Refined Prototypical Contrastive Learning for Few-Shot Hyperspectral Image Classification

Quanyong Liu, Jiangtao Peng, *Senior Member, IEEE*, Yujie Ning, Na Chen, Weiwei Sun, *Senior Member, IEEE*, Qian Du, *Fellow, IEEE*, and Yicong Zhou, *Senior Member, IEEE*

***Abstract*— Recently, prototypical network-based few-shot learning (FSL) has been introduced for small-sample hyperspectral image (HSI) classification and has shown good performance. However, existing prototypical-based FSL methods have two problems: prototype instability and domain shift between training and testing datasets. To solve these problems, we propose a refined prototypical contrastive learning network for FSL (RPCL-FSL) in this article, which incorporates supervised contrastive learning (CL) and FSL into an end-to-end network to perform small-sample HSI classification. To stabilize and refine the prototypes, RPCL-FSL imposes triple constraints on prototypes of the support set, i.e., CL-, self-calibration (SC)-, and cross-calibration (CC)-based constraints. The CL module imposes an internal constraint on the prototypes aiming to directly improve the prototypes using support set samples in the CL framework, and the SC and CC modules impose external constraints on the prototypes by using the prediction loss of support set samples and the query set prototypes, respectively. To alleviate a domain shift in the FSL, a fusion training strategy is designed to reduce the feature differences between training and testing datasets. Experimental results on three HSI datasets demonstrate that the proposed RPCL-FSL outperforms existing state-of-the-art deep learning and FSL methods.**

***Index Terms*— Contrastive learning (CL), few-shot learning (FSL), hyperspectral image (HSI) classification, prototypical network.**

I. Introduction

HYPERSPECTRAL image (HSI) is a 3-D data cube, which combines the spectral and spatial information, reflecting the radiation characteristics and the spatial geometric relationship of ground objects [1], [2], [3]. Due to the rich information and subtle spectral difference identification ability, HSI has been widely used in various fields, such as mineral identification, military target detection, and precision agriculture.

HSI classification is important for HSI analysis. In early research, some traditional algorithms, such as k-nearest neighbor (NN) [4] and support vector machine (SVM) [5], were used for HSI classification. These algorithms only focus on the spectral information of samples. Subsequent classification methods, such as composite kernels [6], [7], joint sparse representation [8], [9], [10], extended morphological profiles [11], and Gabor wavelets [12], fully consider the spatial–spectral features of HSIs and obtain better classification results [13]. However, the above algorithms always require specific designed features, such as morphological and Gabor features, and are less adaptable to different application scenarios.

In the past few years, deep learning methods have been introduced for HSI classification and achieved excellent performance [14], [15], [16], [17]. One of the biggest advantages of deep learning methods is its ability to adaptively learn features from different types of data. Chen et al. [18] first proposed a deep HSI classification network based on the stacked autoencoder (SAE). They further introduced a convolutional neural network (CNN)-based HSI classification method [19], which adopts sparse connectivity and weight sharing to achieve effective feature extraction. After that, Zhong et al. [20] proposed a spectral–spatial residual network (SSRN) for HSI classification, which relieves the gradient vanishing or explosion problems as the increase of network layers. Compared with traditional algorithms, deep learning approaches can automatically learn features from a large amount of labeled data without the need of artificially designed specific feature patterns [21]. However, their excellent performance relies on the use of large numbers of labeled samples [22]. In practice, collecting labeled samples is time-consuming, and the available labeled samples are usually limited. Thus, we usually face a small-sample classification scenario and the high-precision small-sample classification of HSIs is still a challenging task.

In order to solve the label scarcity problem in the deep learning framework, researchers have proposed various small-sample learning strategies. An intuitive idea is to directly augment samples based on the available labeled samples. This method is called data augmentation method [23], [24], which uses a series of transformations such as clipping, flipping,

Manuscript received 11 November 2022; revised 8 February 2023; accepted 12 March 2023. Date of publication 15 March 2023; date of current version 24 March 2023. This work was supported in part by the National Natural Science Foundation of China under Grant 42171351, Grant 42122009, and Grant 41971296; and in part by the Natural Science Foundation of Hubei Province under Grant 2021CFA087. *(Corresponding authors: Jiangtao Peng; Weiwei Sun.)*

Quanyong Liu, Jiangtao Peng, Yujie Ning, and Na Chen are with the Hubei Key Laboratory of Applied Mathematics, Faculty of Mathematics and Statistics, Hubei University, Wuhan 430062, China (e-mail: pengjt1982@hubu.edu.cn).

Weiwei Sun is with the Department of Geography and Spatial Information Techniques, Ningbo University, Ningbo 315211, China (e-mail: sunweiwei@nbu.edu.cn).

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

Yicong Zhou is with the Department of Computer and Information Science, University of Macau, Macau 999078, China (e-mail: yicongzhou@um.edu.mo).

Digital Object Identifier 10.1109/TGRS.2023.3257341

rotating, and adding random noise to enlarge the number of labeled samples. Except for data augmentation, simplifying the network structure can also reduce the dependency on labeled samples. Many lightweight networks adopt depthwise convolution and pointwise convolution [25], [26] to reduce the network parameters in the convolution process and show good performance for small-sample HSI classification.

Recently, self-supervised learning techniques have been used for small-sample HSI classification. It mines the supervised information by pretext task from a large number of unlabeled samples to train the network and then uses a small amount of label information to fine-tune the network for classification. Liu et al. [27] proposed a deep multiview learning (DMVL) method, which generates multiple views for each sample using the band separation method and trains the network by optimizing self-supervised contrastive learning (CL) loss for different views. Zhao et al. [28] introduced a contrastive self-supervised learning method for small-sample HSI classification based on Siamese networks.

Due to its effectiveness in identifying new unseen classes using few training samples, few-shot learning (FSL) methods are naturally suitable for small-sample classification [29], [30], [31], [32], [33]. Different from small-sample classification, FSL adopts an episodic learning strategy for meta-learning. In the meta-training phase, the training dataset is decomposed into different meta-tasks, each of which contains different categories. By training on different meta-tasks with different categories, the model can generalize to classify new unseen categories in the meta-testing phase. There are three main meta-learning methods to solve the few-shot classification problem, namely, model-, metric-, and optimization-based schemes. Among these methods, metric-based methods are widely used in the field of HSI classification due to its simplicity and effectiveness. Liu et al. [34] proposed a deep FSL (DFSL) method for small-sample HSI classification, which creates a deep 3-D residual network to learn a metric space to maximize the class discriminant and separability. Gao et al. [35] proposed a deep relation network for hyperspectral few-shot classification, which selects a few samples from the testing dataset to fine-tune the trained model to narrow the differences between domains. Then, Li et al. [36] introduced a deep cross-domain FSL (DCFSL) method for HSI classification, which embeds an adversarial-based domain adaptation module into an FSL framework to overcome domain shift. Xi et al. [37] proposed a class-covariance metric-based FSL (CMFSL) method for HSI classification. It designs a spectral prior refinement module to alleviate the domain shift and uses class-covariance metric instead of ordinary Euclidean distance for better classification. Zhang et al. [38] proposed a graph information aggregation cross-domain FSL (Gia-CFSL) framework, which employs a graph information aggregation-based domain alignment strategy to suppress domain shift.

Currently, most FSL-based HSI classification methods focus on the problem of domain shift, and they try to narrow the difference between the training and testing datasets in FSL by designing or applying various domain adaptation strategies. However, FSL is totally different from the domain adaptation [39], [40], [41]. Domain adaptation usually assumes that the feature space and label space of source and target domains are the same, and the distribution of two domains is different but related. While the feature and label of training and testing datasets are generally different in FSL. Thus, simply embedding the domain adaptation module into an FSL framework is not always effective. In addition, existing metric-based FSL methods usually compute prototypes directly from support set samples, which ignores the validity of prototypes and may decrease the subsequent network performance. To solve this problem, many prototype-based FSL variants have been proposed [38], [42], [43]. Cheng et al. [44] proposed a Siamese-prototype network (SPNet), which calibrates the prototypes through self-calibration (SC) and intercalibration modules to obtain more representative prototypes. Zhang et al. [45] proposed a global prototypical network (GPN), which uses the strategy of global prototype to train the network. The global prototype vector is continuously updated in the iterative training, which can finally obtain more accurate category prototypes. However, these methods only emphasize the external constraints on the prototype, ignoring the generation process of prototype.

To solve the aforementioned problems, a refined prototypical contrastive learning network for FSL (RPCL-FSL) is proposed for HSI classification. The proposed RPCL-FSL framework incorporates supervised CL and FSL. In the network, all samples go through an adaptive mapping layer and a deep 3-D residual network to extract embedded features, and a metric function is employed to learn the similarity between the support and the query set. To obtain effective and stable prototypes, we impose triple constraints on the support set prototypes. First, the support set prototypes are internally restricted by supervised CL using the labeled support set samples. Then, SC and cross-calibration (CC) modules are designed to impose external constraints on the prototypes using the prediction loss of support set samples and the query set prototypes, respectively. Based on the triple constraints, representative refined prototypes can be obtained.

The main contributions of the proposed RPCL-FSL are given as follows.

1) We propose a novel framework for few-shot HSI classification based on CL, which combines supervised CL with FSL for the first time and achieves excellent performance.
2) We design triple-constraint-based prototypical learning modules in the prototype-based FSL network, which is able to obtain more representative prototypes without adding any parameters and solves the unstable model performance due to the poor quality of support set prototypes.
3) We adopt a fusing training strategy to train the model alternately using labeled samples of the training and testing datasets. The model not only learns the meta-knowledge that can be transferred but also learns the discriminative embedding features for the testing dataset, which alleviates the domain shift problem in FSL effectively.

II. Related Work

A. Few-Shot Learning

It is an example of meta-learning, which allows models to learn to "learn" and then be able to handle similar types of tasks, not just a single classification task. It hopes that the machine learning model can be like the human brain, and after learning a large amount of data from certain categories, only a small number of labeled samples are needed to identify new categories.

