# Cross-Scene Hyperspectral Image Classification Based on Fusion of Active Learning Strategies

Yue Zhang, Mengjiao Tang, Yuyu Fu, Yao Rong, Sanfeng Hu
*Yunnan Key Laboratory of Statistical Modeling and Data Analysis, Yunnan University*
650504 Kunming, China
zhangyue_ym7y@stu.ynu.edu.cn; qingqing_tmj@126.com; fuyuyu@stu.ynu.edu.cn;
yaorong0618@outlook.com; sanfenghu@outlook.com

***Abstract*—This paper proposes a novel active learning (AL)-incorporated cross-scene hyperspectral image (HSI) classification method, which combines a graph convolutional network (GCN)-based classifier with an adaptive fusion framework for multiple AL selection criteria. Specifically, we extend the existing dynamic multiscale GCN from single-scene to cross-scene classification by introducing a feature alignment module and a cross-scene loss function to enhance domain adaptation. Additionally, our AL module fuses multiple widely used sampling criteria through score-to-probability transformation and categorical distribution fusion, in order to mitigate biases from individual strategies and leverage their complementary strengths. This enables the model to refine its target-scene training set, improving classification performance with minimal labeling effort. Experiments on the Pavia dataset demonstrate that the proposed method outperforms several recent cross-scene HSI classification approaches.**

***Index Terms*—Cross-scene classification, hyperspectral image (HSI) classification, active learning (AL), fusion of active learning strategies.**

## I. Introduction

Cross-scene hyperspectral image (HSI) classification has attracted increasing interest in recent years [1]–[3]. It reduces the significant costs of labeling high-dimensional HSI samples across different scenes. The core idea of cross-scene classification is to leverage the relatively abundant labeled samples from a source scene to assist in classifying a newly acquired, similar target scene with very few or no labeled samples [1].

With the rapid advancement of deep learning, models such as convolutional neural networks (CNNs), deep neural networks, and graph convolutional networks (GCNs) have been extensively applied to HSI classification [4]–[6]. However, compared to single-scene classification, cross-scene classification faces additional challenges due to scene dissimilarities caused by variations in sensor acquisition, geographical locations, and weather conditions [7]. These dissimilarities necessitate the use of transfer learning techniques to mitigate domain shifts and enhance classification performance. Consequently, cross-scene classification is often framed as a transfer learning problem, as it involves transferring knowledge from a source scene to a target scene [8].

This work was supported in part by the National Natural Science Foundation of China under Grant 62263032 and 12231017, in part by the Youth Project of the "Xingdian" Talent Support Plan of Yunnan Province, and in part by the Fundamental Research Project of Yunnan Province under Grant 202301AU070213. *(Corresponding author: Mengjiao Tang)*

Domain Adaptation (DA), a form of transductive transfer learning [9], is commonly used to reduce domain bias between source and target scenes. Various DA methods have been integrated with deep learning models to improve cross-scene classification accuracy. For instance, the deep cross-domain few-shot learning (DCFSL) method [10] employed a conditional adversarial DA strategy to learn a domain-adaptive feature embedding space, ensuring better alignment between source and target distributions. Similarly, [11] proposed a class-covariance-based few-shot learning (CMFSL) framework, which leverages a spectral-prior-based refinement module to further minimize domain shifts. In [7], a dual-stream discriminative attention network (DSDAN) was proposed, utilizing a lightweight hybrid CNN with dual-stream processing to extract spatial–spectral features from both domains, while a discriminative attention block enhances domain alignment.

Besides the development of cross-scene classifiers, identifying salient target samples for labeling when exploring a new scene is also crucial for improving model performance. Active learning (AL) is an iterative process that selects the most informative examples from a pool of unlabeled samples. In cross-scene classification, AL strategies can leverage knowledge from a similar source scene to identify the most relevant target samples for annotation, thereby improving classification accuracy in both scenes while minimizing labeling costs. Despite its success in single-scene HSI classification [12]–[14], the application of AL to cross-scene HSI classification remains limited. In [15], a salient sample query (SSQ) AL process was introduced to query and identify salient samples in both the source and target domains, which were subsequently used to construct a deep mapping network for improving classification performance.

