



## Article

# Weighted Contrastive Prototype Network for Few-Shot Hyperspectral Image Classification with Noisy Labels

Dan Zhang <sup>1,2</sup>, Yiyuan Ren <sup>3</sup>, Chun Liu <sup>3</sup>, Zhigang Han <sup>4</sup> and Jiayao Wang <sup>4,\*</sup>

<sup>1</sup> The College of Surveying and Mapping Engineering, Yellow River Conservancy Technical Institute, Kaifeng 475004, China; 2004800320@yrcti.edu.cn

<sup>2</sup> Henan Province Surveying and Mapping Real Scene 3D Technology Engineering Research Center, Yellow River Conservancy Technical Institute, Kaifeng 475004, China

<sup>3</sup> School of Computer and Information Engineering, Henan University, Zhengzhou 450046, China; renyiyuan@henu.edu.cn (Y.R.); liuchun@henu.edu.cn (C.L.)

<sup>4</sup> College of Geography and Environmental Science, Henan University, Zhengzhou 450046, China; zgihan@henu.edu.cn

\* Correspondence: wjy@henu.edu.cn

**Abstract:** Few-shot hyperspectral image classification aims to develop the ability of classifying image pixels by using relatively few labeled pixels per class. However, due to the inaccuracy of the localization system and the bias of the ground survey, the potential noisy labels in the training data pose a very significant challenge to few-shot hyperspectral image classification. To solve this problem, this paper proposes a weighted contrastive prototype network (WCPN) for few-shot hyperspectral image classification with noisy labels. WCPN first utilizes a similarity metric to generate the weights of the samples from the same classes, and applies them to calibrate the class prototypes of support and query sets. Then the weighted prototype network will minimize the distance between features and prototypes to train the network. WCPN also incorporates a weighted contrastive regularization function that uses the sample weights as gates to filter the fake positive samples whose labels are incorrect to further improve the discriminative power of the prototypes. We conduct experiments on multiple hyperspectral image datasets with artificially generated noisy labels, and the results show that the WCPN has excellent performance that can sufficiently mitigate the impact of noisy labels.

**Keywords:** hyperspectral image (HSI); few-shot learning; prototype network; contrastive learning; noisy labels



**Citation:** Zhang, D.; Ren, Y.; Liu, C.; Han, Z.; Wang, J. Weighted Contrastive Prototype Network for Few-Shot Hyperspectral Image Classification with Noisy Labels. *Remote Sens.* **2024**, *16*, 3527. <https://doi.org/10.3390/rs16183527>

Academic Editor: Andrea Garzelli

Received: 2 August 2024

Revised: 9 September 2024

Accepted: 11 September 2024

Published: 23 September 2024



**Copyright:** © 2024 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

## 1. Introduction

Hyperspectral images (HSIs) are three-dimensional cubes composed of hundreds of spectral channels. Each pixel of the images is a high-dimensional spectral vector. With rich spatial and spectral information, HSIs can reflect the radiometric properties and spatial geometric relationships of ground targets simultaneously. This results in HSIs that can capture more subtle differences between ground targets, and have great application values in various fields such as environmental monitoring [1,2], precision agriculture [3,4], and military applications [5,6].

HSI classification is to predict the classes of each pixel in the images, which is one of the important tasks in remote sensing applications. With the wide use of deep learning, significant improvements have been made in the accuracy of HSI classification [7,8]. In general, existing HSI classification methods can be divided into two categories, i.e., pixel-wise methods and spectral-spatial joint methods. The former focus on the spectral vectors behind each pixel, and apply the classifiers directly to them to predict their classes. In contrast, the latter consider the spectral information contained in the vectors but also the local spatial information around each pixel. For example, by cutting the fixed-size

(e.g.,  $9 \times 9$ ) patches centering around each pixel from HSIs and taking them as samples to be classified, the spectral–spatial joint methods can capture the spectral–spatial fusion features and have shown great advantages in HSI classification.