FSL involves two datasets: one is the training set with a large number of labeled samples for model training and the other is a testing dataset with only very few labeled samples. The number of categories in the training dataset is generally required to be larger than that in the testing dataset and the categories of two datasets can be completely different.

Existing FSL methods usually adopt episodic training manner. The algorithm consists of two stages: meta-training and meta-testing. In the meta-training stage, different meta-tasks are constructed from the training dataset to train the model such that the model has the generalization or learning ability to handle the classification of unseen categories. In the meta-testing stage, the trained model is employed to classify new categories in the testing dataset with the need of limited labeled samples in these categories.

B. Prototypical Networks

In the meta-training stage of FSL, N categories and K samples in each category are first randomly selected from the training set. The selected $N \times K$ samples consists of the support set. Then, a batch of samples from the remaining data in the N categories are from the query set. A meta-task consists of a support set and a query set, which is called the N-way K-shot problem. The prototype for each category is usually defined as the mean of samples in the category. The model is trained by minimizing the distance from the samples in the query set to the corresponding class prototypes in the support set.

As a metric-based FSL method, a prototypical network calculates the mean of support samples (i.e., support prototype) for each category in the embedding space and makes the prediction by calculating the distance between the testing sample and each support prototype [30].

It is worth noting that different meta-tasks are constructed during each training process. The model can learn common parts of different meta-tasks under this mechanism, which facilitates the classification for new unseen meta-tasks.

C. Supervised CL

Self-supervised CL can learn valuable information from a large number of unlabeled samples through a CL paradigm and has shown excellent performance for small-sample classification.

Assume that a training batch contains N samples. For each sample x in the training batch, by performing random cropping, Gaussian blur, or other data enhancement methods, a new sample x^+ is generated. Thus, we can obtain a contrastive group, which consists of $2N$ samples, i.e., original N samples and their corresponding N augmented samples. The sample x^+ is highly similar to x and is denoted as a positive sample of x. The remaining $2N-2$ samples in the contrastive group are recorded as negative samples x^-.

The purpose of CL is to learn a feature encoder f such that the distance between x and positive sample x^+ is as close as possible and the distance between x and negative samples x^- is as far as possible

$$\text{score}\big(f(x), f(x^+)\big) \gg \text{score}\big(f(x), f(x^-)\big) \tag{1}$$

where score$(\cdot)$ represents the similarity score function between two features.

On this basis, Khosla et al. [46] proposed a supervised CL algorithm, which is able to extend the self-supervised CL to a fully supervised setting, enabling label information to be fully exploited. Supervised CL randomly selects two labeled samples from each category. The samples from the same category are recorded as positive sample pair, while the samples from different categories are negative samples. Therefore, when other conditions remain unchanged, the basis of CL has changed from "whether from the same sample" to "whether from the same category."

III. Proposed Method

Fig. 1 shows the procedure of the proposed RPCL-FSL network. It incorporates the supervised CL and FSL into an end-to-end network framework, imposes a triple restriction on the support set prototypes to solve the prototype instability problem, and adopts a fusion training (FT) strategy to alleviate the domain shift.

For FSL, two datasets are given: the training dataset D_s with C_s classes and the testing dataset D_t with C_t classes, where C_s and C_t represent the number of classes in the training and testing datasets, respectively. It is worth noting that D_t is the target dataset needed to be classified, and C_t is smaller than C_s.

As shown in Fig. 1, the proposed model uses an FT strategy that first trains the network using the training dataset and then fine-tunes the network using the testing dataset, that is, it consists of two meta-training processes on the training and testing datasets, respectively. The training process on the training and testing datasets is the same. In the following, we take the training dataset as an example to describe the meta-training process. First, contrastive support and query sets are constructed. Then, a trainable adaptive mapping layer is used to solve the problem of spectral dimensionality differences between training and testing datasets. Finally, a feature extractor is trained using a refined prototypical network. When the meta-training on the training dataset is finished, the model performs the meta-training on the test dataset and fine-tunes the feature extractor using limited labeled testing samples.

A. Contrastive Support Set and Query Set

The construction process of contrastive support set for training and testing datasets is the same. For the training

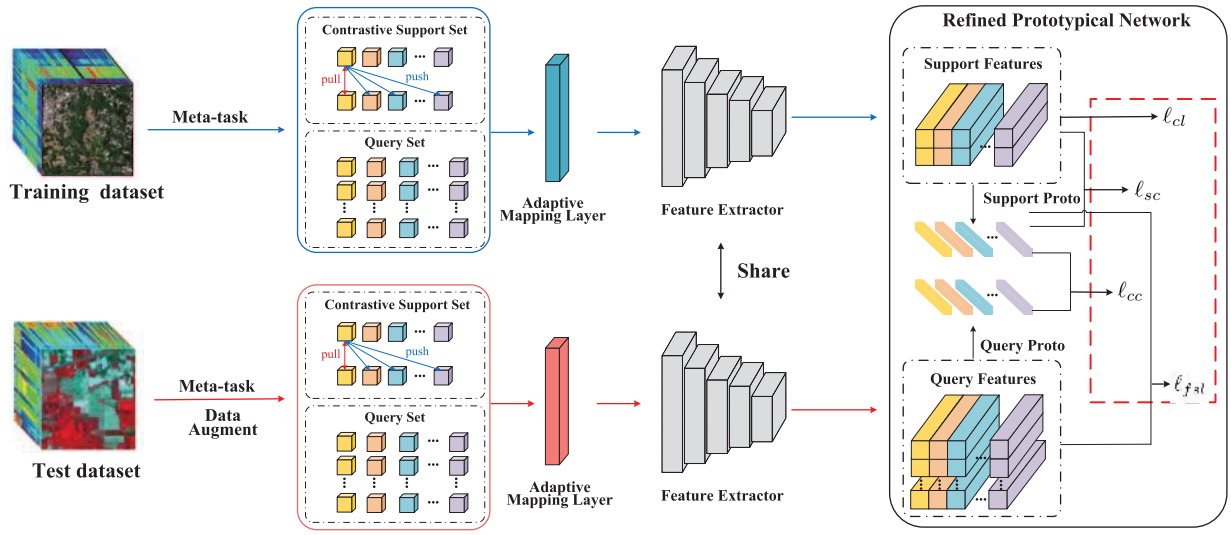


Fig. 1. Flowchart of the proposed RPCL-FSL, which consists of two meta-training processes on the training and testing datasets. In the meta-training process, contrastive support and query sets are first constructed, and then, a trainable adaptive mapping layer is used to unify the spectral dimensionality of training and testing datasets. Finally, a feature extractor is trained using a refined prototypical network.

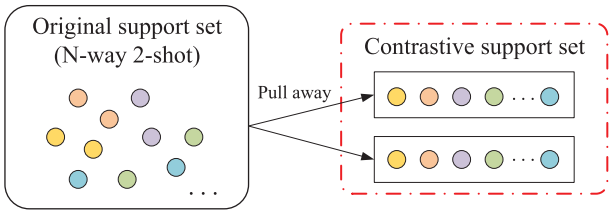


Fig. 2. Illustration of contrastive support set.

dataset, N classes from C_s training classes are randomly selected to form an episode, where N is equal to C_t. Then, M_s and M_q labeled samples from each of selected N classes are chosen to form the support set $S_s = \{(x_i, y_i)\}_{i=1}^{n_s}$ and the query set $Q_s = \{(x_j, y_j)\}_{j=1}^{n_q}$, respectively, where $n_s = M_s \times N$ and $n_q = M_q \times N$. In this article, M_s is set to 2 and M_q is set to 19 as in [34] to keep it consistent with existing algorithms.

Here, we embed the labels into the construction process of the support set. As M_s as 2, only two labeled samples are selected from each of N classes to form the support set, and an N-way two-shot paradigm is constructed. The construction of the contrastive support set is shown in Fig. 2. We put two labeled samples from the same class into two individual groups and obtain two contrastive groups, where each group has N samples from N different classes. In the contrastive groups, each sample only has one positive sample from the same class and the rest $2N - 2$ samples are negative samples.

B. Adaptive Mapping Layer and Feature Extraction Network

In order to alleviate the domain shift problem between D_s and D_t, we train the embedded feature extractor alternately with labeled samples in D_s and D_t so that the model can not only discover the transferable knowledge in the training dataset but also can learn discriminant information in the test dataset. However, a problem with alternately training is that the spectral dimensions of training and test datasets are different. To unify the dimensions of training and testing datasets, an adaptive mapping layer is used [47].

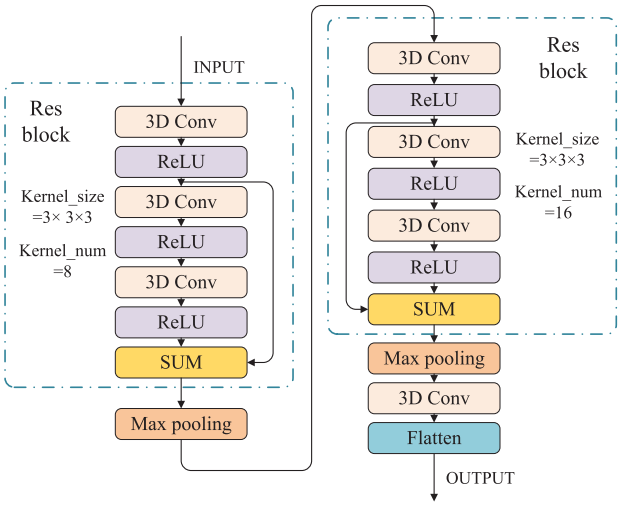


Fig. 3. Architecture of deep residual 3-D CNN.

The adaptive mapping layer is essentially a 2-D convolutional layer consisting of d convolution kernels of size $1 \times 1 \times$ ch, where ch and d are the dimensions of input and output data, respectively. For training and testing datasets with different input dimensions ch, by performing an adaptive mapping layer, the dimension of output data is the same, i.e., d.