In this paper, we propose a novel AL-incorporated cross-scene classification method that employs a newly devised cross-scene GCN as the base classifier. Unlike existing AL methods, our AL module fuses decision results from multiple classical AL strategies, including entropy-based, confidence-based, margin-based, and cluster-based sampling, through probability transformation and distribution fusion, while adaptively balancing the contributions of different strategies to ensure a more robust and adaptive selection process. Additionally, in the base classifier, we extend the existing dynamic multiscale GCN (DMS-GCN) from single-scene to cross-scene

classification by introducing a feature alignment module and a cross-scene loss function. While graph-based learning captures spatial–spectral relationships to enhance feature representation and domain adaptation, active learning enriches the training set by selecting informative target-scene samples, improving classification performance with minimal labeling effort.

The rest of the paper the proposed cross-scene GCN classification method with an integrated AL module (Section II), elaborates the fusion method of multiple AL strategies (Section III), presents experimental results (Section IV), and concludes in Section V.

## II. Proposed Method

Throughout this paper, symbols with subscripts $s$ and $t$ represent samples or statistics from the source and target scenes, respectively. Let the source scene data be denoted as $\mathcal{X}_s = \{\mathbf{x}_s^i\}_{i=1}^{n_s} \subseteq \mathbb{R}^d$, which contains both labeled and unlabeled samples, and the target scene data as $\mathcal{X}_t = \{\mathbf{x}_t^i\}_{i=1}^{n_t} \subseteq \mathbb{R}^d$, where $n_s$ and $n_t$ represent the number of source and target samples, respectively, and $d$ denotes the data dimension. In cross-scene HSI classification, labeled samples from the target scene are often scarce (or even absent), whereas those from the source scene are relatively abundant. The objective is to predict the labels for the unlabeled portion of $\mathcal{X}_t$.

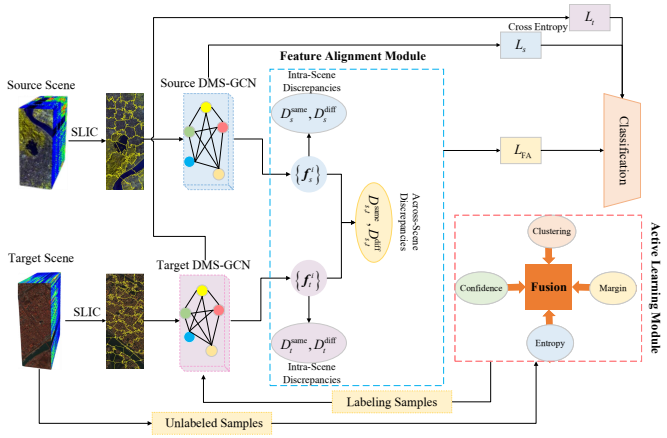



Fig. 1. Architecture of the proposed CSGCN-FAL for cross-scene HSI classification

The architecture of the proposed network for cross-scene HSI classification is illustrated in Fig. 1. It is built upon the DMS-GCN model, originally designed for single-scene semi-supervised hyperspectral image classification [16]. To extend its applicability to cross-scene classification, we introduce several key modifications. First, a feature alignment module is incorporated to mitigate feature distribution differences between the source and target scenes, ensuring robust performance across scenes. Second, a cross-scene loss Function is specifically designed to optimize classification performance in the cross-scene scenario. Finally, an AL module adaptively selects the most representative samples for annotation, thereby improving classification accuracy under limited labeled data constraints.

### A. Feature Extraction Module

This module aims to extract relevant features from the source- and target-scene samples. Specifically, the DMS-GCN model proposed in [16] is employed to separately extract features from the source and target scene samples. Prior to this, the Simple Linear Iterative Clustering (SLIC) algorithm is applied to generate an initial region map, which reduces the number of nodes and improves computational efficiency. Let the output of the layer right before the final full connection layer of the DMS-GCN model correspond to feature maps $\varphi_s$ and $\varphi_t$ for the source and target scenes, respectively. The extracted features for each pixel are then obtained as:

$$[\boldsymbol{f}_s^1, \ldots, \boldsymbol{f}_s^{n_s}] = \varphi_s(\boldsymbol{x}_s^1, \ldots, \boldsymbol{x}_s^{n_s}), \tag{1}$$

$$[\boldsymbol{f}_t^1, \ldots, \boldsymbol{f}_t^{n_t}] = \varphi_t(\boldsymbol{x}_t^1, \ldots, \boldsymbol{x}_t^{n_t}). \tag{2}$$

