graph cannot handle the data with multiple types of features, several multi-graph fusion methods are proposed to handle the multi-view data and improve the classification result. For example, an adaptive multi-modal graph fusion method was proposed by Cai *et al.* [45], which integrates heterogeneous visual features by learning a commonly shared class indicator matrix and graph weights simultaneously. Karasuyama *et al.* [46] proposed a linear combination method of multiple graphs with sparse weights, in which the sparse weights are beneficial for eliminating irrelevant graphs and identify important ones. In [47], [48], curriculum learning was employed for multi-modal learning. Specifically, each modality is associated with a "teacher", and all the teachers cooperate with each other to establish an accurate curriculum sequence and improve label propagation quality. These multiple graph-based learning methods have been proved to outperform the learning approaches on a single graph. In our proposed DSSPL, a graph fusion process is implemented from the spectral and spatial graphs for HSI classification, and the fusion guides the label propagation process, forming a unified framework that iteratively refines pixel labels and updates the fused graphs.

III. The Proposed Method

This section provides the details of our proposed DSSPL method. Fig. 1 presents the overall diagram of DSSPL, which consists of four stages. Firstly, the original HSI is over-segmented to a certain amount of compact small regions (Section III-A). Secondly, the spectral graph and spatial graph

are respectively constructed based on different features (Section III-B). Thirdly, PL is exploited for label propagation on a graph with scarce labeled examples (Section III-C). Finally, spectral graph and spatial graph are fused and updated according to a feedback strategy (Section III-D). These two stages of label propagation and graph updating alternate and compose an iterative process, which is the core component of our DSSPL.

A. Superpixel Generation

First of all, a segmentation process is conducted on the original HSI to generate compact image regions (a.k.a. superpixels), in which Simple Linear Iterative Clustering (SLIC) [49] is adopted to produce homogeneous regions containing spatial neighboring pixels with very similar spectral features. SLIC initializes cluster centers by sampling pixels at a regular grid, then repeatedly assigns pixels to the best matching centers and computes new cluster centers. After segmentation, each superpixel is regarded as a graph node, and the features of the superpixels are extracted for graph construction. In this way, the number of graph nodes can be largely reduced, and thus the heavy computational burden is alleviated. It should be noted that if one superpixel contains the most pixels belonging to a certain class, then this class label is assigned to the whole superpixel. In HSI classification, image segmentation has been proved to produce better classification maps than directly using raw HSI data [50], [51].

B. Graph Construction

After superpixel generation on an HSI, N superpixels are obtained which are denoted by $\{S_i\}_{i=1}^{N}$. Each superpixel S_i contains n_i pixels. The spectral feature of the j-th pixel in S_i is denoted as $\mathbf{x}_{ij}$, which is the original spectral reflectance of each pixel in the raw HSI data. The position of the j-th pixel is denoted as $\mathbf{p}_{ij}$ ($j = 1, 2, \ldots, n_i$). For each superpixel S_i, its local spectral feature $\mathbf{f}_i^{spec}$, spatial location feature $\mathbf{f}_i^{spat}$, and contextual spectral feature $\mathbf{f}_i^{c}$ are extracted by deploying the mean of spectral features, mean of pixel locations, and weighted combination of neighbor spectral features [29], [52], which are:

$$\begin{cases} \mathbf{f}_i^{spec} = \dfrac{1}{n_i}\sum_{j=1}^{n_i} \mathbf{x}_{ij}, \\ \mathbf{f}_i^{spat} = \dfrac{1}{n_i}\sum_{j=1}^{n_i} \mathbf{p}_{ij}, \\ \mathbf{f}_i^{c} = \sum_{S_j \in \mathcal{N}^{spat}(S_i)} \omega_{ij}\mathbf{f}_j^{spec}, \end{cases} \tag{1}$$

where $\mathcal{N}^{spat}(S_i)$ denotes the set containing all the adjacent superpixels of S_i, and $\mathbf{f}_i^{spec}$, $\mathbf{f}_i^{spat}$ and $\mathbf{f}_i^{c}$ are column vectors. The coefficient ω_{ij} is defined according to the distance between the spatial location features of superpixels S_i and S_j, which is computed by:

$$\omega_{ij} = \frac{\exp(-\|\mathbf{f}_i^{spat} - \mathbf{f}_j^{spat}\|^2)}{\sum_{S_j \in \mathcal{N}^{spat}(S_i)} \exp(-\|\mathbf{f}_i^{spat} - \mathbf{f}_j^{spat}\|^2)}. \tag{2}$$

Then the complete feature vector $\mathbf{f}_i$ characterizing S_i is obtained by cascading the above features, namely:

$$\mathbf{f}_i = [\mathbf{f}_i^{spec}; \mathbf{f}_i^{spat}; \mathbf{f}_i^{c}]; \tag{3}$$

where "$[\cdot;\cdot;\cdot]$" denotes the operation of vector concatenation in column. After feature normalization, the similarity between the two superpixels S_i and S_j can be expressed as:

$$w_{ij} = \exp\left(-4\|\mathbf{f}_i - \mathbf{f}_j\|^2 / d(\mathbf{f}_i; \mathcal{N}_k(S_i))^2\right); \tag{4}$$

where $d(\mathbf{f}_i; \mathcal{N}_k(S_i))$ is the Euclidean distance between $\mathbf{f}_i$ and the feature vector of the k-th nearest neighbor of S_i. It is noteworthy that $\mathbf{f}_i^{spec}$ and $\mathbf{f}_i^{spat}$ are both "inner-superpixel" features, while $\mathbf{f}_i^{c}$ is an "intra-superpixel" feature. To summarize, the similarity w_{ij} is measured based on a synthetic evaluation of spectral consistency, spatial distance, and similarity of adjacent superpixels.

Afterwards, two k-Nearest Neighbors (k-NN) graphs are constructed by utilizing the above similarity. Specifically, k-NN can be viewed from the perspective of either spatial distance or spectral distance. These two perspectives respectively induce spectral graph and spatial graph defined as $\mathcal{G}^{spec} = (\mathcal{V}; \mathcal{E}^{spec})$ and $\mathcal{G}^{spat} = (\mathcal{V}; \mathcal{E}^{spat})$, where $\mathcal{V}$ and $\mathcal{E}$ are the sets of nodes and edges accordingly. It is noted that in our proposed DSSPL, $\mathcal{G}^{spec}$ and $\mathcal{G}^{spat}$ share the same set of nodes, namely $\mathcal{V} = \{S_1, S_2, \ldots, S_N\}$, but $\mathcal{E}^{spec}$ and $\mathcal{E}^{spat}$ are different, as they correspond to the spectral and spatial connective relationship among superpixels. The two graphs $\mathcal{G}^{spec}$ and $\mathcal{G}^{spat}$ are represented by the adjacency matrices $\mathbf{W}^{spec}$ and $\mathbf{W}^{spat}$, respectively, in which the (i, j)-th elements in them are respectively expressed as:

$$w_{ij}^{spec} = \begin{cases} w_{ij}, & \text{if } S_i \in \mathcal{N}^{spec}(S_j) \text{ or } S_j \in \mathcal{N}^{spec}(S_i) \\ 0, & \text{otherwise} \end{cases}; \tag{5}$$

$$w_{ij}^{spat} = \begin{cases} w_{ij}, & \text{if } S_i \in \mathcal{N}^{spat}(S_j) \text{ or } S_j \in \mathcal{N}^{spat}(S_i) \\ 0, & \text{otherwise} \end{cases}; \tag{6}$$

where spectral neighbor set $\mathcal{N}^{spec}(S_i)$ contains k nearest neighbors regarding spectral features. According to [22], k is set to 10 in DSSPL. As shown in Fig. 1, the two graphs $\mathcal{G}^{spec}$ and $\mathcal{G}^{spat}$ are dynamically fused according to the learned labels in each iteration. It is worth noting that the iterative process initializes with $\mathcal{G}^{spat}$ for label propagation.