After the adaptive mapping layer, a deep 3-D residual network [34] is introduced to extract spatial–spectral features, as shown in Fig. 3. As shown in Fig. 3, a deep 3-D CNN with two residual blocks, two pooling layers, and a convolutional layer is used as the embedding function. The residual blocks can ensure that the model has a deeper network and a faster update speed. Also, each residual block is connected with a 3-D max-pooling layer to reduce computation and aggregate features.

C. Refined Prototypes Using Triple Constraints

This article uses the prototypical network-based FSL to process the dataset, that is, the feature extractor f_θ is trained

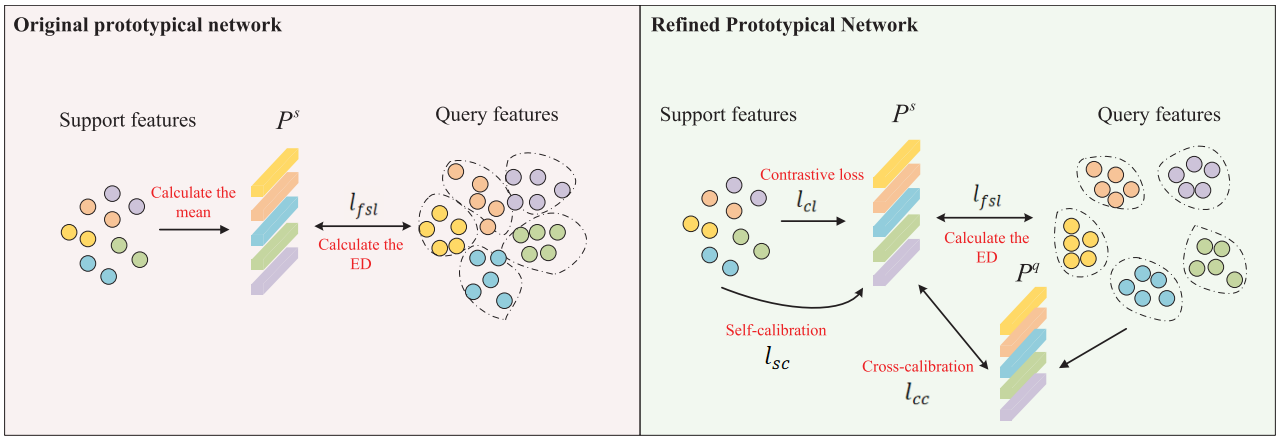


Fig. 4. Main innovations compared with the original prototypical network. ED stands for Euclidean distance, and P^s and P^q are the prototypes of the support and query sets, respectively. The prototypes of the original prototypical network are directly obtained from the mean of the support set samples, and then, the network is trained by minimizing the Euclidean distance between the query set samples and the prototype. The refined prototypical network imposes triple restrictions on the support set prototype, where a supervised CL strategy is used to improve the feature learning ability, an SC strategy is adopted to calibrate the support set prototypes using prediction loss of support set samples, and a CC strategy is proposed to reduce its deviation from the real category prototype.

by minimizing the distance between the query set samples and the category prototype of the contrastive support set, as shown in Fig. 4(a).

The category prototype of the cth class in the support set is calculated as

$$P_s^c = \frac{1}{M_s} \sum_{x_i \in S_s^c} f_\theta(x_i) \tag{2}$$

where S_s^c represents the set of samples belonging to the cth class in the support set.

The samples in the query set Q_s are fed through the feature extraction network to extract embedded features. For a query set sample $x_j \in Q_s$ with true label c_j, the prototypical network can predict the probability of the sample x_j belonging to the class c_j as

$$p(y_j = c_j | x_j) = \frac{\exp\left(-d\left(f_\theta(x_j), P_s^{c_j}\right)\right)}{\sum_{c=1}^{N} \exp\left(-d\left(f_\theta(x_j), P_s^{c}\right)\right)} \tag{3}$$

where $d(\cdot)$ represents the Euclidean distance.

The loss for each episode can be calculated using the negative log probability as

$$\ell_{\text{fsl}} = -\frac{1}{n_q} \sum_{j=1}^{n_q} \log p(y_j = c_j | x_j). \tag{4}$$

From (3), we can see that the category prototypes of the initial support set have a great influence on the training of the model. As the samples in the contrastive support set are randomly selected, the computed category prototypes may have a large deviation from the real overall category prototypes, resulting in unstable model performance. In order to alleviate the above problem, this article proposes a triple restriction on the prototype of support set, as shown in Fig. 4(b). In detail, a supervised CL strategy is used to improve the feature learning ability of f_θ such that the similarity between samples in the same class and the dissimilarity between samples from different classes are increased. Then, an SC strategy is adopted to calibrate the support set prototypes using prediction loss of support set samples. Finally, a CC strategy is proposed to reduce its deviation from the real category prototype.

For the N-way two-shot meta task, there are two contrastive sets $S = \{x_1, x_3, x_5, \ldots, x_{2N-1}\}$ and $\bar{S} = \{x_2, x_4, x_6, \ldots, x_{2N}\}$, where x_{2k-1} and x_{2k} are from the same category. The aim of CL is to learn a feature extractor to increase the similarity between positive sample pairs and decrease the similarity between negative sample pairs.

After constructing the positive and negative sample pairs, the noise contrast estimation (NCE)-based loss function $\ell_{m,n}$ between $f_\theta(x_m)$and $f_\theta(x_n)$ is calculated as

$$\ell_{m,n} = -\log \frac{\exp\left(\frac{s(f_\theta(x_m), f_\theta(x_n))}{\tau}\right)}{\sum_{k=1}^{2N} \mathbf{1}_{[k \neq m]} \exp\left(\frac{s(f_\theta(x_m), f_\theta(x_k))}{\tau}\right)} \tag{5}$$

where $x_m, x_n \in S_s$ belong to the same category and $s\left(f_\theta(x_m), f_\theta(x_n)\right)$ measures the similarity between $f_\theta(x_m)$ and $f_\theta(x_n)$. τ is a temperature coefficient and set to 0.5 in the experiments [46], [48].

Considering that $\ell_{2k-1,2k} \neq \ell_{2k,2k-1}$ in (5), the final average contrastive loss ℓ_{cl} of an episode is calculated as

$$\ell_{\text{cl}} = \frac{1}{2N} \sum_{k=1}^{N} (\ell_{2k-1,2k} + \ell_{2k,2k-1}). \tag{6}$$

1) Self-Calibration: The prototypical network trains the embedded feature extractor by minimizing the loss between the support set prototypes and the query set samples. Considering that the prediction loss of support set samples also helps to calibrate the support set prototypes, an SC module is introduced to increase the discriminant ability of the extracted prototypes [44]. Similar to the prediction of query samples in (3), the prediction for a support set sample $x_i \in S_s$ with the true label c_i can be written as

$$p(y_i = c_i | x_i) = \frac{\exp\left(-d\left(f_\theta(x_i), P_s^{c_i}\right)\right)}{\sum_{c=1}^{N} \exp\left(-d\left(f_\theta(x_i), P_s^{c}\right)\right)}. \tag{7}$$

Similar to (4), the loss of SC module can be calculated as

$$\ell_{\mathrm{sc}} = -\frac{1}{n_s}\sum_{i=1}^{n_s}\log p(y_i = c_i|x_i). \tag{8}$$

2) Cross Calibration: We can know that labeled samples in the query set are only used to calculate the prediction loss, and other information has not been mined. To fully exploit the query set samples, we propose a CC module to further calibrate the support set prototypes. The numbers of labeled samples per class in the support and query sets are 2 and 19 (i.e., $M_s = 2$ and $M_q = 19$), respectively. As the number of labeled samples in the query set is much larger, the deviation between the prototype of the query set and the real prototype is smaller. Therefore, we calibrate the support set prototypes with the use of query set prototypes.

The category prototype for the cth class of the query set is calculated as

$$P_q^c = \frac{1}{M_q}\sum_{x_j \in Q_s^c} f_\theta(x_j) \tag{9}$$

where Q_s^c represents the set of samples belonging to the cth class in the query set.

The loss of CC module can also be calculated as

$$\ell_{\mathrm{cc}} = -\frac{1}{N}\sum_{c=1}^{N}\log\frac{\exp\left(-d\left(P_q^c, P_s^c\right)\right)}{\sum_{c=1}^{N}\exp\left(-d\left(P_q^c, P_s^c\right)\right)}. \tag{10}$$

The total loss for the RPCL-FSL can be expressed as the sum of four losses

$$\ell = \ell_{\mathrm{fsl}} + \ell_{\mathrm{cl}} + \ell_{\mathrm{sc}} + \ell_{\mathrm{cc}}. \tag{11}$$

D. Fusion Training

The algorithm involves two datasets, i.e., training data D_s and test data D_t, which are derived from different sensors and have completely different categories of objects. Therefore, there have large domain shifts between the two datasets, and the model trained on D_s may not perform well on the data D_t. To alleviate the domain shift problem, we adopt an FT strategy, which uses the labeled samples from D_s and D_t to train the feature extractor alternately.

In the training stage, we first select 200 labeled samples from each class in the D_s. For the test data D_t, only five labeled samples per class are available, and a data augmentation method (e.g., cropping and restoration) is used to expand the number of labeled samples per class from 5 to 200. We use a rectangle with an aspect ratio of 3:4 and a size range of 8%–100% of the original image to crop the original image and then restore the cropped image to the size of the original image (i.e., 9×9). At this time, the number of labeled samples per class for the training and testing datasets is 200. Then, we alternately train the model. For example, for 3000 training iterations, the first 1000 epochs are trained using labeled samples in D_s and the last 2000 epochs are trained using labeled samples in D_t. The detail of the fusing training process is shown in Algorithm 1.