### B. Feature Alignment Module

To achieve effective cross-scene classification, it is necessary to impose constraints on the extracted features to reduce domain differences between the two scenes. Additionally, within each scene, further feature alignment enhances model stability. Here, we align the extracted features both within a single scene and across different scenes, following a widely used principle in machine learning and pattern recognition: intra-class feature similarity should be maximized, while inter-class feature separability should be maintained. To this end, this module first computes the feature discrepancy terms.

Suppose the data from $\mathcal{X}_s$ and $\mathcal{X}_t$ belong to $C$ categories. Let $\mathcal{C}_s$ and $\mathcal{C}_t$ denote the sets of indices of the labeled data in the source and target scenes, respectively, with $\mathcal{C}_s^k$ and $\mathcal{C}_t^k$ representing the subsets of indices corresponding to samples labeled as class $k$, for $k = 1, \ldots, C$. By definition, $\bigcup_{k=1}^{C} \mathcal{C}_s^k = \mathcal{C}_s$ and $\bigcup_{k=1}^{C} \mathcal{C}_t^k = \mathcal{C}_t$. The intra-class discrepancies within each scene and across scenes are measured by

$$D_s^{\text{same}} = \sum_{k=1}^{C} \sum_{i,j \in \mathcal{C}_s^k} \left\| \boldsymbol{f}_s^i - \boldsymbol{f}_s^j \right\|_2, \tag{3}$$

$$D_t^{\text{same}} = \sum_{k=1}^{C} \sum_{i,j \in \mathcal{C}_t^k} \left\| \boldsymbol{f}_t^i - \boldsymbol{f}_t^j \right\|_2, \tag{4}$$

$$D_{s,t}^{\text{same}} = \sum_{k=1}^{C} \sum_{i \in \mathcal{C}_s^k} \sum_{j \in \mathcal{C}_t^k} \left\| \boldsymbol{f}_s^i - \boldsymbol{f}_t^j \right\|, \tag{5}$$

$$D^{\text{same}} = D_s^{\text{same}} + D_t^{\text{same}} + D_{s,t}^{\text{same}} \tag{6}$$

where

- $D_s^{\text{same}}$ (respectively, $D_t^{\text{same}}$) measures the intra-class feature discrepancy among source (respectively, target) scene samples.
- $D_{s,t}^{\text{same}}$ measures the cross-scene intra-class feature discrepancy between source and target samples;
- $D^{\text{same}}$ is the total intra-class discrepancy across both scenes.

Similarly, the total inter-class discrepancy (both within and across scenes) is computed as

$$D^{\text{diff}} = D_s^{\text{diff}} + D_t^{\text{diff}} + D_{s,t}^{\text{diff}}, \tag{7}$$

where $D_s^{\text{diff}} = \sum_{k\neq\ell}\sum_{i\in\mathcal{C}_s^k}\sum_{j\in\mathcal{C}_s^\ell}\left\|\boldsymbol{f}_s^i - \boldsymbol{f}_s^j\right\|_2$, $D_t^{\text{diff}} = \sum_{k\neq\ell}\sum_{i\in\mathcal{C}_t^k}\sum_{j\in\mathcal{C}_t^\ell}\left\|\boldsymbol{f}_t^i - \boldsymbol{f}_t^j\right\|_2$, and $D_{s,t}^{\text{diff}} = \sum_{k\neq\ell}\sum_{i\in\mathcal{C}_s^k}\sum_{j\in\mathcal{C}_t^\ell}\left\|\boldsymbol{f}_s^i - \boldsymbol{f}_t^j\right\|$. The feature alignment loss is then formulated as:

$$L_{\text{FA}} = \frac{D^{\text{same}}}{D^{\text{diff}}}, \tag{8}$$

which enforces inter-class distinctiveness and intra-class consistency, ensuring proper feature alignment.