C. Poisson Learning

After the initial graph of HSI (*i.e.*, $\mathcal{G}^{spat}$) is acquired, PL is applied to handle the extremely limited labels on the graph. Most of the existing approaches are not specifically designed for the classification on an extremely low label rate, so their performance in this situation largely deteriorates. In contrast, PL explores the degeneration process of label propagation when there are very few labeled examples, and deploys a specific strategy to optimize the classifier. Considering a graph with N nodes including m labeled nodes across c classes, let the label matrix dubbed as $\mathbf{Y}^{N\times c} = (\mathbf{y}_1, \ldots, \mathbf{y}_m, \mathbf{y}_{m+1}, \ldots, \mathbf{y}_N)^{\top}$, in which the first m column vectors represent the labels of m labeled nodes. Besides, in the

initial label matrix $Y^{(0)}$ for propagation, the $(i;j)$-th element is expressed by

$$(Y^{(0)})_{ij} = y_{ij} = \begin{cases} 1; \text{ if } 1 \le i \le m \text{ and } S_i \in \text{class } j \\ 0; \text{ otherwise} \end{cases}: \quad (7)$$

In this study, we use a classical label propagation method, namely GFHF [19], to conduct label propagation. Thus, it needs to solve the following problem:

$$\begin{cases} u(S_i) = y_i; i = 1;2;:::;m \\ \mathcal{L}(u(S_i)) = 0; i = 1;:::;N \end{cases}; \quad (8)$$

where u is the mapping function for each graph node, and the unnormalized graph Laplacian operator $\mathcal{L}$ is given by

$$\mathcal{L}(u(S_i)) = \sum_{j=1}^{N} w_{ij}(u(S_i) - u(S_j))^2: \quad (9)$$

Although the above Laplacian equation-based GFHF has been widely used for semi-supervised learning, its performance dramatically decreases when the label rate is extremely low [20], [21]. The main reasons can be explained from the following two aspects:

1) *Theoretical proof*: Nadler *et al.* [53] presented a theoretical study of Laplacian equation-based label propagation on a high-dimensional graph model. Let the margin density $p(S)$ be an unknown smooth density on a compact domain $! \subset \mathcal{R}^d$ and $d \ge 2$. For $N \to \infty$, there exist continuous functions $u(S)$ satisfying the constraints in Eq. (8), but $u(S) \longrightarrow c$ for all $S_i; i = (N-m);:::;N$, in which c is a constant. This theorem implicates that as the number of unlabeled examples increases, the solution of Eq. (8) has the form of an almost everywhere constant function, leading to an inaccurate label propagation.

2) *Random walk interpretation*: A more intuitive way of the issues with Laplacian learning at a low label rate comes from the interpretation of random walk, which also leads to the original concept of Poisson Learning [22]. We release random walkers from labeled nodes and let them explore the graph with the transition probabilities from the j-th node to i-th node, which can be expressed as:

$$\mathbb{P}(X_k^j = S_i | X_{k-1}^j = S_j) = d_j^{-1} w_{ij}; \quad (10)$$

where d_j is the degree of the j-th node with $d_j = \sum_{i=1}^{N} w_{ij}$, and X_k^j denotes the position of the random walker released from the j-th labeled node in the k-th iteration. Each random walker X_k^j carries its label information y_j when exploring the graph. Then the solution of Eq. (8) could be described by the expectation of the labels y_j of all the X_k^j that visit S_i with $0 \le k \le t$ and $1 \le j \le m$, which is [54]:

$$u_t^L(S_i) = \sum_{k=0}^{t} \sum_{j=1}^{m} y_j \mathbb{P}(X_k^j = S_i): \quad (11)$$

When there are enough labeled nodes, the random walkers with label information could reach all the nodes in a short time. However, when there are very few labeled nodes, it takes a long time for a few random walkers to traverse every node in the graph. It is also worth noting that if the walking takes much time, the distribution of X_k^j is very close to an invariant distribution:

$$\lim_{k \to \infty} \mathbb{P}(X_k^j = S_i) = \frac{d_i}{\sum_{i=1}^{N} d_i}: \quad (12)$$

In this case, the above solution records a blind average of existing labels, which is [22]:

$$\sum_{j=1}^{m} y_j \mathbb{P}(X_k^j = S_j) \approx \frac{d_i}{\sum_{i=1}^{N} d_i} \sum_{j=1}^{m} y_j: \quad (13)$$

This suggests that when the walking time is too large, the solution of Laplace equation does not depend on the structure of a graph but only reflects a blind average of labels. As a result, Laplace equation-based GFHF returns almost identical labels for all unlabeled nodes, which conveys very little useful class information.

To address the above problem, PL is utilized to replace the Laplace equation (as in Eq. (8)) with the following Poisson equation:

$$\mathcal{L}(u(S_i)) = \sum_{j=1}^{N} (y_j - \bar{y})_{ij} \text{ for } i = 1;2;:::;N; \quad (14)$$

where $\sum_{i=1}^{n} d_i u(S_i) = 0$. The average label vector $\bar{y}$ is defined as $\bar{y} = \frac{1}{m}\sum_{j=1}^{m} y_j$, and $_{ij}$ is 1 if and only if $i = j$. The Poisson equation is also solved from the view of random walk. Based on Eq. (11), PL proposes to subtract off the tail behavior from $u_t(S_i)$ to avoid the solution dominated by the average of labels. Meanwhile, PL normalizes $u_t(S_i)$ by d_i, which achieves the process by:

$$u_t(S_i) = \frac{1}{d_i} \sum_{k=0}^{t} \sum_{j=1}^{m} (y_j - \bar{y}) \mathbb{P}(X_k^j = S_i): \quad (15)$$

Considering the transition probability of random walkers, we have

$$\begin{aligned} &\sum_{k=0}^{t} \sum_{j=1}^{m} (y_j - \bar{y}) \mathbb{P}(X_{k+1}^j = S_i) \\ &= \sum_{k=0}^{t} \sum_{j=1}^{m} (y_j - \bar{y}) \sum_{l=1}^{N} \frac{w_{il}}{d_l} \mathbb{P}(X_k^j = S_l) \\ &= \sum_{l=1}^{N} \frac{w_{il}}{d_i} u_T(S_l): \end{aligned} \quad (16)$$

Therefore, $u_{t+1}(S_i)$ can be expressed by $u_t(S_i)$ as:

$$\begin{aligned} u_{t+1}(S_i) &= d_i^{-1} \sum_{k=0}^{t+1} \sum_{j=1}^{m} (y_j - \bar{y}) \mathbb{P}(X_k^j = S_i) \\ &= d_i^{-1} \sum_{j=1}^{m} (y_j - \bar{y})_{ij} + d_i^{-1} \sum_{l=1}^{N} \frac{w_{il}}{d_i} u_t(S_l) \\ &= d_i^{-1} \sum_{j=1}^{m} (y_j - \bar{y})_{ij} + u(S_i) - d_i^{-1} \mathcal{L}(u_t(S_i)) \\ &= u_t(S_i) + d_i^{-1} \left(\sum_{j=1}^{m} (y_j - \bar{y})_{ij} - \mathcal{L}(u_t(S_i)) \right): \end{aligned} \quad (17)$$

It has been proved in [22] that when $t \to \infty$, we have $u_t \to u$ with u being the unique solution of the Poisson equation in Eq. (14). From Eq. (15) we can see that the solution of Poisson equation only records the short-time behavior of the random walkers, avoiding the interference of the shift bias caused by the long walking time. According to Eq. (17), the PL problem can be solved via an iterative process as:

$$\mathbf{Y}^{(t+1)} = \mathbf{Y}^{(t)} + \mathbf{D}^{-1}(\mathbf{B}^{\top} - (\mathbf{D} - \mathbf{W})\mathbf{Y}^{(t)}); \tag{18}$$

where $\mathbf{B} = [\mathbf{Y} - \mathbf{1}_N \mathbf{1}_N^{\top}\bar{\mathbf{y}}^{\top}; \mathbf{0}_{c\times(N-m)}]$, and $\mathbf{D}$ is a diagonal degree matrix with $D_{ii} = d_i$.

D. Graph Updating via a Feedback Strategy

The PL algorithm improves the performance of label propagation when there are only a few labeled examples available. However, PL only works on a fixed pre-established graph, which may not be accurate when the supervisory information is very limited. To alleviate the issue, we introduce a graph updating operation to timely extract and strengthen the supervisory information during the iteration process. In Section III-B, $\mathcal{G}^{spec}$, and $\mathcal{G}^{spat}$ have been constructed, which record the spectral and spatial connection relationship among nodes. As introduced in Section III-B, $\mathcal{G}^{spec}$ and $\mathcal{G}^{spat}$ are both sparse graphs, which record the nearest neighbors of graph nodes in terms of spectral feature and spatial location. To synthetically capture the graph structure, a multi-graph fusion method is proposed. In the fusion strategy, the label propagation is modeled by multivariate Gaussian, and the individual graph distribution is modeled by using the Ising model [55]. Thus, the fusion parameter is estimated by maximizing the posterior probability of multi-graph under the condition of labels by Bayesian theory. The fused graph is denoted as $\mathcal{G}^{ss} = (\mathcal{V}; \mathcal{E}^{ss})$ with the spectral-spatial adjacency matrix $\mathbf{W}^{ss}$. Moreover, the up-to-date labels are fed back to refine the graph for emphasizing the connection between nodes with similar label vectors. The details are explained as follows.