Algorithm 1 Fusing Training Process

Input: The feature extractor f
The contrastive support set S_s and query set Q_s in D_s
The contrastive support set S_t and query set Q_t in D_t
Output: The loss of model

if epoch $<$ 1000 **then**
  Train with S_s and Q_s
  Calculate the prototype of $S_s \rightarrow P_s$
  Calculate the prototype of $Q_s \rightarrow P_q$
  for $x_j \in Q_s$ **do**
    Calculate the ℓ_{fsl} of an episode by Eq. (4)
  end for
  for $x_m, x_n \in S_s$ **do**
    Calculate the ℓ_{cl} of an episode by Eq. (6)
  end for
  for $x_i \in S_s$ **do**
    Calculate the ℓ_{sc} of an episode by Eq. (8)
  end for
  for $P_q^c \in P_q$ **do**
    for $P_s^c \in P_s$ **do**
      Calculate the ℓ_{cc} of an episode by Eq. (10)
    end for
  end for
  Return the Loss ℓ of an episode by Eq. (11)
else
  Train with the same process as above using S_t and Q_t
end if

IV. Experimental Results and Discussion

A. Datasets

Four HSI datasets, including Indian Pines (IP), University of Pavia (UP), Salians (SA), and Chikusei, are used in the experiments. In the FSL setting [36], Chikusei is selected as the training dataset, and the other three datasets are selected as the testing datasets.

1) Training Dataset (Chikusei): These data were captured by the Headwall Hyperspec-VNIR-C sensor in Chikusei with a spatial resolution of 2.5 m. The scene has the spatial size of 2517 × 2335 and 128 spectral bands. It contains 19 land cover classes, as shown in Table I. The pseudo-color composite image and the ground-truth map of Chikusei are shown in Fig. 5.

2) Testing Datasets:

a) Indian pines: This dataset was acquired by the airborne visible infrared imaging spectrometer (AVIRIS) sensor at the Indiana Pine test site in northwestern Indiana. The IP dataset has 145 × 145 pixels and 224 spectral bands. After deleting 20 bad bands, the rest 200 spectral bands are used in the experiment. The spatial resolution is 20 m. This scene contains 16 classes. Table II shows the land cover classes and the corresponding numbers of samples in the IP dataset. The pseudo-color composite image and the ground-truth map are shown in Fig. 6.

b) University of Pavia: This dataset was acquired by the ROSIS-03 sensor at the UP test site in northern Italy. The UP dataset has 610 × 340 pixels and 115 spectral bands, in which

TABLE I
LAND COVER CLASSES AND NUMBERS OF SAMPLES IN CHIKUSEI

Class	Name	Pixels
1	Water	2345
2	Bare soil (school)	2859
3	Bare soil (park)	236
4	Bare soil (farmland)	48525
5	Natural plants	4297
6	Weeds in farmland	1108
7	Forest	20516
8	Grass	6515
9	Rice field (groim)	13369
10	Rice field (first stage)	1268
11	Row crops	5961
12	Plastic house	2193
13	Manmade (nan-dark)	1220
14	Manmade (dark)	7664
15	Manmade (blue)	431
16	Manmade (red)	222
17	Manmade grass	1040
18	Asphalt	801
19	Paved ground	145
	Total	77592

TABLE II
LAND COVER CLASSES AND NUMBERS OF SAMPLES IN IP

Class	Name	Pixels
1	Alfalfa	46
2	Corn-notill	1428
3	Corn-mintill	830
4	Corn	237
5	Grass-pasture	483
6	Grass-trees	730
7	Grass-pasture-mowed	28
8	Hay-windrowed	478
9	Oats	20
10	Soybean-notill	972
11	Soybean-mintill	2455
12	Soybean-clean	593
13	Wheat	205
14	Woods	1265
15	Buildings-Grass-Trees-Drives	386
16	Stone-Steel-Towers	93
Total		10249

TABLE III
LAND COVER CLASSES AND NUMBERS OF SAMPLES IN UP

Class	Name	Pixels
1	Asphalt	6631
2	Meadows	18649
3	Gravel	2099
4	Trees	3064
5	Metal sheets	1345
6	Bare soil	5029
7	Bitumen	1330
8	Bricks	3682
9	Shadows	947
Total		42776

TABLE IV
LAND COVER CLASSES AND NUMBERS OF SAMPLES IN SA

Class	Name	Pixels
1	Brocoli_green_weeds 1	2009
2	Brocoli_green_weeds 2	3726
3	Fallow	1976
4	Fallow_rough_plow	1394
5	Fallow_smooth	2678
6	Stubble	3959
7	Celery	3579
8	Grapes_untrained	11271
9	Soil_vinyard_develop	6203
10	Corn_senesced_green_weeds	3278
11	Lettuce_romaine_4wk	1068
12	Lettuce_romaine_5wk	1927
13	Lettuce_romaine_6wk	916
14	Lettuce_romaine_7wk	1070
15	Vinyard_untrained	7268
16	Vinyard_vertical_trellis	1807
Total		54129

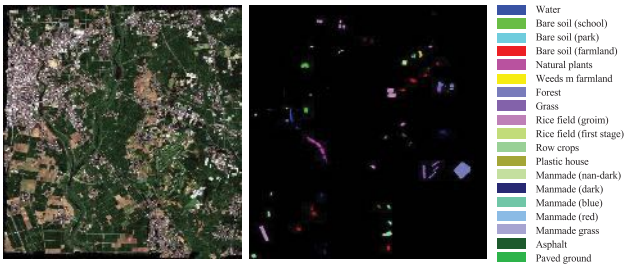


Fig. 5. Pseudo-color composite image and ground-truth map of Chikuse.

specific 103 spectral bands are involved in the experiment. The spatial resolution of this image is 1.3 m. The dataset contains nine classes, as shown in Table III. The pseudo-color composite image and the corresponding ground-truth map are shown in Fig. 7.

c) Salians: This dataset was acquired by the AVIRIS sensor at the Salinas Valley test site in California. The SA dataset has 512 × 217 pixels and 224 spectral bands, in which specific 204 spectral channels are involved in the experiment. The spatial resolution of this image is 3.7 m. The dataset contains 16 classes, as shown in Table IV. The pseudo-color composite image and the corresponding ground-truth map are shown in Fig. 8.

B. Experiment Setting

Several state-of-art deep learning and FSL-based algorithms are employed for comparison, including HybirdSN [49], SSRN [20], A2S2K [50], SSCL [48], DFSL [34], DCFSL [36], CMFSL [37], and Gia-CFSL [38]. HybirdSN, SSRN, A2S2K, and SSCL are deep learning-based HSI classification methods. For these methods, five labeled samples per class are selected

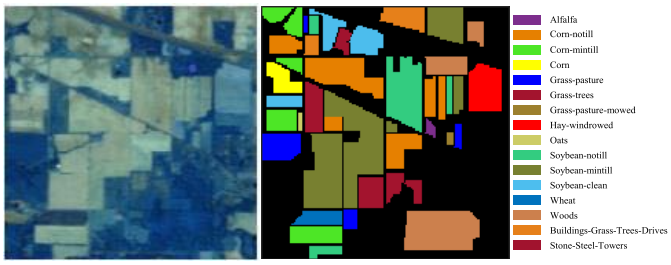


Fig. 6. Pseudo-color composite image and ground-truth map of IP.

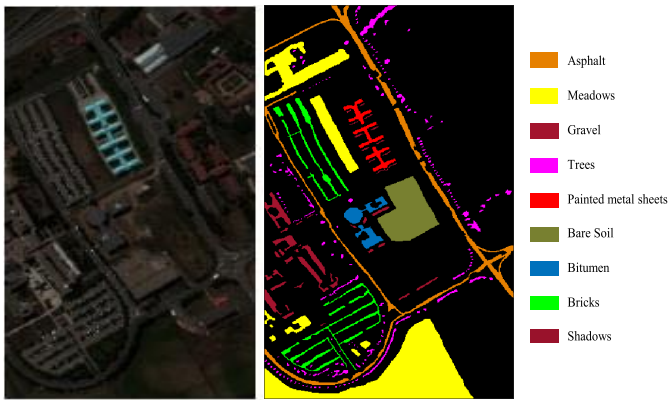


Fig. 7. Pseudo-color composite image and ground-truth map of UP.

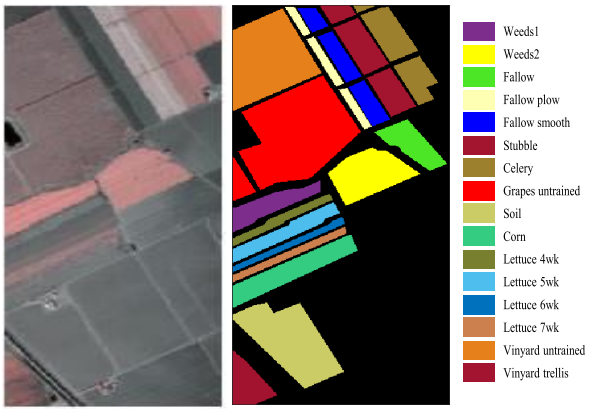


Fig. 8. Pseudo-color composite image and ground-truth map of SA.

as the training set and the rest as the testing set. DFSL, DCFSL, CMFSL, and Gia-CFSL are the latest FSL-based classification methods. For DFSL and CMFSL, four datasets (i.e., University of Houston 2013, Chikusei, Kennedy Space Center, and Botswana) are used as the training datasets, and 100 bands of different HSI datasets are chosen using a graph representation band selection method to ensure the consistency of input dimensions [51]. For DCFSL and Gia-CFSL, the Chikusei dataset is chosen as the training dataset, and a feature mapping layer is used to ensure the dimension consistency between training and testing datasets. Following the setting with above four FSL-based methods (i.e., DFSL, DCFSL, CMFSL, and Gia-CFSL), 200 labeled samples per class from the training set are randomly selected for FSL. In the test dataset, only five labeled samples per class are used for FSL and the remaining samples are used for testing. In an episode training, an N-way K-shot paradigm is constructed on the support set, where N is set as the number of classes in the testing dataset (e.g., $N = 16$ for IP and SA and $N = 9$ for UP) and K is set to 2 (i.e., $M_s = 2$). On the query set, the number of samples per class is set as $M_q = 19$ [34].