It is worth mentioning that this module requires a small number of labeled target scene samples, and thus does not apply when the target scene has no labeled samples in some classes.

### C. Overall Loss Function

For the two integrated single-scene DMS-GCN subnetworks, let the final softmax layers produce outputs $\boldsymbol{p}_s^i$ and $\boldsymbol{p}_t^i$, which denote the predicted probability distributions for the $i$-th source and target samples, respectively. The cross-entropy loss functions for the source and target scenes are then given by

$$L_s = -\frac{1}{n_s}\sum_{i\in\mathcal{C}_s}(\boldsymbol{y}_s^i)^{\mathsf{T}}\log\boldsymbol{p}_s^i, \quad L_t = -\frac{1}{n_t}\sum_{i\in\mathcal{C}_t}(\boldsymbol{y}_t^i)^{\mathsf{T}}\log\boldsymbol{p}_t^i,$$

where $\boldsymbol{y}_s^i$ and $\boldsymbol{y}_t^i$ represent the true one-hot class labels. The overall loss function is:

$$L = L_s + L_t + \lambda L_{\text{FA}}, \tag{9}$$

where $\lambda$ is a hyperparameter used to balance classification and domain alignment.

### D. Active Learning Module

The goal of the AL module is to select a small subset of highly informative target-scene samples, $\boldsymbol{x}_t^i, i \notin \mathcal{C}_t$, for annotation. These samples are identified through a fusion of multiple AL criteria, including entropy, confidence, margin, and clustering, as described in the next section. Newly labeled samples are then added to the training set, improving efficiency and classification accuracy.

## III. Fusion Of Active Learning Strategies

### A. Commonly Used Selection Strategies

Initially, there are $n_t^u \triangleq n_t - |\mathcal{C}_t|$ unlabeled samples in the target scene. For each candidate sample in $\{\boldsymbol{x}_t^i\}_{i\notin\mathcal{C}_t}$, the proposed network from the previous section, after initial training, outputs a probability distribution over possible labels, i.e., $\boldsymbol{p}_t^i$. Several commonly used selection strategies in the literature rank these candidates for annotation, including:

- Entropy-based Sampling. The entropy of the predicted probability distribution $\boldsymbol{p}_t^i$ is $H(\boldsymbol{x}_t^i) = -(\boldsymbol{p}_t^i)^{\mathsf{T}}\log\boldsymbol{p}_t^i,\ \forall i \notin \mathcal{C}_t$. Higher entropy indicates greater uncertainty in the model's prediction. Thus, this strategy prioritizes samples with the highest entropy values.
- Confidence-based Sampling. The confidence $\text{Conf}(\boldsymbol{x}_t^i)$ is defined as the maximum probability in $\boldsymbol{p}_t^i$. Selecting samples with the lowest confidence encourages the model to learn in its least confident regions, thereby improving sensitivity to classification boundaries.
- Margin Sampling. The margin $\text{Marg}(\boldsymbol{x}_t^i)$ is defined as the difference between the highest and second-highest predicted probabilities in $\boldsymbol{p}_t^i$. Selecting samples with the smallest margin targets those near the classification boundary, which are typically the most challenging to classify.
- Clustering-based Sampling. An unsupervised clustering method (e.g., K-means) is used to partition the unlabeled samples into $C$ clusters, $\mathcal{A}_1, \ldots, \mathcal{A}_C$. Samples are then ranked based on their Euclidean distance from the centroid of their assigned cluster, with those farthest from the centroid prioritized for selection: $\text{dist}(\boldsymbol{x}_t^i) = \sum_{k=1}^{C}\delta(\boldsymbol{x}_t^i \in \mathcal{A}_k)\|\boldsymbol{x}_t^i - \boldsymbol{\mu}_k\|_2$, where $\boldsymbol{\mu}_k$ represents the centroid of cluster $\mathcal{A}_k$, and $\delta$ is the indicator function. This strategy enhances diverse coverage of the entire data space and helps avoid bias toward any particular region.