Firstly, the fusion of graphs is based on the assumption that the spectral-spatial adjacency matrix $\mathbf{W}^{ss}$ can be obtained by a linear combination of $\mathbf{W}^{spec}$ and $\mathbf{W}^{spat}$ [56], namely:

$$\mathbf{W}^{ss} = (1 - \alpha)\mathbf{W}^{spec} + \alpha\mathbf{W}^{spat}; \tag{19}$$

where α is a trade-off coefficient, and it is critical to decide the proper value of α. Analogously, $\mathbf{L}^{ss}$ is a linear combination of Laplacian matrices $\mathbf{L}^{spec}$ and $\mathbf{L}^{spat}$ as:

$$\mathbf{L}^{ss} = (1 - \alpha)\mathbf{L}^{spec} + \alpha\mathbf{L}^{spat}. \tag{20}$$

According to [19], the propagation of label vector $\hat{\mathbf{y}}_i = (y_{1i}; y_{2i}; \cdots; y_{Ni})^{\top}$ with $i = 1; \cdots; c$ on each graph can be regarded as a Gaussian process, in which the covariance matrix equals to the corresponding Laplacian matrix. Based on our assumption that the adjacency matrix of fused graph is a linear combination of adjacency matrices of all individual graphs, the likelihood probability $p(\mathbf{Y}|\mathbf{W}^{ss})$ can be written as

$$p(\mathbf{Y}|\mathbf{W}^{ss}) \propto \exp\left(-\frac{1}{2}\sum_{i=1}^{c}\hat{\mathbf{y}}_i^{\top}(\mathbf{L}^{ss} + \mathbf{I}/l^2)\hat{\mathbf{y}}_i\right); \tag{21}$$

where $\mathbf{L}^{ss} + \mathbf{I}/l^2$ is the regularized Laplacian matrix with l being the regularization parameter, and $\mathbf{I}$ is an identity matrix. According to the classical Ising model [57], the prior probability of $\mathbf{W}^{ss}$ can then be expressed as:

$$p(\mathbf{W}^{ss}) \propto \exp\left(\sum_{(i;j)\in E^{ss}} w_{ij}^{ss}\mathbf{y}_i^{\top}\mathbf{y}_j\right); \tag{22}$$

where w_{ij}^{ss} is the $(i; j)$-th element of $\mathbf{W}^{ss}$. To construct the model of graph fusion under the condition of specific labels, the posterior probability of $\mathbf{W}^{ss}$ can be computed according to the Bayesian theory as

$$\begin{aligned} &p(\mathbf{W}^{ss}|\mathbf{Y}) \propto p(\mathbf{W}^{ss})p(\mathbf{Y}|\mathbf{W}^{ss}) \\ \Rightarrow &p(\mathbf{W}^{ss}|\mathbf{Y}) \propto \\ &\exp\left(\sum_{(i;j)\in E^{ss}} w_{ij}^{ss}\mathbf{y}_i^{\top}\mathbf{y}_j - \frac{1}{2}\sum_{i=1}^{c}\hat{\mathbf{y}}_i^{\top}(\mathbf{L}^{ss} + \mathbf{I}/k^2)\hat{\mathbf{y}}_i\right). \end{aligned} \tag{23}$$

In each iteration, given the predicted label matrix $\mathbf{Y}$, the optimal fused graph is supposed to show the highest posterior possibility in describing the labels. Consequently, the optimal α can be obtained by solving the following problem:

$$\alpha = \arg\max p(\mathbf{W}^{ss}|\mathbf{Y}). \tag{24}$$

Therefore, the parameter α varies according to the current propagation result, which leads to a dynamic spectral-spatial description on the investigated HSI with the highest posterior probability $p(\mathbf{W}^{ss}|\mathbf{Y})$.

Moreover, the spectral-spatial adjacency matrix $\mathbf{W}^{ss}$ is further improved by fusing the information extracted from the output labels in each iteration [58]. Specifically, the learned label matrix $\mathbf{Y}$ contains the information about class labels, and the correlation of these labels $\mathbf{Y}\mathbf{Y}^{\top}$ can be viewed as the similarity between data points in the label space. This correlation is fed back into the input adjacency matrix to emphasize the relationship among the nodes with high similarity, where the feedback process can be expressed as:

$$\mathbf{W} = \mathbf{W}^{ss}(\mathbf{W}^{ss} + \beta\mathbf{Y}\mathbf{Y}^{\top})(\mathbf{W}^{ss})^{\top} + \gamma\mathbf{I}; \tag{25}$$

where β is a pre-defined nonnegative coefficient, and $\mathbf{I}$ is an identity matrix. Thus, the graph is refined by the output labels in the previous iteration and the refined graph in turn improves label propagation results in the next iteration. The implementation of DSSPL is summarized in Algorithm 1.

IV. Complexity Analysis

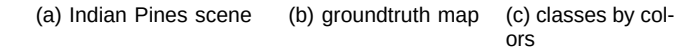
In the iteration phase of our DSSPL, Eq. (24) can be solved by the linear programming with the computational complexity of $\mathcal{O}(N^2)$ [59], where N is the number of graph nodes (*i.e.*, the number of superpixels). The update of the graph $\mathbf{W}$ in Eq. (25) can be accelerated using the sparsity of $\mathbf{W}$, and thus the computational complexity is $\mathcal{O}(kN^2)$ [60], where k is the number of nearest neighbors in adjacency matrices $\mathbf{W}^{spec}$ and $\mathbf{W}^{spat}$. For the propagation of labels using PL via Eq. (18), the computational complexity is $\mathcal{O}(N^2)$ [22]. To summarize, since the proposed DSSPL takes $\mathcal{T}$ iterations, the overall computational complexity is $\mathcal{O}(k\mathcal{T}N^2)$,where $k \ll N$ and $\mathcal{T} \ll N$.

Algorithm 1 The Proposed DSSPL Algorithm

Input: Hyperspectral image H; initial label matrix $Y^{(0)}$; number of superpixels N; number of iterations T;

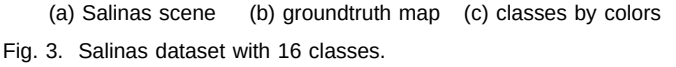
1: Segment H into N superpixels $\{S_i\}_{i=1}^{N}$ via SLIC;
2: Construct the initial k-NN spectral graph W^{spec} and initial k-NN spatial graph W^{spat} via Eq. (5) and Eq. (6);
3: **for** $t = 1$ to T **do**
4: Decide the value of α via Eq. (24);
5: Update the fused spectral-spatial graph W^{ss} via Eq. (19);
6: Update W based on W^{ss} and the feedback of posterior label correlation via Eq. (25);
7: Update degree matrix D;
8: Propagate labels using PL via Eq. (18);
9: Compute binary label vectors $\{y_i\}_{i=1}^{N}$;
10: **end for**

Output: Predicted label for each unlabeled pixel.



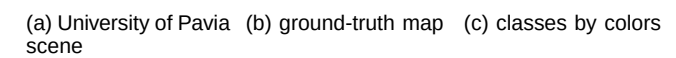
(a) Indian Pines scene (b) groundtruth map (c) classes by colors

Fig. 2. Indian Pines dataset with 16 classes.



(a) Salinas scene (b) groundtruth map (c) classes by colors

Fig. 3. Salinas dataset with 16 classes.



(a) University of Pavia (b) ground-truth map (c) classes by colors scene

Fig. 4. University of Pavia dataset with 9 classes.

V. Experimental Results and Discussions

In this section, we conduct experiments to show the effectiveness of the proposed model. We first introduce the datasets for our experiments as well as the experimental settings. Then, we provide the experimental results, parameter sensitivity analysis, and ablation study.

A. Datasets

Three real-world hyperspectral image datasets[1], i.e., the Indian Pines, Salinas, and University of Pavia, are used in our experiments. The details of each dataset are provided as follows.

1) Indian Pines: This is a well-known Airborne Visible/Infrared Imaging Spectrometer (AVIRIS) image dataset. Figs. 2 (a)(b)(c) show the image data, the corresponding groundtruth, and class colors, respectively. The image was recorded in June 1992 from an area of mixed agriculture and forestry in Northwestern Indiana, USA with the wavelength in the range of 0.4 m to 2.5 m and spatial resolution of 20 m. It has the size of 145×145 pixels with 220 bands, in which 20 water absorption bands (bands 104-108 and 150-163, 220) were removed.

[1]All datasets are available at http://www.ehu.eus/ccwintco/index.php?title=Hyperspectral_Remote_Sensing_Scenes

2) Salinas: This dataset was collected by the AVIRIS sensor over the Salinas Valley, California, with a spatial resolution of 3.7 m per pixel and the spectral resolution of 10 nm. It has a size of 512×217 with 224 bands, consisting of 16 vegetation classes. Figs. 3 (a)(b)(c) show the Salinas scene, groundtruth, and the corresponding groundtruth class labels, respectively.