As it is usually expensive and time-consuming to obtain a large number of labeled samples, which are often required by current deep learning methods, few-shot learning methods have been widely used for HSI classification in recent years. Few-shot learning aims to develop the ability of identifying new samples of some classes by using only few labeled samples per class. To achieve this goal, current few-shot learning methods usually train a model with available source datasets which already consist of abundant labeled samples and then fine-tune the model with these few-shot labeled samples in the target dataset. Based on the classical few-shot learning methods such as prototype network [9] and relation network [10], a series of few-shot methods have been designed for HSI classification. And considering that the source and target datasets are from different domains, many cross-domain few-shot methods including DCFSL [11], Gia-FSL [12], and RPCL-FSL [13] have also been proposed.

However, existing few-shot methods for HSI classification have assumed that the labels of all training samples are correct. These methods will suffer from performance degradation if there are noisy labels in the training data. Noisy labels are the labels that are mistakenly assigned to samples. For HSIs, accurate annotation may be difficult to achieve due to the inaccuracy of the localization system and the bias of the ground survey [14,15]. Also, the annotator may make mistakes when confronted with certain ambiguous samples. All these make it difficult to avoid noisy labels in HSI classification in practice.

The noisy labels problem has been investigated by many works in computer vision applications [16,17]. There are two kinds of solutions: one is to detect the noisy labels and clean the samples before training [18], and the other is to train a robust model directly on the corrupted dataset [19]. Due to the differences between hyperspectral images and natural images, these solutions cannot be directly applied to HSI classification. But by following the above two routes to address noisy labels, some methods have been proposed for HSI classification with noisy labels. For example, Tu et al. [20] proposed the density peak noisy label detection method to identify noisy labels and eliminate anomalous samples from training datasets. In contrast, Xu et al. [15] presented the dual-channel residual network to deal with noisy labels at the model level. But how to address the noisy labels for few-shot HSI classification remains the issue to be resolved.

To address HSI classification with noisy labels under the few-shot setting, this paper proposes a Weighted Contrastive Prototype Network (WCPN). While combining few-shot learning and contrastive learning to solve few-shot HSI classification, the main idea of the proposed method is to identify the potential noisy samples and assign different weights to them to balance their impact. All samples are passed through an adaptive mapping module and a deep 3D residual network to extract the embedding features, and then the proposed method employs a metric function to estimate the weights of each sample by utilizing the similarities between the samples. These weights are then used for the computation of the prototypes of both support and query sets, to calibrate the prototypes at the presence of noisy labels. In order to further improve the self-calibration capability of the model to reduce the impact of noisy labels, we design a regularized contrastive loss function that uses the estimated weights as gates to constrain the model. With these improvements, we aim to make the clean samples closer, and keep clean and noisy samples farther apart. The main contributions of the proposed WCPN are as follows.

1. To the best of our knowledge, a prototype network for handling noisy labels under few-shot setting has been introduced to the HSI classification for the first time. We provide better anti-noise performance than existing methods for few-shot HSI classification.
2. Our proposed weighted prototype network utilizes the weights calculated from similarities between samples to calibrate the prototypes and bring the samples closer to their clean prototypes at the presence of noisy labels.

3. We make full use of the noise information contained in the similarity weights of different samples and propose a new contrastive regularization function. This function can further constrain the model to reduce the impact of noisy samples and learn a clean feature representation.

## 2. Background and Related Work

The work in this paper involves few-shot learning for HSI classification, HSI classification with noisy labels, and contrastive learning with noisy labels. In this section, we give a brief introduction about these related works.

### 2.1. Few-Shot Learning for HSI Classification

As mentioned earlier, few-shot learning aims to solve the tasks where relatively few labeled samples are available. It has aroused much attention in a wide range of applications [21,22], and this research focuses only on its use for HSI classification.

Current works for few-shot HSI classification mainly follow the metric-based learning method which strives to learn a distance function that can measure the similarity between samples. Such a distance function is expected to have good generalization ability so that it can be well migrated to these samples in the target tasks. The classes of unlabeled samples in the target tasks can then be predicted based on their similarity with the given few labeled samples or their prototypes. By following the classical metric-based few-shot learning models, many methods have been presented for few-shot HSI classification. For example, based on the twin network [23], which receives a pair of samples and generates their similarity, ASSP-SCNN replaced the feature extractor with 3D CNN for HSI classification [24]. Based on the prototype network which predicts the sample classes according to their distance to different prototypes [9], DFSL [25] took a deep 3D residual network to learn the metric space in order to maximize class discriminability and separability; SSPN [26] used the local pattern coding to extract better features for HSI classification; HSEMD-Net [27] used the Earth Mover's distance instead of the Euclidean distance to improve the model performance. Based on relational network, which is an improved version of prototype network [10], RN-FSC [28] and RL-Net [29] follow the meta-learning to train the model for few-shot HSI classification.