### B. Fusion Method

Each of the aforementioned four selection strategies evaluates the importance of sample from a different perspective, reflecting different preferences in selection. By integrating these strategies, we can leverage their complementary strengths and mitigate biases from individual strategies, resulting in a more balanced and robust sample selection process.

Instead of directly fusing the final selection results, we combine selection decisions from each strategy according to the following two steps:

*1) Selection Score Transformation:* Each strategy evaluates a sample based on its selection score, defined as

$$S_\ell(\boldsymbol{x}_t^i) = \begin{cases} H(\boldsymbol{x}_t^i), & \ell = 1, \\ 1 - \text{Conf}(\boldsymbol{x}_t^i), & \ell = 2, \\ 1 - \text{Marg}(\boldsymbol{x}_t^i), & \ell = 3, \\ \text{dist}(\boldsymbol{x}_t^i), & \ell = 4, \end{cases} \tag{10}$$

This transformation ensures higher scores correspond to higher selection preference for all strategies. To enhance robustness, for strategy $\ell$, we retain only the top $m_\ell$ highest-scoring samples and set all other scores to zero. The adjusted scores are then converted into selection probability distributions, denoted as[1] $\boldsymbol{q}_\ell = (q_{\ell,i}) \in \Delta^{n_t^u-1}$, using the softmax function:

$$q_{\ell,i} = \frac{e^{S_\ell(\boldsymbol{x}_t^i)}}{\sum_i e^{S_\ell(\boldsymbol{x}_t^i)}}, \quad i = 1, \ldots, n_t^u. \tag{11}$$

*2) Fusion of the Selection Probabilities:* The final selection probability is obtained by optimally fusing the four individual probability distributions $\boldsymbol{p}_\ell, \ell = 1, 2, 3, 4$. We define the fused probability $\boldsymbol{q}$ as the solution to the following optimization problem:

$$\min_{\substack{\boldsymbol{q}\in\Delta^{n_t^u-1} \\ \{w_\ell\}_{\ell=1}^4}} \sum_{\ell=1}^{4} w_\ell D(\boldsymbol{q}_\ell, \boldsymbol{q}), \quad \text{s.t. } \sum_\ell w_\ell = 1, w_\ell > 0, \tag{12}$$

[1] $\Delta^n$ denotes the probability simplex in $\mathbb{R}^n$:$\Delta^n = \{\boldsymbol{p} = (p_i) \in \mathbb{R}^n : \sum_{i=1}^n p_i = 1, 0 \le p_i \le 1\}$.

where:

- $\boldsymbol{q} = (q_1, \ldots, q_{n_t^u})$ is the fused selection probability.
- $w_\ell$ is the weight assigned to $\boldsymbol{q}_\ell$, indicating the relative importance of the $\ell$-th strategy. To ensure flexibility, the weights $\{w_\ell\}$ can be predefined based on prior knowledge or expert input. In this paper, we adaptively optimize them using the same objective function to avoid manual tuning.
- $D(\boldsymbol{q}_\ell, \boldsymbol{q})$ is a metric function that measures the dissimilarity from each individual probability $\boldsymbol{q}_\ell$ to the fused probability $\boldsymbol{q}$.

We examine two different metrics for $D$ in Problem (12):

- Squared Euclidean Distance: $D(\boldsymbol{q}_\ell, \boldsymbol{q}) = \|\boldsymbol{q}_\ell - \boldsymbol{q}\|^2$, which is a classical measure in vector spaces. When the weights are fixed, the fused probability $\boldsymbol{q}$ corresponds to the weighted arithmetic average probability, a well-established method for fusing discrete (categorical) distributions [17]. This fusion rule is also referred to as the "Opinion Pool," used to reach an agreement on multiple opinions [18].
- $\chi^2$-Divergence:

$$D(\boldsymbol{q}_\ell, \boldsymbol{q}) = \sum_{i=1}^{n_t^u} \frac{(q_{\ell,i})^2}{q_i} - 1, \tag{13}$$

  which measures the dissimilarity between two probability distributions in the probability simplex rather than in the Euclidean space. It is worth noting that $\chi^2$-divergence is not a true distance metric in the rigorous mathematical sense, as it does not satisfy the symmetry and triangle inequality.