3) University of Pavia: This dataset captures the Pavia University of Italy with the ROSIS sensor. It consists of $610 \times 340 \times 115$ pixels with a spatial resolution of 1.3 m and a spectral coverage ranging from 0.43 m to 0.86 m with a spectral resolution of 4 nm (12 most noisy channels were removed before experiments). Fig. 4 (a) shows the scene of the University of Pavia, and nine classes of interest are considered for this image as shown in Fig. 4 (b) and (c).

B. Experimental Settings

To evaluate the performance of the proposed DSSPL, three traditional HSI classification methods, including SVM [61], GFHF [19], and PL [22], are employed for comparison. In addition, we compare the proposed DSSPL with a method based on spectral-spatial feature extraction, i.e., Fusion of Spectral-Spatial (FSS) [62], a linear regression-based method, i.e., Superpixel-guided Training Sample Enlargement Distance Weighted Linear Regression (STSE-DWLR) [63], together with a graph-based method, i.e., Superpixel Graph Learning

TABLE I
THE HYPERPARAMETER SETTINGS OF OUR DSSPL ON DIFFERENT DATASETS.

	N		
Indian Pines	1400	0.1	0.01
Salinas	1400	0.1	0.1
University of Pavia	2200	0.1	0.01

(SGL) [29]. It is worth noting that both STSE-DWLR and SGL are designed for the limited label situation for HSI classification. Meanwhile, we also compare the proposed DSSPL with a Convolutional Neural Network (CNN)-based method, i.e., deep CNN using Pixel-Pair Features (CNN-PPF) [10].

To comprehensively evaluate the performance of the proposed method on a small number of labeled examples, different numbers of labeled pixels (e.g., 3, 5, or 10) of each class are randomly selected in our experiments. Furthermore, to validate the performance of our DSSPL on large size of labeled data, 30 labeled pixels of each class are also selected in our experiment. We adopt four metrics for performance evaluation, including individual class accuracy, Average Accuracy (AA), Overall Accuracy (OA), and kappa coefficient. In addition, classification maps are presented as a qualitative demonstrations to further evaluate all the compared methods. Due to the random selection of labeled examples, all methods are conducted ten times with different labeled pixels on each dataset, and the mean accuracies and standard deviations over the ten implementations are reported. The hyperparameter settings of our DSSPL on each dataset are listed in Table I.

C. Experimental Results and Analyses

1) Classification Results on Classes: Tables II- IV tabulate the accuracies of each class, AAs, OAs, and kappa coefficients produced by the eight classification methods on the three datasets. The best result is highlighted in bold. The classification maps of all these methods on the three datasets are visualized in Figs. 5-7, in which the areas with significant improvement by our DSSPL are marked via red or yellow rectangles. From these comparison results, it can be observed that our DSSPL obtains the best performance in terms of AA, OA and kappa among all the competitors. For instance, in terms of OA, our DSSPL achieves 4.49%, 0.45%, and 1.05% improvements over the second best method on the Indian Pines, Salinas, and University of Pavia datasets, respectively.

For the performance on individual classes on the Indian Pines dataset, our DSSPL achieves the highest accuracies across 10 out of 16 classes when compared with other methods. Specifically, Classes 2, 3 and 4 represent the categories of “corn-notill”, “corn-mintill” and “corn”, respectively, in which the spectrum is quite similar with each other. For the presence of limited labeled examples, some methods neglect the spatial correlation of pixels, such as SVM, thus they cannot obtain good performance. As a sequel, it is important to introduce spatial information to cooperate with limited spectral features, and this is the main reason that FSS and DSSPL can obtain relatively high accuracies on the three classes, as shown in Table II. Specifically, DSSPL improves the accuracies of the three classes, and the records are 70%, 78% and 85% correspondingly. For Class 15 (i.e., “buildings”), it is worth noting that the accuracy is significantly improved, from 28% of SVM and 43% of CNN-PPF to 88% of DSSPL. Moreover, the Indian Pines dataset is a typical dataset with imbalanced classes. Specifically, there are four classes with less than 100 examples, i.e., class 9 with 20 data points, class 7 with 28 data points, class 1 with 46 data points, and class 16 with 93 data points. From Table IV, it can be noted that DSSPL achieves more than 98% accuracy on all these four classes, and obtains higher performance over other compared methods in classes 1, 7, and 16. For class 9, DSSPL reaches more than 99% accuracy, which is only slightly lower than SGL.

In the Salinas dataset, since the scene is simpler when compared with the other two datasets, most of the compared methods achieve more than 94% in terms of OA. In this case, DSSPL still outperforms other methods and improves AA, OA and kappa by 0.52%, 0.45%, 0.49% when compared with STSE_DWLR, which performs the second best. Specifically, DSSPL achieves 11 best results out of a totally 16 classes. The Salinas dataset does not show an obvious imbalanced data issue, and the accuracies of DSSPL in all the classes are larger than 90%, which proves the robustness of our proposed method. For Classes 8 and 15, which correspond to “grapes untrained” and “vinyard untrained”, DSSPL improves the accuracies by more than 41% and 33% when compared with SVM.

Different from the Indian Pines and Salinas datasets, the regions of different classes are usually small fragments in the University of Pavia dataset, such as “buildings”, “trees” and “roads”. In this case, DSSPL still outperforms the second best method by 1.68%, 1.05%, and 6.21% on AA, OA and kappa coefficient, respectively. It is worth noting that DSSPL achieves 5 best results out of 9 classes. Specifically, for Class 4 (i.e., “tree”), the accuracies of other methods are all lower than 65%, while DSSPL achieves more than 82% accuracy. Similar to the Salinas dataset, there are no particularly small-sized classes, and our DSSPL shows the accuracies of more than 80% on all the classes, which implies the stability of the proposed DSSPL. Overall, the results demonstrate that DSSPL performs the best among all compared approaches across the three datasets in the extremely limited labeled example situation.

2) Impact of the Number of Labeled Examples: To investigate the impact of the numbers of labeled examples, Table V reports the OAs performance of the proposed DSSPL and the compared methods under different numbers (e.g., 3, 5, 10, and 30) of labeled examples in each class. In the Indian Pines dataset, since there are two classes that only contain less than 30 pixels, we instead select 15 pixels from these classes as labeled examples. From the results, we can draw several observations as follows:

The performance (i.e., OAs) of all methods can be improved by increasing the number of labeled examples. Our DSSPL outperforms most of the compared methods on the three datasets when there are 10 or fewer labeled pixels of each class. Although STSE_DWLR achieves the

TABLE II
PER CLASS ACCURACIES, AAS, OAS, AND KAPPA COEFFICIENTS OF DIFFERENT METHODS ON THE INDIAN PINES DATASET WITH 5 TRAINING EXAMPLES OF EACH CLASS.

class	SVM [61]	CNN-PPF [10]	GFHF [64]	PL [22]	FSS [62]	SGL [29]	ST_SLRR [63]	DSSPL
1	0.8848±0.0503	0.6610±0.4565	0.8609±0.0092	0.9130±0.0939	0.7715±0.2624	0.9652±0.0174	0.9848±0.0179	0.9913±0.0052
2	0.2759±0.0820	0.0806±0.0661	0.5564±0.1059	0.4863±0.1156	0.8388±0.1671	0.4856±0.0666	0.4791±0.1938	0.7078±0.1223
3	0.3899±0.1103	0.0856±0.0720	0.5141±0.1133	0.4592±0.1468	0.7759±0.1931	0.5749±0.1588	0.6425±0.1078	0.7829±0.1365
4	0.5586±0.1462	0.5746±0.4312	0.7460±0.1827	0.8468±0.1410	0.7965±0.2227	0.8384±0.1181	0.8350±0.2785	0.7540±0.1922
5	0.6031±0.1702	0.0720±0.1228	0.7714±0.0831	0.7979±0.0963	0.6907±0.3308	0.8110±0.0896	0.8025±0.0953	0.8370±0.0941
6	0.7207±0.1287	0.2265±0.2612	0.7011±0.1765	0.8192±0.1276	0.8138±0.2669	0.8301±0.1209	0.9116±0.1366	0.8940±0.1971
7	0.9286±0.0238	0.4435±0.3174	0.9714±0.0151	0.9429±0.0811	0.7264±0.3080	0.9679±0.0107	0.9786±0.0184	0.9821±0.0189
8	0.7144±0.0947	0.5476±0.3179	0.9920±0.0057	0.9973±0.0043	0.9129±0.1154	0.9969±0.0058	0.9994±0.0010	0.9987±0.0011
9	0.8750±0.1007	0.5800±0.4445	0.9525±0.0368	0.8647±0.0787	0.6610±0.4340	1.0000±0.0000	0.9900±0.0316	0.9900±0.0316
10	0.4840±0.1594	0.3065±0.2118	0.7213±0.1368	0.7267±0.1522	0.7508±0.1868	0.8110±0.0833	0.6989±0.2337	0.8748±0.2131
11	0.3906±0.1404	0.2621±0.1825	0.6360±0.0879	0.6475±0.0725	0.9042±0.0508	0.6610±0.1068	0.8329±0.1925	0.8290±0.1099
12	0.3745±0.0927	0.0954±0.0771	0.6440±0.1784	0.6125±0.1700	0.7921±0.2153	0.6857±0.1333	0.7764±0.1543	0.7248±0.1270
13	0.9254±0.0435	0.6010±0.4274	0.8951±0.0000	0.9906±0.0015	0.8855±0.1820	0.9951±0.0000	0.9951±0.0000	0.9951±0.0000
14	0.6917±0.1135	0.7606±0.2984	0.7324±0.1355	0.8283±0.1297	0.7483±0.2947	0.8603±0.1060	0.8304±0.0977	0.8712±0.1187
15	0.2803±0.0653	0.4328±0.2344	0.7554±0.0972	0.7902±0.1310	0.6480±0.3557	0.8166±0.0777	0.8728±0.0735	0.8506±0.1173
16	0.8903±0.0433	0.9000±0.0970	0.8903±0.0034	0.9892±0.0000	0.8963±0.1414	0.9806±0.0065	0.9514±0.0049	0.9903±0.0034
AA	0.5276±0.0489	0.4144±0.1913	0.7057±0.0221	0.8033±0.0183	0.7883±0.2071	0.8300±0.0244	0.8576±0.0345	0.8702±0.0312
OA	0.4879±0.0355	0.3087±0.1396	0.6499±0.0273	0.7060±0.0356	0.8142±0.0112	0.7270±0.0393	0.8052±0.0598	0.8391±0.0484
Kappa	0.4310±0.0355	0.2488±0.1193	0.6641±0.0301	0.6592±0.0395	0.8028±0.0127	0.6940±0.0411	0.7868±0.0683	0.8232±0.0535