The above works for few-shot HSI classification usually take the source datasets which already consist of lots of labeled samples to train the model and learn a suitable distance function, and then transfer it to the target dataset. They often assume that the source datasets and the target datasets are from the same domains. To relax this assumption and address the cross-domain few-shot HSI classification, DCFSL [11] followed the adversarial way to train the model with the aim to obtain domain-independent features; SSFT [30] adopted a feature-wise transformation module to extract more generalized features; CMFSL [31] used the modules of spectral prior-based refinement and a lightweight cross-scale convolution to increase the feature extraction ability under few-shot setting; Gia-CFSL [12] used a domain alignment strategy to suppress domain bias; GPN [32] used a global prototype strategy to train the network, where the global prototypes are continuously updated during iterative training; RPCL-FSL [13] integrated supervised contrastive learning into few-shot HSI classification, in order to obtain better prototypes and improve the classification accuracy. Liu et al. [33] incorporated contrastive learning and a transformer-based cross-attention module into few-shot HSI classification to enhance multi-level sample relations and improve classification performance.

Different from these works, the proposed method in this paper aims to explore the similarities between the samples themselves, and study how to utilize them to identify the noisy samples and reduce their impact on cross-domain few-shot HSI classification.

### 2.2. HSI Classification with Noisy Labels

The majority of existing research approaches for learning with noisy labels in the literature reduce the impacts of noise by creating robust loss functions [34,35], a noise

transition matrix [36,37], confident examples selection [38,39], reweighting examples [40,41], introducing regularization [42,43], and generating pseudo labels [44,45]. Specifically for few-shot learning, the existing methods focus on selecting credible samples from the model level. Mazumder et al. proposed RNNP [46] which combined data augmentation with k-means to produce refined prototypes. RapNets [47] utilized a BiLSTM-based attention module to overcome representation with noisy labels. Liang et al. proposed TraNFS [48] utilizing the Transformer's attention mechanism to trade-off between noisy samples and correct samples. These methods can mitigate the effects of noisy labels somewhat, but mainly focus on the noisy samples in the support set.

In the field of HSI, many methods have been also proposed to address the noisy label problem by detecting and removing noisy labels before training or directly training robust models with noisy data. Among the methods for detecting and removing noisy labels, Jiang et al. [49] proposed RLPA to construct a spectral-spatial probabilistic transferring matrix, and utilized hyperpixel-constrained random labels to reduce the noisy labels. Tu et al. proposed an SDP method [50] based on spatial density peak clustering and a SPWD method [14] incorporating super-pixel weighted distance to detect mislabeled samples in the training set. Among the methods for training robust models on noisy data, Jiang et al. proposed MSSAs [51] that used spectral and spatial similarity to construct affinity graphs to regularize the process of noisy labels cleaning, thus transforming noisy labels cleaning into an optimization problem with graph constraints. Xu et al. proposed a two-channel residual network DCRN [15] to reduce the impact of noisy labels by utilizing a noise robust loss function to detect and reject these abnormal samples. All of the above methods mitigate the impact of noisy labels in HSIs to a certain extent. Different from these works, the proposed method in this paper focuses on how to address the noisy labels and train a robust model for HSI classification in the few-shot setting.