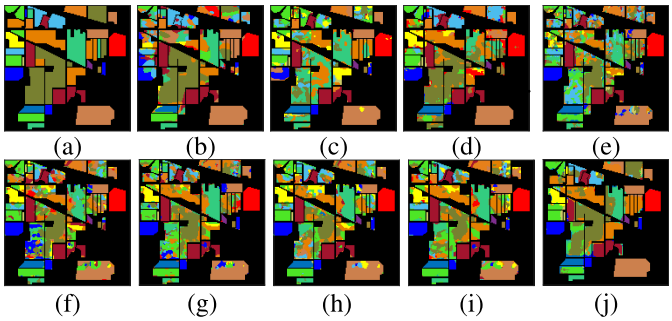


Fig. 9. Classification map on the IP dataset. (a) Ground truth. (b) HybirdSN (66.17%). (c) SSRN (61.85%). (d) A2S2K (69.08%). (e) SSCL (55.00%). (f) DFSL (59.70%). (g) DCFSL (65.12%). (h) CMFSL (66.18%). (i) Gia-CFSL (67.02%). (j) RPCL-FSL (78.44%).

In the meta-test phase, all samples of the testing dataset are fed into the trained feature extractor to obtain discriminative features, and then, the NN is used to classify the unlabeled samples. The overall accuracy (OA), the average accuracy (AA), and the κ coefficient are used to evaluate the classification performance of different methods. We randomly run each experiment ten times and report the averaged results.

All experiments are performed on a 2.99-GHz CPU, GeForce RTX 2080 Ti GPU, and 64-GB memory computer with Pytorch. The Adam is used as the optimizer, and the number of training iterations is set to 3000. The learning rate is set to 0.001. The window size of the input patch is set to 9×9.

C. Comparing With Other Methods

Table V shows the classification results of different methods on the IP dataset. It can be seen that the proposed RPCL-FSL method shows the best results in terms of OA, AA, and κ coefficient and exceeds the second-best Gia-CFSL algorithm by 11.42%, 3.47%, and 12.24%, respectively. For Classes 1, 7, 9, and 16 with limited samples, FSL methods provide excellent results. In particular, Gia-CFSL and our RPCL-FSL correctly classify all samples in these four classes. This demonstrates that the FSL strategy is effective for the small-sample classification. On large Classes 2 and 14 (i.e., "Corn-notill" and "Woods") with more than 1000 samples, the proposed RPCL-FSL improves the existing FSL methods by about 30% and 15%, respectively. This may be because the learned prototypes using triple constraints are more accurate, and hence, the feature extractor is more discriminative.

Fig. 9 visually shows the classification maps of different methods. It is clear that the proposed RPCL-FSL yields much better results on Classes 2, 14, and 15 in brown.

Table VI lists the experimental results on the UP dataset. Compared with other methods, the proposed RPCL-FSL also shows the best overall results with an OA of 86.44%. As there are only 45 training samples, deep learning methods show bad results with OA less than 80%. The FSL-based methods improve the deep learning methods to a certain extent. The proposed RPCL-FSL shows the best results on Classes 4 and 8. Fig. 10 visually shows the classification maps of different

TABLE V
CLASSIFICATION RESULTS (%) ON THE IP DATASET

Class	HybirdSN	SSRN	A2S2K	SSCL	DFSL	DCFSL	CMFSL	Gia-CFSL	RPCL-FSL
1	89.24	80.00	36.53	87.18	80.56	90.91	93.94	**100**	**100**
2	63.14	44.10	68.15	36.31	29.20	54.98	48.55	66.76	**89.45**
3	61.24	48.64	65.64	54.56	79.02	72.46	**83.48**	79.78	67.87
4	43.57	50.86	49.04	34.78	87.22	83.93	**93.30**	75.78	87.06
5	50.39	92.27	86.01	73.53	85.20	**92.98**	92.13	90.83	92.67
6	78.15	96.69	**97.58**	89.21	88.75	93.31	97.07	97.07	82.48
7	78.27	47.91	19.00	**100**	**100**	**100**	**100**	**100**	**100**
8	90.14	99.36	47.06	96.82	**100**	98.92	99.57	99.35	99.36
9	27.41	28.30	25.86	**100**	**100**	**100**	**100**	**100**	**100**
10	55.32	45.71	64.71	49.74	54.16	67.88	**67.99**	67.95	64.73
11	**82.34**	71.40	74.96	33.17	42.54	40.25	37.88	37.81	65.63
12	45.17	45.90	**56.76**	43.17	37.91	29.31	49.66	49.22	51.53
13	90.14	64.93	76.92	**99.49**	97.95	97.40	96.35	99.48	97.00
14	69.24	95.95	92.32	74.40	73.39	83.39	84.19	71.22	**98.88**
15	31.54	21.78	51.46	65.96	58.24	66.49	64.08	**81.18**	75.32
16	25.84	24.30	57.14	96.51	**100**	**100**	98.75	**100**	**100**
OA	66.17	61.85	69.08	55.00	59.70	65.12	66.18	67.02	**78.44**
AA	61.32	59.88	60.57	70.93	75.88	79.51	81.68	82.28	**85.75**
Kappa	65.57	57.04	65.21	49.92	55.17	60.97	62.36	63.32	**75.56**

TABLE VI
CLASSIFICATION RESULTS (%) ON THE UP DATASET

Class	HybirdSN	SSRN	A2S2K	SSCL	DFSL	DCFSL	CMFSL	Gia-CFSL	RPCL-FSL
1	60.75	**99.98**	92.51	56.81	98.24	83.69	71.97	74.34	87.57
2	81.27	98.00	84.82	64.51	**98.45**	80.52	92.01	91.32	92.50
3	34.85	49.12	55.66	79.14	48.04	**90.60**	80.58	89.64	67.62
4	92.47	51.45	69.45	73.50	50.94	88.82	89.21	84.36	**94.44**
5	**100**	**100**	96.32	**100**	99.92	**100**	**100**	**100**	99.02
6	**92.47**	91.57	57.00	62.09	77.96	90.73	76.83	88.26	59.45
7	**100**	79.35	40.64	91.78	73.53	100	99.39	86.47	85.88
8	45.27	42.47	45.85	14.38	84.24	55.75	71.21	40.05	**86.89**
9	99.14	11.24	68.21	92.47	75.68	**99.68**	99.25	99.57	99.36
OA	74.28	80.27	68.50	62.65	78.58	82.79	85.20	83.64	**86.44**
AA	89.46	69.24	67.83	70.52	78.56	**87.75**	86.72	83.78	85.86
Kappa	70.34	74.55	63.66	53.93	78.50	77.97	80.53	78.56	**82.05**

methods, where RPCL-FSL shows better overall results than other methods, especially on Class 2 "Meadows" in yellow.

Table VII lists the classification results on the SA dataset. As can be seen from Table VII, in the case of small samples, although some deep learning algorithms can achieve good results due to the regular and scattered distribution of different categories, our RPCL-FSL still maintains advantages in three indicators. In terms of OA, AA, and κ coefficient, it surpasses the second-best method by 2.24%, 1.20%, and 2.43%, respectively. For the SA dataset, the two large Classes 8 "Grapes_untrained" and 15 "Vinyard_untrained" are spatially adjacent and spectrally similar and hence are easy to be confused. Some deep learning and FSL methods, such as HybirdSN, SSRN, DCFSL, and GiaCFSL, show very poor results on these two classes. Some other methods, such as A2S2K, DFSL, and CMFSL, provide acceptable results on only one of these two classes at the cost of sacrificing the accuracy on another class. However, our RPCL-FSL provides the best results on these two large classes. Fig. 11 visually shows the classification maps of different methods. It is clear that our RPCL-FSL yields much better results on the Classes 8 "Grapes_untrained" (brown color) and 15 "Vinyard_untrained" (red) located in the top-left corner.

To investigate the performance of the proposed method in the case of different numbers of labeled samples, we randomly select three, five, seven, and nine labeled samples from the testing dataset for FSL and the rest samples are used for testing. The proposed RPCL-FSL method is

TABLE VII
CLASSIFICATION RESULTS (%) ON THE SA DATASET

Class	HybirdSN	SSRN	A2S2K	SSCL	DFSL	DCFSL	CMFSL	Gia-CFSL	RPCL-FSL
1	**100**	75.12	**100**	**100**	99.70	**100**	**100**	**100**	99.15
2	**100**	74.52	99.80	99.84	99.60	99.95	98.63	99.62	**100**
3	96.41	**100**	96.40	89.15	98.83	**100**	97.76	94.87	92.44
4	97.12	98.42	83.27	83.74	98.20	92.43	98.34	98.27	**99.42**
5	86.42	53.14	89.22	97.49	73.19	94.42	**98.58**	97.75	91.58
6	**100**	77.51	99.97	95.35	97.60	99.90	99.95	99.97	97.64
7	**100**	**100**	98.48	97.06	99.22	**100**	99.86	**100**	99.97
8	78.45	62.47	89.48	62.25	80.24	65.39	63.20	70.81	**92.70**
9	89.42	99.41	99.45	99.94	**100**	97.64	99.61	**100**	99.90
10	94.25	**96.12**	90.05	94.20	74.83	88.78	95.93	92.54	85.97
11	95.12	55.47	97.06	**99.53**	95.30	98.59	97.92	95.00	97.08
12	91.54	92.75	97.41	79.72	81.12	95.63	94.27	96.20	**99.32**
13	99.42	96.24	97.09	99.89	**100**	99.01	**100**	**100**	99.78
14	60.12	47.53	81.96	96.53	32.83	95.48	**100**	97.46	98.21
15	53.39	65.71	60.39	82.83	57.72	71.84	85.99	73.17	**90.52**
16	**100**	87.45	99.48	91.68	95.62	95.11	94.22	97.78	88.85
OA	83.24	76.17	88.08	86.21	84.76	87.11	89.41	89.12	**91.65**
AA	90.10	80.11	92.47	91.83	86.50	93.38	95.27	94.59	**95.79**
Kappa	81.20	77.04	86.80	84.74	83.07	85.74	88.28	87.92	**90.71**