Fig. 5. Groundtruth map and classi cation maps on the Indian Pines dataset using 5 training examples of each class produced by eight classi cation methods.

best performance with 10 labeled pixels of each class on the Indian Pines dataset, our DSSPL also obtains the competitive performance and outperforms other methods by a large margin. Overall, the proposed DSSPL performs better than existing HSI classi cation methods in the situation of the small-sized labeled examples.

The proposed DSSPL achieves the competitive performance and outperforms most of the compared methods in the situation of 30 labeled pixels of each class. Although FSS achieves the highest accuracies on the Indian Pines and University of Pavia datasets, our DSSPL still ranks the second place and performs better than other methods. On the other hand, DSSPL ranks the rst place over all compared methods on the Salinas dataset. This could be interpreted by the characteristics of the three datasets. It is noted that when utilizing the same amount of labeled examples, the performance of OA is higher on the Salinas dataset than on the other two datasets. This indicates that the Salinas is an “easier” dataset than the other two datasets. It can be also observed that the Indian Pines and University of Pavia datasets contain underlying “more complex” data structure than Salinas. In this circumstance, over-segmentation will lose more information than independent treatment over every pixel. Therefore, superpixel-based DSSPL could perform worse than pixel-wise FSS when there are relatively abundant

TABLE III
PER CLASS ACCURACIES, AAS, OAS, AND KAPPA COEFFICIENTS OF DIFFERENT METHODS ON THE SALINAS DATASET WITH 5 TRAINING EXAMPLES OF EACH CLASS.

class	SVM [61]	CNN-PPF [10]	GFHF [19]	PL [22]	FSS [62]	SGL [29]	STSBELM [63]	DSSPL
1	0.9719±0.0130	0.7510±0.4070	0.9985±0.0000	0.9985±0.0000	0.9370±0.0951	0.9905±0.0240	1.0000±0.0000	1.0000±0.0000
2	0.9159±0.1016	0.5708±0.4024	0.9888±0.0062	0.9606±0.0457	0.9679±0.0348	0.9786±0.0166	1.0000±0.0000	1.0000±0.0000
3	0.8261±0.1421	0.2613±0.4114	0.9958±0.0033	1.0000±0.0000	1.0000±0.0000	0.9613±0.0810	1.0000±0.0000	1.0000±0.0000
4	0.9907±0.0039	0.1148±0.3015	0.9753±0.0247	0.9799±0.0068	0.9625±0.0618	0.9799±0.0065	0.8911±0.2440	0.9714±0.0778
5	0.9077±0.0706	0.3869±0.4996	0.9585±0.0600	0.9599±0.0289	0.9814±0.0273	0.9677±0.0249	0.9942±0.0019	0.9564±0.0378
6	0.9759±0.0127	0.7805±0.4120	0.9942±0.0000	0.9942±0.0000	0.9655±0.0414	0.9948±0.0017	0.9887±0.0099	0.9981±0.0004
7	0.9824±0.0160	0.3983±0.3579	0.9971±0.0019	0.9961±0.0027	0.9439±0.0519	0.9960±0.0038	0.8517±0.1988	0.9884±0.0113
8	0.5439±0.1453	0.2916±0.3459	0.9048±0.0457	0.8592±0.0343	0.9093±0.0854	0.8836±0.0659	0.9517±0.0466	0.8686±0.0452
9	0.9493±0.0114	0.7770±0.4097	0.9991±0.0004	0.9495±0.0733	0.9162±0.0913	0.9802±0.0282	1.0000±0.0000	0.9985±0.0046
10	0.6873±0.0931	0.1750±0.2060	0.7322±0.0510	0.7729±0.0792	0.9439±0.0675	0.7939±0.0901	0.8883±0.2688	0.8846±0.0771
11	0.8460±0.1069	0.2826±0.3331	0.9562±0.0099	0.9522±0.0059	0.9516±0.0428	0.9355±0.0440	0.9326±0.0956	0.9583±0.0435
12	0.9524±0.0402	0.2534±0.2426	1.0000±0.0000	0.9485±0.0397	1.0000±0.0000	1.0000±0.0000	1.0000±0.0000	1.0000±0.0000
13	0.9798±0.0158	0.6232±0.4727	0.9771±0.0000	0.9796±0.0148	0.9780±0.0359	0.9781±0.0029	0.9152±0.1527	0.9028±0.1450
14	0.8823±0.0519	0.5114±0.3886	0.9407±0.0135	0.9062±0.0603	0.9488±0.0542	0.9149±0.0437	0.9600±0.0296	0.9621±0.0264
15	0.6401±0.1260	0.0334±0.0607	0.9266±0.0408	0.9603±0.0425	0.9313±0.0620	0.9670±0.0315	0.9783±0.0287	0.9797±0.0235
16	0.8345±0.1022	0.3582±0.3672	0.9792±0.0028	0.9956±0.0047	0.9514±0.0314	0.9904±0.0174	1.0000±0.0000	1.0000±0.0000
AA	0.8679±0.0139	0.4106±0.2100	0.9590±0.0065	0.9508±0.0130	0.9556±0.0518	0.9570±0.0072	0.9595±0.0456	0.9647±0.0220
OA	0.7981±0.0267	0.3943±0.2196	0.9473±0.0100	0.9347±0.0129	0.9549±0.0074	0.9469±0.0122	0.9630±0.0346	0.9675±0.0178
Kappa	0.7764±0.0291	0.3499±0.2210	0.9414±0.0111	0.9276±0.0142	0.9498±0.0082	0.9411±0.0135	0.9589±0.0385	0.9638±0.0199

Fig. 6. Groundtruth map and classification maps on the Salinas dataset using 5 training examples of each class produced by eight classification methods.

labeled examples. Besides, DSSPL yields a higher OA, AA, and kappa coefficient than SGL and STSBELM that are designed for HSI classification problem under small-sized labeled data. Therefore, the results indicate the effectiveness and stability of the proposed DSSPL.

D. Convergence Analysis

Fig. 8 demonstrates the convergence curves of the proposed DSSPL on all used datasets. Specifically, we plot the stopping criteria e in each iteration, in which e is defined as the percentage of changed classification area comparing the classification map in this iteration to the one in the last iteration. It can be observed that the value of e decreases when the number of iterations n^{iter} increases on all three datasets, and our model converges in about 6-13 iterations. Therefore, this is proved that the proposed DSSPL converges within a small number of iterations.

E. Parameter Sensitivity

In this experiment, we will evaluate the parameter sensitivity of the classification performance to different experimental settings of our DSSPL. Fig. 9 shows the OAs of our DSSPL with respect to the different numbers of superpixels N on the three datasets. From the curves presented in Fig. 9, it can be seen that OA increases when the number of superpixels (N) varies from 200 to 1400 on the Indian Pines and Salinas datasets, and we can obtain satisfactory performance with 1400 superpixels on the two datasets. Besides, for the University of Pavia dataset, OA starts to fluctuate after N reaches about 2200. Thus, in our experiment, N is set to 1400 for both the Indian Pines and Salinas datasets, and 2200 for the University of Pavia dataset.