### 2.3. Contrastive Learning with Noisy Labels

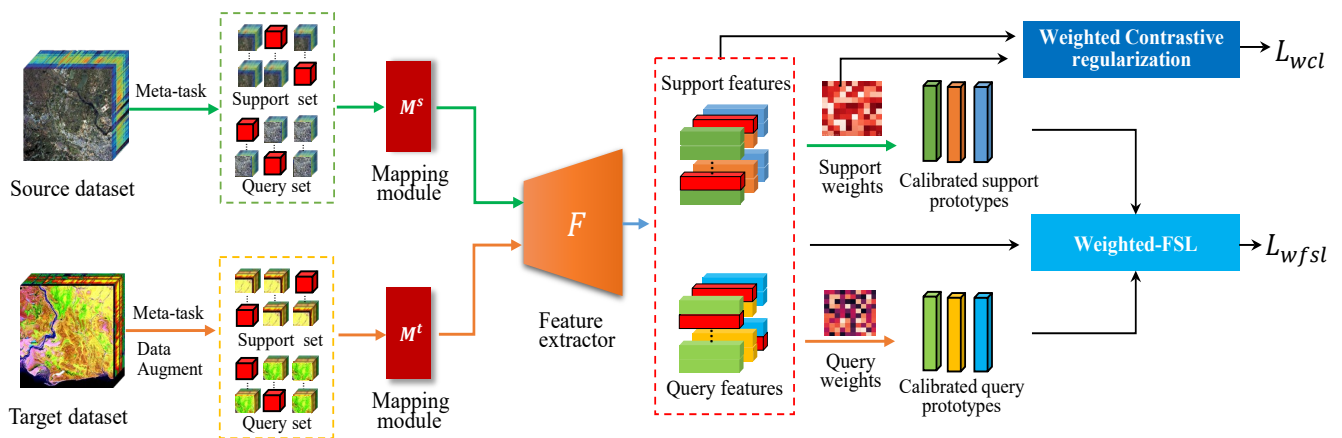
Contrastive learning [52,53] has demonstrated impressive results not only in representation learning but also in various downstream tasks, where the learned representations have shown strong generalization capabilities. The essence of contrastive learning lies in learning to differentiate between similar and dissimilar samples by contrasting their representations. In particular, by maximizing the agreement between positive samples and minimizing it for negative samples, the model is able to learn rich, meaningful feature embeddings from data. One specific variant of contrastive learning is supervised contrastive learning [54], which extends this concept by incorporating class labels. The objective of supervised contrastive learning is to ensure that samples from the same class are pulled closer together in the representation space, while samples from different classes are pushed apart. This label-guided approach further enhances the discriminative power of the learned representations, making it particularly effective in scenarios where class distinctions are crucial.

The noisy labels problem also has much impact on the supervised contrast learning. To address this issue, there are some methods proposed by now. For example, the methods of ProtoMix [55] and NGC [56] perform pseudo-label generation and supervised contrastive learning against noisy labels. Sel-CL proposed by Li et al. [57] identifies plausible examples for constructing confidence pairs by measuring the consistency between learning and given labels; Yi et al. [58] combined the non-robustness of cross-entropy loss with a novel contrast regularization function so that noisy data do not dominate the learning of features. Our work takes full advantage of contrast learning by considering the different weights of different samples in the presence of noisy samples, thereby providing better regularization of features.

## 3. The Proposed Method

The proposed WCPN solves the task of few-shot HSI classification with noisy labels. The framework of the proposed method is shown in Figure 1. In an episodic manner of

meta-learning, firstly, the support set and query set of a task are constructed from the source dataset and target dataset. Considering the presence of noisy labels, there are some noisy samples in the constructed support and query set, which are indicated by the red squares in Figure 1. Then, two trainable adaptive mapping modules are used to unify the spectral dimensions of the samples from the source dataset and target dataset, and a 3D residual module is taken to extract the features of all the samples. Following that, the weights of these samples in the support and query set are calculated, respectively, and the class prototypes of the support and query set are obtained accordingly. Finally, the weighted few-shot learning loss and weighted contrastive loss are derived to update the model. The model is trained alternately with the tasks constructed from the source and target datasets. When testing, the KNN classifier is used once the features of the support and query samples are obtained.



**Figure 1.** Flowchart of the proposed WCPN, which consists of two meta-training processes running alternately on the source and target datasets. In the meta-training process, the support and query sets are first constructed, where the red squares are the noisy samples. Then two trainable adaptive mapping modules are used to unify the spectral dimensions of the source and target datasets and a feature extractor is adopted to extract the features of the support and query samples. After that, calibrated prototypes are obtained by calculating the weights of each sample in the support and query sets. Lastly, the weighted few-shot learning loss and weighted contrastive regularization loss are used to update the entire model.