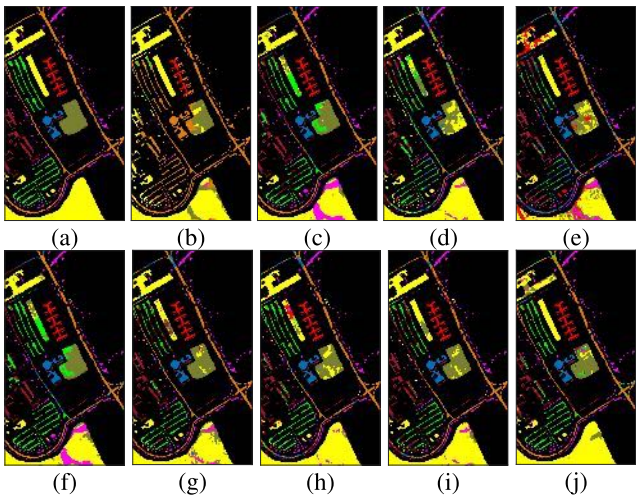


Fig. 10. Classification map on the UP dataset. (a) Ground truth. (b) HybirdSN (74.28%). (c) SSRN (80.27%). (d) A2S2K (68.50%). (e) SSCL (62.65%). (f) DFSL (78.58%). (g) DCFSL (82.79%). (h) CMFSL (85.20%). (i) Gia-CFSL (83.64%). (j) RPCL-FSL (86.44%).

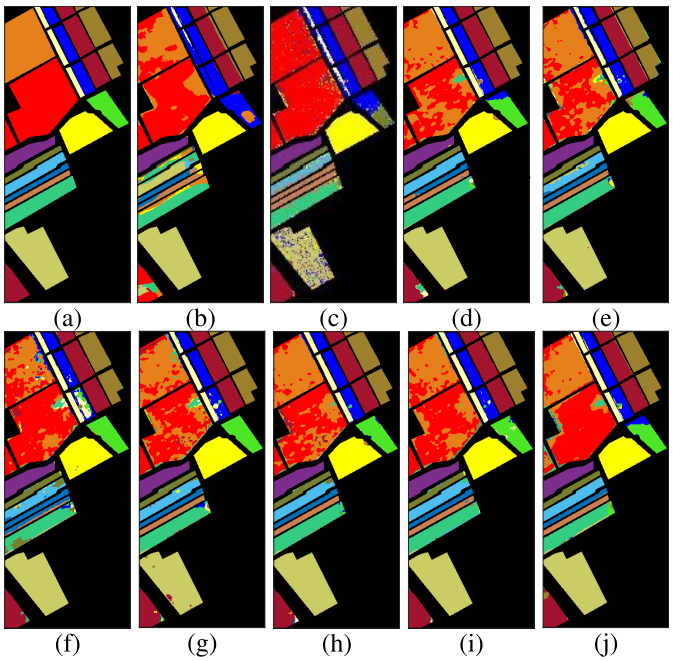


Fig. 11. Classification map on the SA dataset. (a) Ground truth. (b) HybirdSN (83.24%). (c) SSRN (76.17%). (d) A2S2K (88.08%). (e) SSCL (86.21%). (f) DFSL (84.44%). (g) DCFSL (87.11%). (h) CMFSL (89.41%). (i) Gia-CFSL (89.12%). (j) RPCL-FSL (91.65%).

compared with four other FSL-based methods, i.e., DFSL, DCFSL, CMFSL, and Gia-CFSL. The OA of five methods versus a different number of labeled samples is shown in Fig. 12. It can be seen that the OA of different methods increases as the number of labeled samples increases on the three datasets. Our proposed RPCL-FSL outperforms other algorithms consistently in different numbers of labeled samples.

D. Ablation Analysis

The proposed RPCL-FSL method imposes triple constraints on the prototypes, i.e., CL module, SC module, and CC module, and employs an FT strategy to alleviate the domain shift between the training and testing datasets. To verify the effectiveness of different modules of RPCL-FSL, we perform an ablation analysis by removing each module from the entire framework.

There are four variants in the ablation analyses: 1) "RPCL-FSL (no FT)"—training the network without the fusion

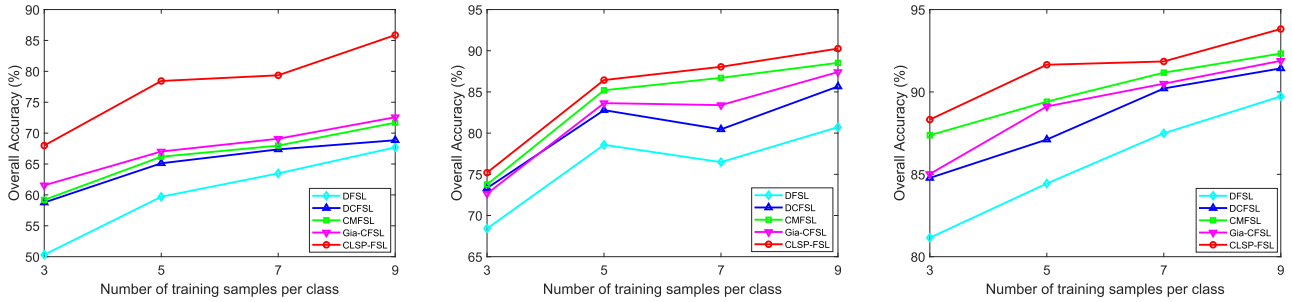


Fig. 12. OA (%) of different methods with different numbers of labeled samples per class in the testing set: (a) IP, (b) UP, and (c) SA.

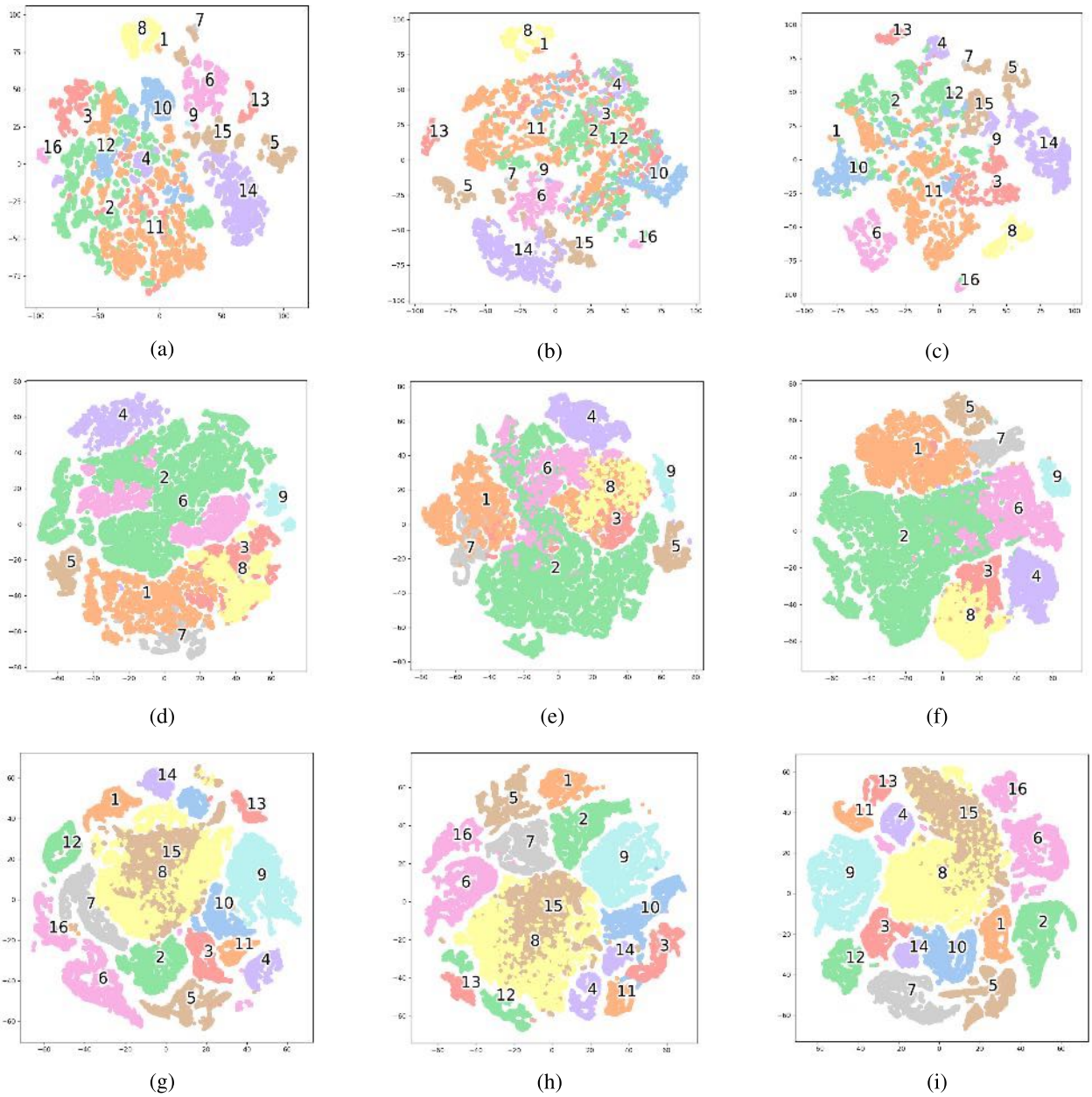


Fig. 13. 2-D feature visualization on three datasets. (a) DCFSL features on IP. (b) Gia-CFSL features on IP. (c) RPCL-FSL features on IP. (d) DCFSL features on UP. (e) Gia-CFSL features on UP. (f) RPCL-FSL features on UP. (g) DCFSL features on SA. (h) Gia-CFSL features on SA. (i) RPCL-FSL features on SA.