Fig. 10 exhibits the OAs of the proposed DSSPL with respect to different values of the coefficients and . Here, we report the average results over ten repetitions. From the results in Fig. 10, it can be observed that we obtain stable OA performance, i.e., higher than 80% for the Indian Pines

TABLE IV
PER CLASS ACCURACIES, AAS, OAS, AND KAPPA COEFFICIENTS OF DIFFERENT METHODS ON THE UNIVERSITY OF PAVIA DATASET WITH 5 TRAINING EXAMPLES OF EACH CLASS.

class	SVM [61]	CNN-PPF [10]	GFHF [19]	PL [22]	FSS [62]	SGL [29]	ST_SLRR [63]	DSSPL
1	0.6662±0.0718	0.7414±0.0653	0.7078±0.0608	0.7450±0.0576	0.7000±0.1787	0.8013±0.0912	0.7769±0.0480	0.8272±0.0749
2	0.5282±0.1606	0.4258±0.0459	0.7907±0.1194	0.7905±0.1154	0.7947±0.0826	0.8080±0.0585	0.8225±0.1192	0.8273±0.0509
3	0.5264±0.1406	0.7805±0.1541	0.9795±0.0257	0.9123±0.0598	0.9506±0.1027	0.9986±0.0019	0.9812±0.0243	0.9524±0.3424
4	0.6227±0.1229	0.9143±0.0246	0.5192±0.1345	0.5348±0.0955	0.3285±0.1200	0.6478±0.1248	0.5944±0.1624	0.8283±0.0610
5	0.9659±0.0580	0.9937±0.0044	0.8871±0.0026	0.8695±0.0286	0.9547±0.0749	0.9680±0.0038	0.9854±0.0050	0.9631±0.0300
6	0.5730±0.1285	0.4460±0.0646	0.9676±0.0309	0.9337±0.0861	0.9738±0.0194	0.7775±0.0851	0.9774±0.0292	0.9229±0.0003
7	0.8474±0.0838	0.9362±0.0268	0.8754±0.0631	0.8759±0.0178	0.9967±0.0070	1.0000±0.0000	0.8842±0.0958	0.8806±0.3001
8	0.6559±0.1267	0.7063±0.1303	0.8463±0.1587	0.8798±0.1598	0.8772±0.0276	0.7773±0.1492	0.7512±0.0140	0.8329±0.1557
9	0.9982±0.0013	0.9843±0.0107	0.9860±0.0062	0.9989±0.0000	0.3041±0.1080	0.9989±0.0000	0.9989±0.0000	0.9989±0.0000
AA	0.7315±0.0261	0.7698±0.0173	0.7744±0.0233	0.8078±0.0327	0.8378±0.0330	0.8653±0.0698	0.8769±0.0439	0.8937±0.0507
OA	0.6209±0.0588	0.5996±0.0189	0.7252±0.0520	0.7151±0.0451	0.8039±0.0449	0.8569±0.0368	0.8302±0.0386	0.8874±0.0231
Kappa	0.5335±0.0583	0.5213±0.0202	0.7774±0.0621	0.7641±0.0521	0.7774±0.0540	0.7608±0.0300	0.7884±0.0450	0.8505±0.0286

Fig. 7. Groundtruth map and classi cation maps on the University of Pavia dataset using 5 training examples of each class produced by eight classi cation methods.

and University of Pavia datasets, and higher than 90% for the Salinas dataset, with respect to $2 f 0:05; 0:1; 0:15; 0:2g$ and $2 f 0:001; 0:01; 0:1g$. It suggests that the classi cation performance is not very sensitive to the two hyperparameters. In this study, the coef cient is set to 0.01 for both the Indian Pines and University of Pavia datasets, 0.001 for the Salinas dataset, and is set to 0.1 for all three datasets.

F. Ablation Study

The proposed DSSPL consists of three key components for improving HSI classi cation performance, i.e., PL-based label propagation, dynamic graph fusion, and feedback strategy. To verify the effectiveness of the three key components, we compare the proposed DSSPL with different models in this ablation study. These compared models include: 1) Dynamic Spectral-Spatial GFHF (D-SS-GFHF), where GFHF is used instead of PL with the same dynamic spectral-spatial graph fusion and feedback strategy; 2) Dynamic Spectral PL (D-Spec-PL), in which PL is conducted on a single spectral graph with the feedback strategy; 3) Dynamic Spatial PL (D-Spat-PL), in which PL is conducted on a single spatial graph with the feedback strategy; 4) Spectral-Spatial PL (SS-PL),

TABLE V
OAS OF DIFFERENT METHODS ON THE INDIAN PINES, SALINAS AND UNIVERSITY OF PAVIA DATASETS WITH DIFFERENT NUMBERS OF TRAINING EXAMPLES OF EACH CLASS n^{tr}.

	n^{tr}	SVM [61]	CNN-PPF [10]	GFHF [19]	PL [22]	FSS [62]	SGL [29]	STSELWLR [63]	DSSPL
Indian Pines	3	0.4062±0.1053	0.2169±0.1122	0.6056±0.0561	0.6251±0.0718	0.7043±0.0245	0.6882±0.0845	0.7514±0.0792	0.7974±0.0514
	5	0.4879±0.0355	0.3087±0.1396	0.6499±0.0273	0.7060±0.0356	0.8142±0.0112	0.7270±0.0393	0.8052±0.0598	0.8581±0.0484
	10	0.5562±0.0349	0.6132±0.0435	0.7125±0.0315	0.7238±0.0367	0.8691±0.0126	0.7695±0.0268	0.8829±0.0217	0.8810±0.0359
	30/15*	0.6939±0.0122	0.7986±0.0139	0.8026±0.0169	0.7948±0.0256	0.8666±0.0073	0.8570±0.0144	0.9265±0.0156	0.9433±0.0130
Salinas	3	0.7091±0.0836	0.1168±0.0824	0.8124±0.0716	0.8347±0.6920	0.8528±0.0156	0.8831±0.0597	0.9259±0.0438	0.9368±0.1240
	5	0.7981±0.0267	0.3943±0.2196	0.9473±0.0100	0.9347±0.0129	0.9549±0.0074	0.9469±0.0122	0.9630±0.0387	0.9675±0.0178
	10	0.8863±0.0283	0.7765±0.0289	0.9547±0.0245	0.9412±0.0148	0.9789±0.0054	0.9617±0.0107	0.9745±0.0149	0.9789±0.0164
	30	0.8813±0.0210	0.9051±0.0037	0.9723±0.0086	0.9622±0.0050	0.9860±0.0029	0.9695±0.0083	0.9814±0.0087	0.9837±0.0041
University of Pavia	3	0.5584±0.0718	0.4780±0.0359	0.7015±0.1482	0.7098±0.0561	0.7482±0.0388	0.7581±0.0892	0.7931±0.0581	0.8119±0.0426
	5	0.6209±0.0588	0.5996±0.0189	0.7252±0.0520	0.7151±0.0451	0.8039±0.0449	0.8569±0.0368	0.8302±0.0357	0.8574±0.0231
	10	0.6831±0.0648	0.7417±0.0272	0.7524±0.0485	0.7418±0.0659	0.8517±0.0128	0.8826±0.0259	0.9218±0.0224	0.9242±0.0300
	30	0.8130±0.0223	0.8919±0.0154	0.8269±0.0395	0.8325±0.0577	0.9764±0.0131	0.9234±0.0140	0.9369±0.0187	0.9429±0.0094

* In the Indian Pines dataset, 15 pixels are selected from Classes 7 and 9 respectively, because they contain less than 30 pixels.

TABLE VI
OAS, AAS, AND KAPPA COEFFICIENTS OF DIFFERENT GRAPH UPDATING AND LABEL PROPAGATION METHODS ON THE INDIAN PINES, SALINAS AND UNIVERSITY OF PAVIA DATASETS WITH 5 TRAINING EXAMPLES OF EACH CLASS.