### C. Alternating Optimization Algorithm for the Fusion

The objective function in (12) is non-convex jointly in $\boldsymbol{q}$ and $w_\ell$, but it is convex and easy to solve when one variable is fixed while optimizing the other. Thus, an alternating minimization approach can be employed to solve (12). At iteration step $t$, the optimization proceeds as follows (where the superscript $(t)$ denotes estimators at this step):

*1) With Fixed Weights $\{w_\ell^{(t-1)}\}$:* Since $D$ is convex in $\boldsymbol{q}$, the optimal $\boldsymbol{q}$ can be explicitly derived. When $D$ is the squared Euclidean distance, the optimal $\boldsymbol{q}$ corresponds to the weighted arithmetic mean:

$$\boldsymbol{q}^{(t)} = \sum_{\ell=1}^{4} w_\ell^{(t-1)} \boldsymbol{q}_\ell; \tag{14}$$

When $D$ is the $\chi^2$-divergence, by the Lagrange method,

$$q_i^{(t)} = \frac{1}{A} \sqrt{\sum_{\ell=1}^{4} w_\ell^{(t)} (q_{\ell,i})^2}, \ i = 1, \ldots, n_t^u. \tag{15}$$

where $A = \sum_{i=1}^{n_\ell^u} \sqrt{\sum_{\ell=1}^{4} w_\ell^{(t)} (q_{\ell,i})^2}$ is the normalizing factor.

*2) With Fixed Fused Probability $\boldsymbol{q}^{(t)}$:* Given the fused probability, the optimization of the weights reduces to a linear programming problem

$$\{w_\ell^{(t)}\} = \arg\min_{w_\ell} \sum_{\ell=1}^{4} d_\ell^{(t)} w_\ell, \text{ s.t. } \sum_{\ell} w_\ell = 1, w_\ell > 0, \tag{16}$$

where $d_\ell^{(t)} = D(\boldsymbol{q}_\ell, \boldsymbol{q}^{(t)})$, for $\ell = 1, 2, 3, 4$, is computed using either the squared Euclidean distance or the $\chi^2$-divergence. This linear program can be efficiently solved using methods such as the simplex algorithm.

The algorithm iteratively updates (14) and (16) (or alternatively, (15) and (16)) until convergence, yielding a solution for (12). In our numerical experiments, the algorithm demonstrates good empirical convergence performance.

## IV. Experiments

To evaluate the performance of the proposed method, we conduct experiments using the RPaviaU-DPaviaC dataset, which includes the ROSIS Pavia University (RPaviaU) scene and the DAIS Pavia Center (DPaviaC) scene. These two scenes share a common set of seven land cover classes. Our proposed method, Cross-Scene multiscale GCN with a Fusion of Active Learning strategies (CSGCN-FAL), is evaluated with two variants: one using the squared Euclidean distance (CSGCN-FAL1) and the other using the $\chi^2$-divergence (CSGCN-FAL2).

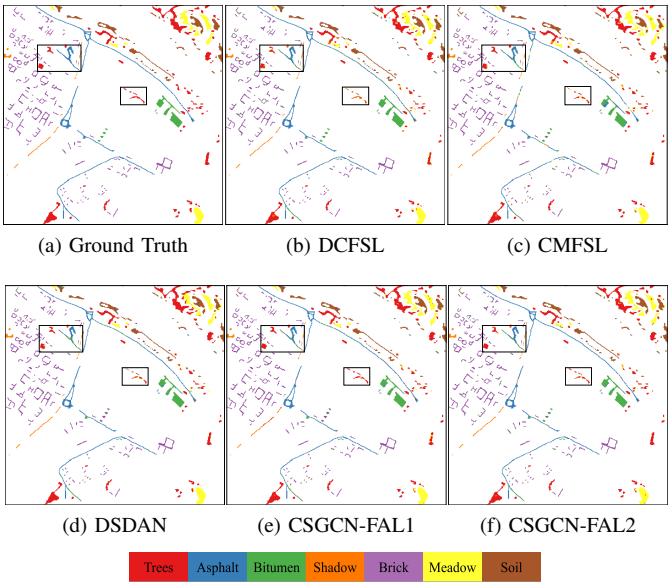
For comparative analysis, we consider several recent cross-scene HSI classification methods, including DCFSL [10], CMFSL [11], and DSDAN [7]. To ensure a fair evaluation and save space, we do not include single-scene HSI classification methods, as prior research has demonstrated that such methods generally underperform compared to those specifically designed for cross-scene classification.