strategy; 2) "RPCL-FSL (no CL)"—deleting the CL module and corresponding loss; 3) "RPCL-FSL (no SC)"—deleting the SC module and corresponding loss; and 4) "RPCL-FSL (no CC)"—deleting the CC module and corresponding loss. The OA different ablation models are listed in Table VIII. It can be seen that the FT strategy can reduce the distribution

TABLE VIII
ABLATION COMPARISON OF EACH VARIANT OF RPCL-FSL

	RPCL-FSL (NO FT)	RPCL-FSL (NO CL)	RPCL-FSL (NO SC)	RPCL-FSL (NO CC)	RPCL-FSL
IP	64.37%	75.25%	76.33%	77.85%	**78.44%**
UP	71.28%	82.92%	85.24%	84.79%	**86.44%**
SA	81.53%	89.19%	90.27%	90.16%	**91.65%**

differences between the training and testing datasets and improves the model performance by about 10%–15%. In the FT framework, three losses (i.e., CL, SC, and CC losses) all contribute to the final classification. In detail, the CL module imposes an internal constraint on the prototypes, which can improve the OA by about 2%–4%. By imposing external constraints on the prototypes, the SC and CC modules can improve the OA by about 2%. By considering all modules, the proposed RPCL-FSL provides excellent results. It is clear that all modules contribute to the final classification on three datasets.

E. Feature Visualization

In order to intuitively reflect the feature extraction performance of different methods, we use the t-distributed stochastic neighbor embedding (t-SNE) technique to show the 2-D projected features of DCFSL, Gia-CFSL, and RPCL-FSL on three datasets in Fig. 13.

On the IP dataset in Fig. 13(a)–(c), it is clear that our RPCL-FSL is significantly better than the other two algorithms. In particular, our RPCL-FSL can clearly distinguish the second category "Corn-notill," the third category "Corn-mintill," and the fourth category "Corn," which have similar spectral characteristics.

On the UP dataset, the first category "Asphalt" and the seventh category "Bitumen" are similar materials, so they have similar spectral characteristics and are difficult to distinguish. Compared with DCFSL and Gia-CFSL, our RPCL-FSL can well separate these two categories. In addition, in the third and eighth categories, our RPCL-FSL also shows relatively better results.

On the SA dataset, the eighth category "Grapes_untrained" and the fifteenth category "Vinyard_untrained" are difficult to distinguish. DCFSL and Gia-CFSL mix these two categories into one cluster, while our RPCL-FSL obviously generates two clusters.

DCFSL and Gia-CFSL use the domain adaptation strategy to solve the domain shift between the training and testing datasets. However, the training objective of FSL is different from the domain adaptation and it is very difficult to eliminate the domain shift by simply embedding a domain adaptation module into the FSL framework. Our RPCL-FSL imposes triple constraints on the prototypes to stable the prototypical network and adopts an FT strategy to alleviate the domain shift. From Fig. 13(c), (f), and (i), it can be seen that the features of RPCL-FSL are more separable than the features of other two methods. It confirms that triple constraints and fusing training can improve the feature separability in the embedding space.

TABLE IX
NUMBER OF PARAMETERS AND COMPUTATIONAL TIME OF DIFFERENT METHODS

Indian Pines	DFSL	DCFSL	CMFSL	Gia-CFSL	RPCL-FSL
Params(M)	0.13	16.28	2.37	3.31	0.26
Tr (s)	6367	1942	3541	5781	769
FE (s)	11.14	35.97	18.65	1.96	1.76
University of Pavia	DFSL	DCFSL	CMFSL	Gia-CFSL	RPCL-FSL
Params(M)	0.13	16.24	2.37	3.26	0.29
Tr (s)	6367	1696	3954	3497	459
FE (s)	40.82	30.57	22.79	7.06	6.99
Salinas	DFSL	DCFSL	CMFSL	Gia-CFSL	RPCL-FSL
Params(M)	0.13	16.29	2.37	3.31	0.33
Tr (s)	6367	1949	4217	5992	946
FE (s)	52.31	36.14	29.35	10.76	9.42

F. Analysis of Parameter and Computational Time

Table IX lists the number of parameters (M) and inference time [training time and feature extraction time (seconds)] of DFSL, DCFSL, CMFSL, Gia-CFSL, and RPCL-FSL. Compared with other FSL-based methods, our RPCL-FSL has a relatively smaller parameter size and a faster training speed.

V. CONCLUSION

In this article, we have proposed an RPCL-FSL and applied the model for small-sample HSI classification. The proposed RPCL-FSL incorporates supervised CL and prototype-based FSL into an end-to-end network and employs a fusing training strategy to alleviate the domain shift between the training and testing datasets. To solve the prototype instability problem, triple constraints (i.e., CL-, SC-, and CC-based constraints) are imposed on the prototypes of the support set. The proposed RPCL-FSL can effectively transfer the knowledge from the training dataset to the testing dataset and learn representative refined prototypes for few-shot classification. Experimental results on three datasets have verified the effectiveness of the proposed RPCL-FSL. Despite obtaining promising results, more studies are required to further improve the performance of the RPCL-FSL method. For example, a suitable weighted combination way of four losses and an effective domain adaptation strategy for different datasets deserves further investigation.

REFERENCES

[1] J. M. Bioucas-Dias, A. Plaza, G. Camps-Valls, P. Scheunders, N. M. Nasrabadi, and J. Chanussot, "Hyperspectral remote sensing data analysis and future challenges," *IEEE Geosci. Remote Sens. Mag.*, vol. 1, no. 2, pp. 6–36, Jun. 2013.

[2] A. Plaza et al., "Recent advances in techniques for hyperspectral image processing," *Remote Sens. Environ.*, vol. 113, pp. S110–S122, Sep. 2009.

[3] J. W. P. Sun, H. Li, W. Li, X. Meng, C. Ge, and Q. Du, "Low-rank and sparse representation for hyperspectral image processing: A review," *IEEE Geosci. Remote Sens. Mag.*, vol. 10, no. 1, pp. 10–43, Jun. 2021.

[4] L. Samaniego, A. Bárdossy, and K. Schulz, "Supervised classification of remotely sensed imagery using a modified k-NN technique," *IEEE Trans. Geosci. Remote Sens.*, vol. 46, no. 7, pp. 2112–2125, Jul. 2008.

[5] F. Melgani and L. Bruzzone, "Classification of hyperspectral remote sensing images with support vector machines," *IEEE Trans. Geosci. Remote Sens.*, vol. 42, no. 8, pp. 1778–1790, Aug. 2004.

[6] G. Camps-Valls, L. Gomez-Chova, J. Munoz-Mari, J. Vila-Frances, and J. Calpe-Maravilla, "Composite kernels for hyperspectral image classification," *IEEE Geosci. Remote Sens. Lett.*, vol. 3, no. 1, pp. 93–97, Jan. 2006.

[7] J. Peng, Y. Zhou, and C. L. P. Chen, "Region-kernel-based support vector machines for hyperspectral image classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 53, no. 9, pp. 4810–4824, Sep. 2015.

[8] Y. Chen, N. M. Nasrabadi, and T. D. Tran, "Hyperspectral image classification using dictionary-based sparse representation," *IEEE Trans. Geosci. Remote Sens.*, vol. 49, no. 10, pp. 3973–3985, Oct. 2011.

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

[10] J. Peng, L. Li, and Y. Y. Tang, "Maximum likelihood estimation-based joint sparse representation for the classification of hyperspectral remote sensing images," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 30, no. 6, pp. 1790–1802, Jun. 2019.

[11] J. A. Benediktsson, J. A. Palmason, and J. R. Sveinsson, "Classification of hyperspectral data from urban areas based on extended morphological profiles," *IEEE Trans. Geosci. Remote Sens.*, vol. 43, no. 3, pp. 480–491, Mar. 2005.

[12] L. Shen and S. Jia, "Three-dimensional Gabor wavelets for pixel-based hyperspectral imagery classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 49, no. 12, pp. 5039–5046, Dec. 2011.

[13] L. He, J. Li, C. Liu, and S. Li, "Recent advances on spectral–spatial hyperspectral image classification: An overview and new guidelines," *IEEE Trans. Geosci. Remote Sens.*, vol. 56, no. 3, pp. 1579–1597, Mar. 2017.

[14] 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, Sep. 2019.

[15] S. Mei, X. Li, X. Liu, H. Cai, and Q. Du, "Hyperspectral image classification using attention-based bidirectional long short-term memory network," *IEEE Trans. Geosci. Remote Sens.*, vol. 60, 2022, Art. no. 5509612.

[16] L. Wu, L. Fang, J. Yue, B. Zhang, P. Ghamisi, and M. He, "Deep bilateral filtering network for point-supervised semantic segmentation in remote sensing images," *IEEE Trans. Image Process.*, vol. 31, pp. 7419–7434, 2022.

[17] Y. Xu et al., "Dual-channel residual network for hyperspectral image classification with noisy labels," *IEEE Trans. Geosci. Remote Sens.*, vol. 60, 2022, Art. no. 5502511.

[18] Y. Chen, Z. Lin, X. Zhao, G. Wang, and Y. Gu, "Deep learning-based classification of hyperspectral data," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 7, no. 6, pp. 2094–2107, Jun. 2014.

[19] Y. Chen, H. Jiang, C. Li, X. Jia, and P. Ghamisi, "Deep feature extraction and classification of hyperspectral images based on convolutional neural networks," *IEEE Trans. Geosci. Remote Sens.*, vol. 54, no. 10, pp. 6232–6251, Jul. 2016.