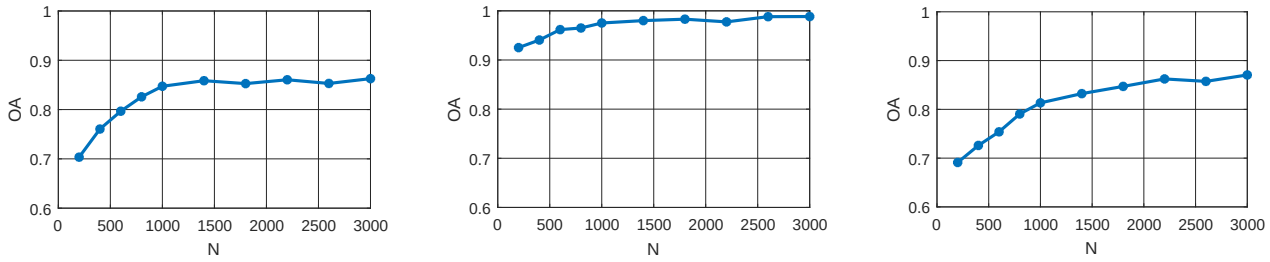
		D-SS-GFHF	D-Spec-PL	D-Spat-PL	SS-PL	DSSPL
Indian Pines	AA	0.8152±0.0435	0.7939±0.0481	0.8426±0.0282	0.8012±0.0458	0.8702±0.0312
	OA	0.8015±0.0394	0.7660±0.0531	0.8249±0.0321	0.7825±0.0625	0.8591±0.0484
	Kappa	0.7649±0.0398	0.7568±0.0614	0.7935±0.0518	0.7529±0.0709	0.8332±0.0535
Salinas	AA	0.9501±0.0254	0.9428±0.0265	0.9586±0.0219	0.9551±0.0267	0.9647±0.0220
	OA	0.9485±0.0204	0.9406±0.0319	0.9536±0.0351	0.9490±0.0236	0.9675±0.0178
	Kappa	0.9435±0.0288	0.9397±0.0281	0.9480±0.0262	0.9411±0.0207	0.9638±0.0199
University of Pavia	AA	0.8345±0.0479	0.8258±0.0518	0.8691±0.0620	0.8426±0.0728	0.8937±0.0507
	OA	0.8013±0.0341	0.8069±0.0465	0.8507±0.0422	0.8212±0.0423	0.8674±0.0231
	Kappa	0.7896±0.0382	0.7912±0.0346	0.8349±0.0395	0.8110±0.0403	0.8405±0.0286

TABLE VII
RUNNING TIME COMPARISON (IN SECONDS) AMONG DIFFERENT METHODS.

	SVM [61]	CNN-PPF [10]	GFHF [19]	PL [22]	FSS [62]	SGL [29]	STSELWLR [63]	DSSPL
Indian Pines	4.65	1495.12	6.00	6.18	46.30	12.43	15.26	6.25
Salinas	6.12	1769.52	29.83	30.10	106.27	48.06	61.49	35.89
University of Pavia	6.85	1545.26	112.63	125.14	197.14	174.28	185.56	130.26

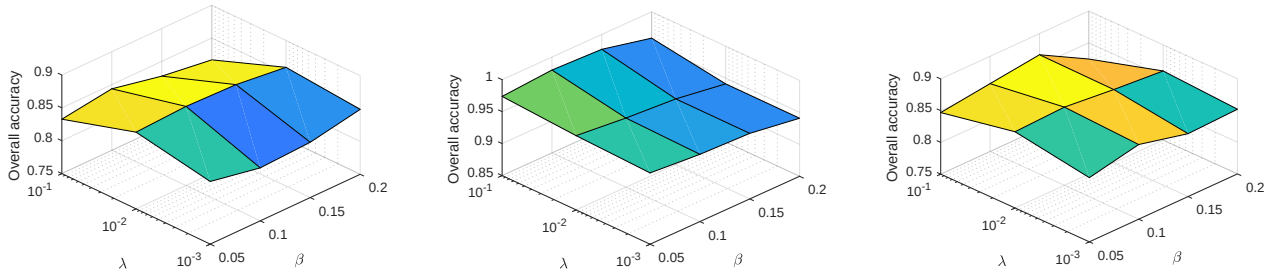
in which PL is conducted on spectral-spatial fusion graph without the feedback strategy. The settings of hyperparameters are the same with Table I. Table VI shows the comparison results of the ablation study on the three datasets. From the results, it can be observed that DSSPL achieves signi - cant improvements over D-SS-GFHF, which demonstrates the effectiveness of PL in handling the situation of extremely scarce labels. Moreover, DSSPL signi cantly improves the OAs, AAs, and kappa coef cients over D-Spec-PL, D-Spat-PL, and SS-PL models, indicating the effectiveness of our dynamic graph fusion and feedback strategy. Thus, the PL-based label propagation, dynamic graph fusion, and feedback strategy work collaboratively to improve the HSI classi cation performance in our DSSPL.

Fig. 8. Convergence of the proposed DSSPL on the Indian Pines, Salinas, and University of Pavia datasets.



(a) Indian Pines (b) Salinas (c) University of Pavia

Fig. 9. Sensitivity analysis on the number of superpixels (*i.e.*, N) on the Indian Pines, Salinas and University of Pavia datasets.



(a) Indian Pines (b) Salinas (c) University of Pavia

Fig. 10. Sensitivity study on the hyperparameters and on the Indian Pines, Salinas, and University of Pavia.

G. Running Time

To further investigate the efficiency of our DSSPL to the baseline methods, Table VII reports the running time of DSSPL and all the compared methods, where the number of labeled pixels is 5 for each class. Except for the experiments employing CNN-PPF, all the other compared methods were run under the same conditions using an Intel Core i7-9700 CPU and 16 GB of RAM. For CNN-PPF, the running time was reported on a server with a 3.60-GHz Intel Xeon CPU with 264 GB of RAM. Table VII shows that the running time of DSSPL is comparable to the other two traditional graph-based methods, *i.e.*, GFHF and PL, and is significantly less than CNN-PPF, FSS, SGL and STSE_DWLR. The high efficiency of our DSSPL is largely due to the superpixel-based graph construction and high convergence speed. Therefore, DSSPL shows both efficacy and efficiency among all the compared methods.

VI. Conclusion

In this paper, we propose a novel DSSPL framework to handle the challenge of extremely limited labeled data in HSI classification. The proposed DSSPL utilizes PL to remove the interference of the average constant output labels caused by extremely scarce examples, which can improve the efficiency of label propagation. Besides, graph fusion is conducted on spectral and spatial graphs to thoroughly describe spectral-spatial connective relationship of pixels. Further, the correlation of pixels can be computed using output labels and then fed back to update the fused graph, which comprehensively utilizes all the supervision information from labeled and unlabeled examples. Experimental results on three public datasets show the effectiveness of our proposed DSSPL in handling the situation of extremely scarce labels (*i.e.*, only 3, 5, or 10 pixels are labeled in each class). More importantly, the proposed DSSPL also achieves competitive results when more labeled examples are used in HSI classification.

References

[1] Y. Chen, Z. Lin, X. Zhao, G. Wang, and Y. Gu, "Deep learning-based classification of hyperspectral data," *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, vol. 7, no. 6, pp. 2094–2107, 2014.

[2] S. K. Roy, G. Krishna, S. R. Dubey, and B. B. Chaudhuri, "HybridSN: Exploring 3-D-2-D CNN feature hierarchy for hyperspectral image classification," *IEEE Geoscience and Remote Sensing Letters*, vol. 17, no. 2, pp. 277–281, 2019.

[3] X. Mei, E. Pan, Y. Ma, X. Dai, J. Huang, F. Fan, Q. Du, H. Zheng, and J. Ma, "Spectral-spatial attention networks for hyperspectral image classification," *Remote Sensing*, vol. 11, no. 8, p. 963, 2019.

[4] H. Sun, X. Zheng, X. Lu, and S. Wu, "Spectral-spatial attention network for hyperspectral image classification," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 58, no. 5, pp. 3232–3245, 2019.

[5] H. Sun, X. Zheng, and X. Lu, "A supervised segmentation network for hyperspectral image classification," *IEEE Transactions on Image Processing*, vol. 30, pp. 2810–2825, 2021.

[6] Y. Bengio, *Learning deep architectures for AI*. Now Publishers Inc., 2009.

[7] B. Neyshabur, S. Bhojanapalli, D. McAllester, and N. Srebro, "Exploring generalization in deep learning," *arXiv preprint arXiv:1706.08947*, 2017.

[8] M. Belkin, D. Hsu, S. Ma, and S. Mandal, "Reconciling modern machine-learning practice and the classical bias–variance trade-off," *Proceedings of the National Academy of Sciences*, vol. 116, no. 32, pp. 15 849–15 854, 2019.

[9] M. E. Paoletti, J. M. Haut, J. Plaza, and A. Plaza, "A new deep convolutional neural network for fast hyperspectral image classification," *ISPRS Journal of Photogrammetry and Remote Sensing*, vol. 145, pp. 120–147, 2018.

[10] W. Li, G. Wu, F. Zhang, and Q. Du, "Hyperspectral image classification using deep pixel-pair features," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 55, no. 2, pp. 844–853, 2016.

[11] S. Zhong, Y. Zhang, and C.-I. Chang, "A spectral–spatial feedback close network system for hyperspectral image classification," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 57, no. 12, pp. 10 056–10 069, 2019.

[12] Y. Duan, H. Huang, and Y. Tang, "Local constraint-based sparse manifold hypergraph learning for dimensionality reduction of hyperspectral image," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 59, no. 1, pp. 613–628, 2020.