To reduce computational cost, in the experiments, we randomly select 50 labeled training samples per class in the source scene. For the comparative methods (which do not use AL), only 5 labeled training samples per class are randomly selected in the target scene. For our proposed method (CSGCN-FAL1 and CSGCN-FAL2) with AL, we initially select 2 samples per class at random and then add 21 additional samples across all classes through AL. Specifically, in the AL loop, we perform a total of 21 rounds, each querying one unlabeled sample from the target scene for annotation. It is worth noting that under this setting, all considered methods use the same final number of labeled training samples in both scenes. Each experiment is repeated ten times, and the average accuracies are recorded. All implementations are conducted using Python 3.8 with PyTorch 1.11.0.

Table I presents the classification performance of all considered methods, with the highest values in bold and the second-highest values underlined. The corresponding classification maps (randomly chosen from one of the ten experiments) are shown in Fig. 2. CSGCN-FAL2 consistently achieves the highest overall accuracy (OA), average accuracy (AA), and Kappa coefficient. CSGCN-FAL1 follows closely but slightly underperforms compared to CSGCN-FAL2. This performance

TABLE I
CLASS-WISE CLASSIFICATION ACCURACIES AND OVERALL METRICS FOR DIFFERENT METHODS ON THE RPAVIAU-DPAVIAC DATASET

| Class | DCFSL [10] | CMFSL [11] | DSDAN [7] | CSGCN-FAL1 | CSGCN-FAL2 |
|---|---|---|---|---|---|
| 1 | 0.8702 | 0.9226 | 0.9181 | **0.9397** | 0.9289 |
| 2 | 0.8664 | 0.8397 | 0.8658 | 0.8741 | **0.9151** |
| 3 | **0.9534** | 0.8211 | 0.8647 | 0.8827 | 0.9248 |
| 4 | 0.9863 | 0.9819 | 0.9683 | 0.8507 | **0.9864** |
| 5 | 0.9292 | 0.9202 | **0.9612** | 0.9562 | 0.9197 |
| 6 | 0.9602 | **0.9943** | 0.9074 | 0.9886 | 0.9675 |
| 7 | 0.9010 | 0.9127 | 0.8983 | 0.9340 | **0.9519** |
| OA | 0.9068 | 0.9099 | 0.9121 | 0.9317 | **0.9337** |
| AA | 0.9238 | 0.9132 | 0.9120 | 0.9180 | **0.9420** |
| Kappa | 0.8872 | 0.8907 | 0.8931 | 0.9168 | **0.9196** |




(a) Ground Truth (b) DCFSL (c) CMFSL

(d) DSDAN (e) CSGCN-FAL1 (f) CSGCN-FAL2

Fig. 2. Classification maps of the considered methods on CRPaviaU-DPaviaC dataset

gap may be attributed to the fusion strategy in CSGCN-FAL1, where the Euclidean distance does not fully account for the underlying structure of the probability simplex. Both CSGCN-FAL1 and CSGCN-FAL2 outperform the other cross-scene classification methods (DCFSL, CMFSL, and DSDAN) in overall performance, demonstrating the effectiveness of exploiting graph structures and utilizing AL. In each class, at least one of the CSGCN-AL models achieves either the highest or second-highest classification accuracy, further validating the advantages of our proposed approach.

Due to space limitations, more experimental results on other real-world datasets are not included here, but they demonstrate similar advantages of our proposed method.

## V. CONCLUSION

In this study, we proposed an efficient cross-scene HSI classification network integrated with a novel AL module that selects target samples for labeling through a fusion of multiple sampling criteria. The base classifier is built upon the existing DMS-GCN for single-scene classification, which extracts HSI features at different scales from the source and target samples separately. To adapt it for cross-scene HSI classification, we introduced a feature alignment module and a cross-scene loss function to mitigate domain shifts. In the AL module, a fusion method with adaptive weighting was developed to balance the contributions of four classical AL selection criteria, and an alternating optimization algorithm was proposed to solve it. Experiments on a real HSI dataset demonstrated the effectiveness of the proposed method, highlighting the advantages of integrating graph-based learning and active learning for cross-domain HSI classification.