[20] Z. Zhong, J. Li, Z. Luo, and M. Chapman, "Spectral–spatial residual network for hyperspectral image classification: A 3-D deep learning framework," *IEEE Trans. Geosci. Remote Sens.*, vol. 56, no. 2, pp. 847–858, Aug. 2017.

[21] S. Mei, X. Chen, Y. Zhang, J. Li, and A. Plaza, "Accelerating convolutional neural network-based hyperspectral image classification by step activation quantization," *IEEE Trans. Geosci. Remote Sens.*, vol. 60, 2022, Art. no. 5502012.

[22] S. Jia, S. Jiang, Z. Lin, N. Li, M. Xu, and S. Yu, "A survey: Deep learning for hyperspectral image classification with few labeled samples," *Neurocomputing*, vol. 448, pp. 179–204, Aug. 2021.

[23] W. Li, C. Chen, M. Zhang, H. Li, and Q. Du, "Data augmentation for hyperspectral image classification with deep CNN," *IEEE Geosci. Remote Sens. Lett.*, vol. 16, no. 4, pp. 593–597, Nov. 2018.

[24] J. M. Haut, M. E. Paoletti, J. Plaza, A. Plaza, and J. Li, "Hyperspectral image classification using random occlusion data augmentation," *IEEE Geosci. Remote Sens. Lett.*, vol. 16, no. 11, pp. 1751–1755, Nov. 2019.

[25] H. Zhang, Y. Li, Y. Jiang, P. Wang, Q. Shen, and C. Shen, "Hyperspectral classification based on lightweight 3-D-CNN with transfer learning," *IEEE Trans. Geosci. Remote Sens.*, vol. 57, no. 8, pp. 5813–5828, Aug. 2019.

[26] B. Cui, X.-M. Dong, Q. Zhan, J. Peng, and W. Sun, "LiteDepthwiseNet: A lightweight network for hyperspectral image classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 60, 2022, Art. no. 5502915.

[27] B. Liu, A. Yu, X. Yu, R. Wang, K. Gao, and W. Guo, "Deep multiview learning for hyperspectral image classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 59, no. 9, pp. 7758–7772, Sep. 2021.

[28] L. Zhao, W. Luo, Q. Liao, S. Chen, and J. Wu, "Hyperspectral image classification with contrastive self-supervised learning under limited labeled samples," *IEEE Geosci. Remote Sens. Lett.*, vol. 19, 2022, Art. no. 6008205.

[29] O. Vinyals et al., "Matching networks for one shot learning," in *Proc. NIPS*, vol. 29, 2016, pp. 1–9.

[30] J. Snell, K. Swersky, and R. Zemel, "Prototypical networks for few-shot learning," in *Proc. NIPS*, vol. 30, 2017, pp. 1–11.

[31] V. Garcia and J. Bruna, "Few-shot learning with graph neural networks," in *Proc. ICLR*, 2018, pp. 1–13.

[32] Q. Sun, Y. Liu, T.-S. Chua, and B. Schiele, "Meta-transfer learning for few-shot learning," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 403–412.

[33] R. Hou, H. Chang, B. Ma, S. Shan, and X. Chen, "Cross attention network for few-shot classification," in *Proc. NIPS*, vol. 32, 2019, pp. 1–12.

[34] B. Liu, X. Yu, A. Yu, P. Zhang, G. Wan, and R. Wang, "Deep few-shot learning for hyperspectral image classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 57, no. 4, pp. 2290–2304, Apr. 2019.

[35] K. Gao, B. Liu, X. Yu, J. Qin, P. Zhang, and X. Tan, "Deep relation network for hyperspectral image few-shot classification," *Remote Sens.*, vol. 12, no. 6, p. 923, Mar. 2020.

[36] 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.

[37] 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.

[38] Y. Zhang, W. Li, M. Zhang, S. Wang, R. Tao, and Q. Du, "Graph information aggregation cross-domain few-shot learning for hyperspectral image classification," *IEEE Trans. Neural Netw. Learn. Syst.*, early access, Jun. 30, 2022, doi: 10.1109/TNNLS.2022.3185795.

[39] J. Peng, Y. Huang, W. Sun, N. Chen, Y. Ning, and Q. Du, "Domain adaptation in remote sensing image classification: A survey," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 15, pp. 9842–9859, 2022.

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

[41] L. Fang, D. Zhu, J. Yue, B. Zhang, and M. He, "Geometric-spectral reconstruction learning for multi-source open-set classification with hyperspectral and LiDAR data," *IEEE/CAA J. Automatica Sinica*, vol. 9, no. 10, pp. 1892–1895, Oct. 2022.

[42] J. Liu, L. Song, and Y. Qin, "Prototype rectification for few-shot learning," in *Proc. ECCV*, 2020, pp. 741–756.

[43] P. Wang, J. Cheng, F. Hao, L. Wang, and W. Feng, "Embedded adaptive cross-modulation neural network for few-shot learning," *Neural Comput. Appl.*, vol. 32, no. 10, pp. 5505–5515, May 2020.

[44] G. Cheng et al., "SPNet: Siamese-prototype network for few-shot remote sensing image scene classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 60, 2022, Art. no. 5608011.

[45] C. Zhang, J. Yue, and Q. Qin, "Global prototypical network for few-shot hyperspectral image classification," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 13, pp. 4748–4759, 2020.

[46] P. Khosla et al., "Supervised contrastive learning," in *Proc. NIPS*, vol. 33, 2020, pp. 18661–18673.

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

[48] S. Hou, H. Shi, X. Cao, X. Zhang, and L. Jiao, "Hyperspectral imagery classification based on contrastive learning," *IEEE Trans. Geosci. Remote Sens.*, vol. 60, 2022, Art. no. 5521213.

[49] 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 Geosci. Remote Sens. Lett.*, vol. 17, no. 2, pp. 277–281, Jun. 2020.

[50] S. K. Roy, S. Manna, T. Song, and L. Bruzzone, "Attention-based adaptive spectral–spatial kernel ResNet for hyperspectral image classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 59, no. 9, pp. 7831–7843, Sep. 2021.

[51] K. Sun et al., "A robust and efficient band selection method using graph representation for hyperspectral imagery," *Int. J. Remote Sens.*, vol. 37, no. 20, pp. 4874–4889, 2016.

Quanyong Liu received the B.S. degree in mathematics and applied mathematics from the Faculty of Mathematics and Statistics, Hubei University, Wuhan, China, in 2021, where he is currently pursuing the M.S. degree in applied mathematics.

His research interests include machine learning and hyperspectral image processing.

Jiangtao Peng (Senior Member, IEEE) received the B.S. degree in information and computing science and the M.S. degree in applied mathematics from Hubei University, Wuhan, China, in 2005 and 2008, respectively, and the Ph.D. degree in pattern recognition and intelligent systems from the Institute of Automation, Chinese Academy of Sciences, Beijing, China, in 2011.

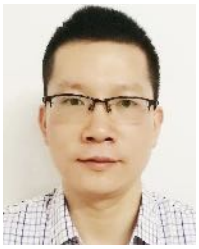
He is currently a Professor with the Faculty of Mathematics and Statistics, Hubei University. His research interests include machine learning and hyperspectral image processing.

Yujie Ning received the B.S. degree in mathematics and applied mathematics from the Faculty of Mathematics and Statistics, Hubei University, Wuhan, China, in 2021, where she is currently pursuing the Ph.D. degree in applied mathematics.

Her research interests include machine learning and hyperspectral image processing.

Na Chen received the B.S. degree in information and computing science and the M.S. degree in applied mathematics from Hubei University, Wuhan, China, in 2006 and 2009, respectively, and the Ph.D. degree in computational mathematics from the Huazhong University of Science and Technology, Wuhan, in 2012.

She is currently an Associate Professor with the Faculty of Mathematics and Statistics, Hubei University. Her research interests include statistical learning theory and image processing.



Weiwei Sun (Senior Member, IEEE) received the B.S. degree in surveying and mapping and the Ph.D. degree in cartography and geographic information engineering from Tongji University, Shanghai, China, in 2007 and 2013, respectively.

From 2011 to 2012, he worked at the Department of Applied Mathematics, University of Maryland at College Park, College Park, MD, USA, as a Visiting Scholar with the famous Prof. John Benedetto to study on the dimensionality reduction of hyperspectral image. From 2014 to 2016, he held a post-doctoral position at the State Key Laboratory for Information Engineering in Surveying, Mapping and Remote Sensing (LIESMARS), Wuhan University, Wuhan, China, to study intelligent processing in hyperspectral imagery. From 2017 to 2018, he worked as a Visiting Scholar at the Department of Electrical and Computer Engineering, Mississippi State University, Mississippi State, MS, USA. He is currently a Full Professor with Ningbo University, Ningbo, Zhejiang, China. He has published more than 70 journal articles. His research interests include hyperspectral image processing with manifold learning, anomaly detection, and target recognition of remote sensing imagery using compressive sensing.

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

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

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

Yicong Zhou (Senior Member, IEEE) received the B.S. degree from Hunan University, Changsha, China, in 1992, and the M.S. and Ph.D. degrees from Tufts University, Medford, MA, USA, in 2008 and 2010, respectively, all in electrical engineering.

He is currently an Associate Professor and the Director of the Vision and Image Processing Laboratory, Department of Computer and Information Science, University of Macau, Macau, China. His research interests include image processing, computer vision, machine learning, and multimedia security.

Dr. Zhou is a Senior Member of the International Society for Optical Engineering (SPIE). He was a recipient of the Third Price of Macau Natural Science Award in 2014. He is the Co-Chair of the Technical Committee on Cognitive Computing of the IEEE Systems, Man, and Cybernetics Society. He serves as an Associate Editor for IEEE TRANSACTIONS ON NEUTRAL NETWORKS AND LEARNING SYSTEMS, IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS FOR VIDEO TECHNOLOGY, IEEE TRANSACTIONS ON GEOSCIENCE AND REMOTE SENSING, and four other journals.