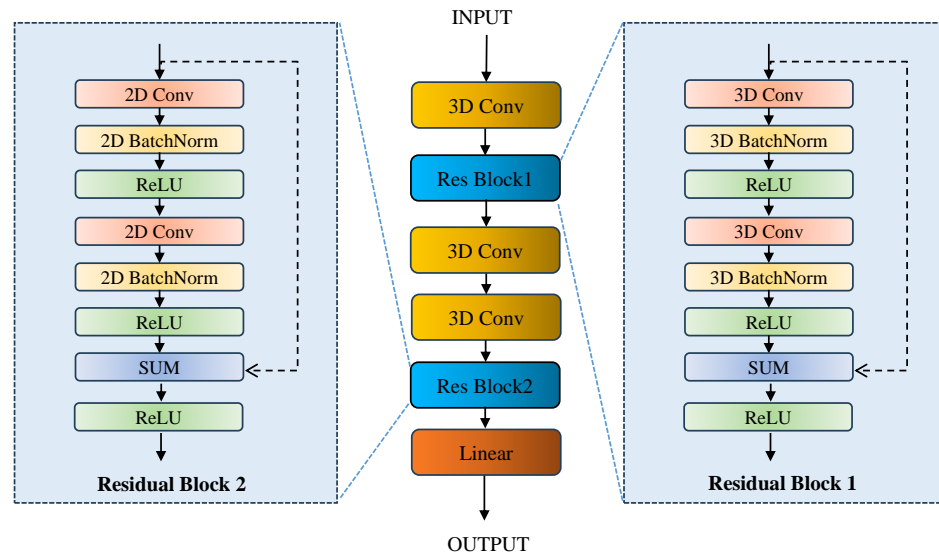
### 3.1. Meta-Learning and Feature Extraction

For few-shot learning, there are two datasets given: the source dataset  $D_s$  with  $C_s$  classes and the target dataset  $D_t$  with  $C_t$  classes. It is worth noting that there are only few labeled samples in  $D_t$  and all the unlabeled samples are to be classified. The  $C_t$  is also smaller than  $C_s$ . In the training phase, a task which consists of the support and query sets is constructed from the dataset in each episode. There are  $N$ -way  $(K + M)$ -shot samples randomly selected from the dataset for constructing the task. The  $N$ -way refers to the number of classes selected, which is often set to the number of classes of the dataset.  $K$  is the number of support samples per class and  $M$  is the number of query samples per class. That is, there are  $N \times K$  samples to form the support set  $S = \{(x_i, y_i)\}_{i=1}^{N \times K}$ , and  $N \times M$  samples to form the query set  $Q = \{(x_j, y_j)\}_{j=1}^{N \times M}$ . In this paper,  $K$  is set to 5 and  $M$  is set to 15.

To alleviate the problem of domain shift between  $D_s$  and  $D_t$ , we train the model alternately with labeled samples in  $D_s$  and  $D_t$ . However, one problem with alternate training is that the spectral dimensions of the samples from source and target datasets are different. In order to unify the dimensionality, two mapping modules are used which are implemented with a two-dimensional convolutional layer consisting of  $d$  convolutional kernels of size  $1 \times 1 \times ch$ , where  $ch$  and  $d$  are the dimensionality of the input and output



data, respectively. Given different patches with the size of  $9 \times 9 \times ch$ , the mapping modules transform them into the new patches with the size of  $9 \times 9 \times d$ . After that, the spatial-spectral joint features are extracted by the feature extractor module which is implemented with a deep 3D residual network. The components of the feature extractor are shown in Figure 2, and for more details of its implementation, please refer to [59].



**Figure 2.** The architecture of the feature extractor.

### 3.2. Weighting Samples with Noisy Labels

Prototype network [9] is a simple and efficient method for few-shot learning. It weights all the samples in the support set equally and generates the class prototypes by means of mean aggregation. With the class prototypes, it predicts the classes of samples according to their distance to the prototypes. However, when noisy labels are present, the generated prototypes will deviate from the true ones and the classification performance is affected. An obvious solution is to remove the noisy samples [57], but this way tends to remove the useful information brought by them at the same time. This paper proposes to weight the samples by using the similarity information among them. While suppressing the noisy samples and amplifying the clean samples, it makes full use of the noisy samples as much as possible.