## REFERENCES


[1] M. Ye, Y. Qian, J. Zhou, and Y. Y. Tang, "Dictionary learning-based feature-level domain adaptation for cross-scene hyperspectral image classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 55, no. 3, pp. 1544–1562, 2017.

[2] Y. Zhu, F. Zhuang, J. Wang, J. Chen, Z. Shi, W. Wu, and Q. He, "Multi-representation adaptation network for cross-domain image classification," *Neural Networks*, vol. 119, pp. 214–221, 2019.

[3] Y. Zhang, W. Li, W. Sun, R. Tao, and Q. Du, "Single-source domain expansion network for cross-scene hyperspectral image classification," *IEEE Trans. Image Process.*, vol. 32, pp. 1498–1512, 2023.

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

[5] A. Ozdemir and K. Polat, "Deep learning applications for hyperspectral imaging: a systematic review," *J. Inst. Electron. Comput.*, vol. 2, no. 1, pp. 39–56, 2020.

[6] F. Ullah, I. Ullah, K. Khan, S. Khan, and F. Amin, "Advances in deep neural network-based hyperspectral image classification and feature learning with limited samples: a survey," *Appl. Intell.*, vol. 55, no. 6, pp. 1–48, 2025.

[7] C. Wang, Y. Guo, and J. Fu, "Dual-stream discriminative attention network for cross-scene hyperspectral image classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 62, 2024, Art. No. 5515512.

[8] W. Sun, K. He, G. Yang, J. Peng, K. Ren, and J. Li, "A cross-scene self-representative network for hyperspectral band selection," *IEEE Trans. Geosci. Remote Sens.*, vol. 62, 2024, Art. No. 5509212.

[9] S. J. Pan and Q. Yang, "A survey on transfer learning," *IEEE Trans. Knowl. Data Eng.*, vol. 22, no. 10, pp. 1345–1359, 2010.

[10] Z. Li, M. Liu, Y. Chen, Y. Xu, W. Li, and Q. Du, "Deep cross-domain few-shot learning for hyperspectral image classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 60, 2022, Art. No. 5501618.

[11] B. Xi, J. Li, Y. Li, R. Song, D. Hong, and J. Chanussot, "Few-shot learning with class-covariance metric for hyperspectral image classification," *IEEE Trans. Image Process.*, vol. 31, pp. 5079–5092, 2022.

[12] P. Liu, H. Zhang, and K. B. Eom, "Active deep learning for classification of hyperspectral images," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 10, no. 2, pp. 712–724, 2017.

[13] C. Li and L. Zhang, "Hyperspectral image classification by combination of active learning and extended multi-attribute profile," in *Proc. 2nd Int. Conf. Comput. Commun.*, Chengdu, China, Oct. 2016, pp. 541–544.

[14] M. Xu, Q. Zhao, and S. Jia, "Multiview spatial–spectral active learning for hyperspectral image classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 60, 2022, Art. No. 5512415.

[15] J. Lin, L. Zhao, S. Li, R. Ward, and Z. J. Wang, "Active-learning-incorporated deep transfer learning for hyperspectral image classification," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 11, no. 11, pp. 4048–4062, 2018.

[16] Y. Yang, X. Tang, X. Zhang, J. Ma, F. Liu, X. Jia, and L. Jiao, "Semi-supervised multiscale dynamic graph convolution network for hyperspectral image classification," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 35, no. 5, pp. 6806–6820, 2024.

[17] G. Koliander, Y. El-Laham, P. M. Djurić, and F. Hlawatsch, "Fusion of probability density functions," *Proc. IEEE*, vol. 110, no. 4, pp. 404–453, 2022.

[18] M. Stone, "The opinion pool," *Ann. Math. Statist.*, vol. 32, no. 4, pp. 1339–1342, 1961.