[13] Z. Li, H. Huang, Y. Duan, and G. Shi, "DLPNet: A deep manifold network for feature extraction of hyperspectral imagery," *Neural Networks*, vol. 129, pp. 7–18, 2020.

[14] H. Huang, Z. Li, H. He, Y. Duan, and S. Yang, "Self-adaptive manifold discriminant analysis for feature extraction from hyperspectral imagery," *Pattern Recognition*, vol. 107, p. 107487, 2020.

[15] G. Camps-Valls, T. V. B. Marsheva, and D. Zhou, "Semi-supervised graph-based hyperspectral image classification," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 45, no. 10, pp. 3044–3054, 2007.

[16] A. Ma, Y. Zhong, B. Zhao, H. Jiao, and L. Zhang, "Semisupervised subspace-based dna encoding and matching classifier for hyperspectral remote sensing imagery," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 54, no. 8, pp. 4402–4418, 2016.

[17] L. Liu, W. Huang, B. Liu, L. Shen, and C. Wang, "Semisupervised hyperspectral image classification via laplacian least squares support vector machine in sum space and random sampling," *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, vol. 11, no. 11, pp. 4086–4100, 2018.

[18] C. Zhang, J. Yue, and Q. Qin, "Global prototypical network for few-shot hyperspectral image classification," *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, vol. 13, pp. 4748–4759, 2020.

[19] X. Zhu, Z. Ghahramani, and J. D. Lafferty, "Semi-supervised learning using gaussian fields and harmonic functions," in *Proceedings of the International Conference on Machine Learning (ICML)*, 2003, pp. 912–919.

[20] D. Slepcev and M. Thorpe, "Analysis of p-laplacian regularization in semisupervised learning," *SIAM Journal on Mathematical Analysis*, vol. 51, no. 3, pp. 2085–2120, 2019.

[21] J. Calder, "The game theoretic p-laplacian and semi-supervised learning with few labels," *Nonlinearity*, vol. 32, no. 1, pp. 301–330, 2018.

[22] J. Calder, B. Cook, M. Thorpe, and D. Slepcev, "Poisson learning: Graph based semi-supervised learning at very low label rates," in *International Conference on Machine Learning*. PMLR, 2020, pp. 1306–1316.

[23] G. Camps-Valls, T. V. Bandos Marsheva, and D. Zhou, "Semi-supervised graph-based hyperspectral image classification," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 45, no. 10, pp. 3044–3054, 2007.

[24] W. Li, J. Liu, and Q. Du, "Sparse and low-rank graph for discriminant analysis of hyperspectral imagery," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 54, no. 7, pp. 4094–4105, 2016.

[25] Z. Xue, P. Du, J. Li, and H. Su, "Simultaneous sparse graph embedding for hyperspectral image classification," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 53, no. 11, pp. 6114–6133, 2015.

[26] W. Liao, M. Dalla Mura, J. Chanussot, and A. Pižurica, "Fusion of spectral and spatial information for classification of hyperspectral remote-sensed imagery by local graph," *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, vol. 9, no. 2, pp. 583–594, 2016.

[27] Q. Liu, Y. Sun, R. Hang, and H. Song, "Spatial–spectral locality-constrained low-rank representation with semi-supervised hypergraph learning for hyperspectral image classification," *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, vol. 10, no. 9, pp. 4171–4182, 2017.

[28] H. Wu and S. Prasad, "Semi-supervised deep learning using pseudo labels for hyperspectral image classification," *IEEE Transactions on Image Processing*, vol. 27, no. 3, pp. 1259–1270, 2018.

[29] P. Sellars, A. I. Aviles-Rivero, and C. Schönlieb, "Superpixel contracted graph-based learning for hyperspectral image classification," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 58, no. 6, pp. 4180–4193, 2020.

[30] S. Wan, C. Gong, P. Zhong, B. Du, L. Zhang, and J. Yang, "Multiscale dynamic graph convolutional network for hyperspectral image classification," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 58, no. 5, pp. 3162–3177, 2020.

[31] S. Wan, C. Gong, P. Zhong, S. Pan, G. Li, and J. Yang, "Hyperspectral image classification with context-aware dynamic graph convolutional network," *IEEE Transactions on Geoscience and Remote Sensing*, pp. 1–16, 2020.

[32] S. Wan, S. Pan, P. Zhong, X. Chang, J. Yang, and C. Gong, "Dual interactive graph convolutional networks for hyperspectral image classification," *IEEE Transactions on Geoscience and Remote Sensing*, pp. 1–14, 2021.

[33] H. Yin, A. R. Benson, J. Leskovec, and D. F. Gleich, "Local higher-order graph clustering," in *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2017, pp. 555–564.

[34] S. Cao, W. Lu, and Q. Xu, "Deep neural networks for learning graph representations," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 30, no. 1, 2016.

[35] L. Qiao, L. Zhang, S. Chen, and D. Shen, "Data-driven graph construction and graph learning: A review," *Neurocomputing*, vol. 312, pp. 336–351, 2018.

[36] F. Nie, D. Xu, I. W.-H. Tsang, and C. Zhang, "Flexible manifold embedding: A framework for semi-supervised and unsupervised dimension reduction," *IEEE Transactions on Image Processing*, vol. 19, no. 7, pp. 1921–1932, 2010.

[37] C. Gong, T. Liu, D. Tao, K. Fu, E. Tu, and J. Yang, "Deformed graph laplacian for semisupervised learning," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 26, no. 10, pp. 2261–2274, 2015.

[38] C. Gong, D. Tao, W. Liu, L. Liu, and J. Yang, "Label propagation via teaching-to-learn and learning-to-teach," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 28, no. 6, pp. 1452–1465, 2017.

[39] M. Gori, G. Monfardini, and F. Scarselli, "A new model for learning in graph domains," in *IEEE International Joint Conference on Neural Networks*, vol. 2, 2005, pp. 729–734.

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

[41] T. N. Kipf and M. Welling, "Semi-supervised classification with graph convolutional networks," *arXiv preprint arXiv:1609.02907*, 2016.

[42] M. Defferrard, X. Bresson, and P. Vandergheynst, "Convolutional neural networks on graphs with fast localized spectral filtering," *arXiv preprint arXiv:1606.09375*, 2016.

[43] W. L. Hamilton, R. Ying, and J. Leskovec, "Inductive representation learning on large graphs," *arXiv preprint arXiv:1706.02216*, 2017.

[44] K. Xu, H. Huang, P. Deng, and Y. Li, "Deep feature aggregation framework driven by graph convolutional network for scene classification in remote sensing," *IEEE Transactions on Neural Networks and Learning Systems*, 2021.

[45] X. Cai, F. Nie, W. Cai, and H. Huang, "Heterogeneous image features integration via multi-modal semi-supervised learning model," in *Proceedings of the IEEE International Conference on Computer Vision*, 2013, pp. 1737–1744.

[46] M. Karasuyama and H. Mamitsuka, "Multiple graph label propagation by sparse integration," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 24, no. 12, pp. 1999–2012, 2013.

[47] C. Gong, D. Tao, S. J. Maybank, W. Liu, G. Kang, and J. Yang, "Multi-modal curriculum learning for semi-supervised image classification," *IEEE Transactions on Image Processing*, vol. 25, no. 7, pp. 3249–3260, 2016.

[48] C. Gong, J. Yang, and D. Tao, "Multi-modal curriculum learning over graphs," *ACM Transactions on Intelligent Systems and Technology (TIST)*, vol. 10, no. 4, pp. 1–25, 2019.

[49] R. Achanta, A. Shaji, K. Smith, A. Lucchi, P. Fua, and S. Süsstrunk, "Slic superpixels compared to state-of-the-art superpixel methods," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 34, no. 11, pp. 2274–2282, 2012.

[50] S. Jia, B. Deng, J. Zhu, X. Jia, and Q. Li, "Superpixel-based multitask learning framework for hyperspectral image classification," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 55, no. 5, pp. 2575–2588, 2017.

[51] Y. Li, T. Lu, and S. Li, "Subpixel-pixel-superpixel-based multiview active learning for hyperspectral images classification," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 58, no. 7, pp. 4976–4988, 2020.

[52] P. Yang, L. Tong, B. Qian, Z. Gao, J. Yu, and C. Xiao, "Hyperspectral image classification with spectral and spatial graph using inductive representation learning network," *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, vol. 14, pp. 791–800, 2021.

[53] B. Nadler, N. Srebro, and X. Zhou, "Semi-supervised learning with the graph laplacian: The limit of infinite unlabelled data," *Advances in Neural Information Processing Systems*, vol. 22, pp. 1330–1338, 2009.

[54] H. Jiang, Y. Song, C. Wang, M. Zhang, and Y. Sun, "Semi-supervised learning over heterogeneous information networks by ensemble of meta-