To calculate the similarity among samples, the cosine metric which performs well for high-dimensional data is used in this paper. With the cosine metric, the average similarity between one sample and all other samples of the same class is computed by Equation (1).

$$a_i^{(c)} = \frac{1}{K-1} \sum_{i \neq j} \frac{h_i^{(c)} \cdot h_j^{(c)}}{\|h_i^{(c)}\| \|h_j^{(c)}\|} \quad (1)$$

where  $K$  is the number of samples per class, and  $a_i^{(c)}$  is the average similarity between the  $i$ th sample to the other samples of class  $c$ . The larger  $a_i^{(c)}$  is, the closer the sample is to other similar samples.

Then, the average similarities  $a_i^{(c)}$  of the samples from the same classes are normalized by the softmax function shown in Equation (2), and the weights  $w_i^{(c)}$  of each sample are generated.

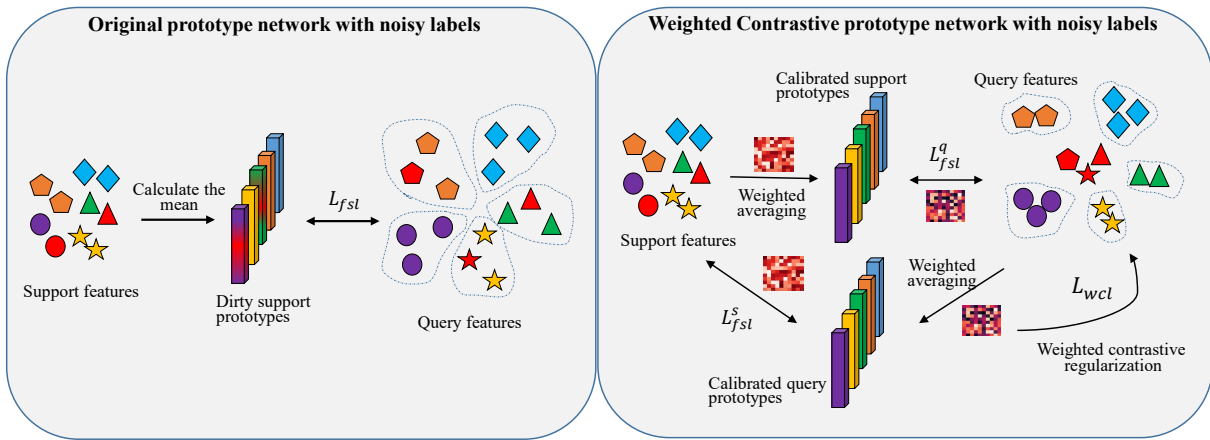
$$w_i^{(c)} = \frac{\exp(a_i^{(c)}/T)}{\sum_j \exp(a_j^{(c)}/T)} \quad (2)$$

where  $T$  is the temperature term that controls softmax diffusion. When  $T \rightarrow 0$ , the sample with the smallest distance to the other samples will be selected as the class prototype, and when  $T \rightarrow \infty$ , it is equal to the mean operation.

The similarity-based way to weight samples can measure the noise distribution of the sample labels by using their inherent characteristics. The smaller  $w$  is, the less similar the sample is to the other samples of the same classes, and the higher the confidence that the sample is considered a noisy sample. In contrast, if  $w$  is larger, the sample is cleaner.

### 3.3. Weighted Prototype Network

The prototype network trains the feature extractor  $f_\theta$  by minimizing the distance between the query set samples and the class prototypes of the support set. As shown in the left part of Figure 3, the class prototypes are computed as the mean of the features of the samples from the same classes, which can be also seen in Equation (3).



**Figure 3.** Comparison of original prototype network with noisy labels and weighted contrastive prototype network with noisy labels. Red samples indicate noisy samples. The prototypes of the original prototype network are obtained from the mean of the support features and then the model is trained by minimizing the Euclidean distance between the query features and the prototypes. The WCPN calibrates the prototypes of support and query sets using the weights of the samples according to the similarities between the samples from the same classes, and updates the contrastive regularization function by taking the weights as gates to filter the potential noisy samples when calculating the contrastive loss.

$$P^c = \frac{1}{K} \sum_{x_i \in S^c} f_\theta(x_i) \quad (3)$$

where  $P^c$  represents the prototype of class  $c$ ,  $S^c$  denotes the set of samples belonging to class  $c$  in the support set, and  $K$  is the number of support samples per class.

After obtaining the support prototypes, the embedded features of the query samples are also extracted by the feature extractor. For the query samples  $x_j \in Q$  with true label  $c_j$ , the prototype network predicts the probability that sample  $x_j$  belongs to  $c_j$  by following Equation (4).

$$p(y_j = c_j | x_j) = \frac{\exp(-d(f_\theta(x_j), P_s^{c_j}))}{\sum_{c=1}^N \exp(-d(f_\theta(x_j), P_s^c))} \quad (4)$$

where  $d(\cdot)$  denotes the Euclidean distance used to calculate the distance between features and prototypes. Following this, the few-shot learning loss can be calculated by Equation (5).

$$L_{fsl} = -\frac{1}{n_q} \sum_{j=1}^{n_q} \log p(y_j = c_j | x_j) \quad (5)$$

From Equations (3) and (4), it can be seen that in the presence of noisy support samples (e.g., red circles and red triangles shown in the left part of Figure 3), the dirty support prototypes will be generated, which will greatly affect the model's performance. In particular, there may also be noisy samples in the query set, which will also affect the training effect. To address this problem, our idea is to weight the samples in both the support set and query set, and use the weights to balance the impacts of noisy samples, as shown in the right part of Figure 3.

Specifically, both the support prototypes and the query prototypes are first computed and calibrated by using the weights obtained by Equation (2). The clean prototypes obtained after calibration are shown in Equation (6) where  $S^c$  is the set of samples of class  $c$  in either the support set or query set.

$$P^c = \frac{1}{K} \sum_{x_i \in S^c} f_{\theta}(x_i) \cdot w_i \quad (6)$$

Second, with the calibrated support prototypes and the weights of query samples, the original few-shot learning loss function of prototype network is updated so that the noisy samples do not dominate the calculation of the loss. The updated loss function is shown in Equation (7), where  $P_s^c$  denotes the class prototypes in support set and  $w_j^q$  denotes the weights of query samples.

$$L_{fsl}^q = -\frac{1}{N \times M} \sum_{j=1}^{N \times M} \log \frac{\exp(-d(f_{\theta}(x_j), P_s^{c_j})) \cdot w_j^q}{\sum_{c=1}^N \exp(-d(f_{\theta}(x_j), P_s^c))} \quad (7)$$

Third, in order to make full use of the information contained in the samples to enhance the training effect, the prediction loss of taking the query prototypes as anchors to predict the classes of the support samples is also utilized in this paper, which is similar with RPCL [13]. By using the calibrated query prototypes and the weights of support samples, the function for calculating such a loss is defined in Equation (8), where  $P_q^c$  denotes the class prototypes in the query set and  $w_j^s$  denote the weights of support samples.

$$L_{fsl}^s = -\frac{1}{N \times K} \sum_{j=1}^{N \times K} \log \frac{\exp(-d(f_{\theta}(x_j), P_q^{c_j})) \cdot w_j^s}{\sum_{c=1}^N \exp(-d(f_{\theta}(x_j), P_q^c))} \quad (8)$$

Finally, the weighted few-shot learning loss function of the prototype network is defined as Equation (9), where  $L_{fsl}^s$  denotes the loss of support set weight calibration and  $L_{fsl}^q$  denotes the loss of query set calibration.

$$L_{wfsl} = L_{fsl}^s + L_{fsl}^q \quad (9)$$

### 3.4. Weighted Contrastive Regularization Function

To further enhance the model's performance, this paper also augments the prototype network with a contrastive regularization function. It is expected to constrain the model so that samples within classes are closer together and samples between classes are farther apart. By following the supervised contrastive learning, the positive samples of a sample in the support set are the other samples of the same class, and all the remaining  $N \times (K - 1)$  are the negative samples. For a positive sample pair, the contrastive loss item can be derived by Equation (10) after obtaining the sample features.



$$L_{cl}^{ij} = -\log \frac{\exp(\text{sim}(z_i, z_j) / \tau)}{\sum_{k=1}^{N \times (K-1)} \exp(\text{sim}(z_i, z_k) / \tau)} \quad (10)$$

where  $z_i$  and  $z_j$  are the feature representations of the samples in the pair, and  $\text{sim}(z_i, z_j)$  measures the cosine similarity between them.  $\tau$  is a temperature coefficient set to 0.5 in our experiment. And  $K$  is the number of samples per class in the support set.

Considering the presence of noisy labels, when there are noisy samples in the same class, these noisy samples are actually the negative samples instead of the positive samples. Thus, it is reasonable to exclude them from the set of positive samples. Inspired by the work of Yi et al. [58], this paper uses an indicator function to filter these fake positive sample pairs, as shown by the weighted contrastive regularization function of Equation (11).

$$L_{wcl} = \frac{1}{N^2 \times K^2} \sum_{i=1}^{N \times K} \sum_{j=1, j \neq i}^{N \times K} L_{cl}^{ij} \cdot \mathbf{1}\{w_i \geq \frac{(1-\beta)}{M_s}\} \quad (11)$$

where  $\mathbf{1}\{w_i \geq \frac{(1-\beta)}{K}\}$  is the indicator function whose output is equal to 1 when  $w_i \geq \frac{(1-\beta)}{K}$  and 0 when  $w_i < \frac{(1-\beta)}{K}$ .  $\beta$  denotes a tolerance hyperparameter. Our idea behind the indicator function is that when all the samples of a class are clean and the same, the weights of the samples should be  $1/K$ . When there are noisy samples, for a sample with the weight  $w_s$ , if  $w_s > 1/K$ , it means that the sample is closer to the other samples than the average case and has a higher probability of being a clean sample. Otherwise, if  $w_s < 1/K$ , it indicates that the sample is further away from the other samples than the average case and has a higher probability of being a noisy sample. Therefore,  $1/K$  is taken as the threshold which is further balanced by the hyperparameter  $\beta$ .

Finally, the overall loss function of the proposed model is defined as Equation (12) where  $\lambda_1$  and  $\lambda_2$  are the weights given to the two kinds of losses. Both  $\lambda_1$  and  $\lambda_2$  are set to be 1 in this paper.

$$L = \lambda_1 L_{wfsl} + \lambda_2 L_{wcl} \quad (12)$$

## 4. Experiments

To validate the proposed method, several experiments have been conducted. In this section, we describe the experimental setup and results.

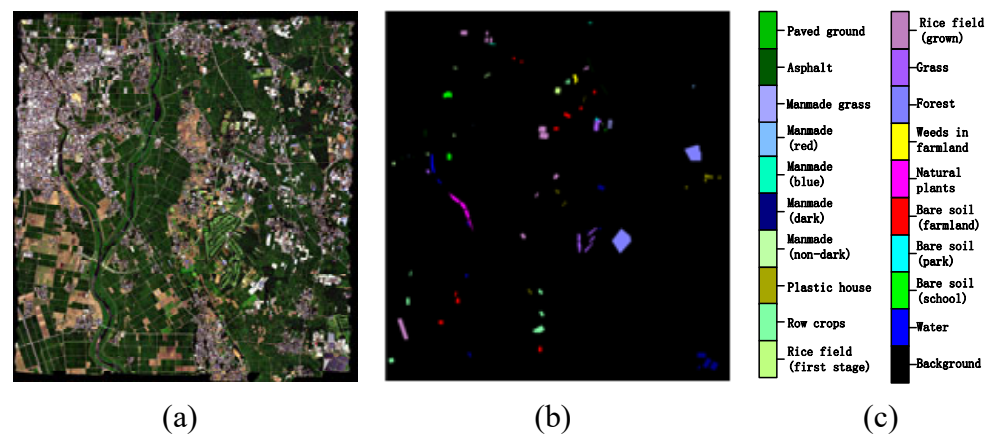
### 4.1. Datasets Description

In order to fairly evaluate the performance of the proposed method, we have selected datasets that are widely used in related works for training and testing. There are four datasets used, including Indian Pines (IP), Pavia University (PU), Salians (SA), and Chikusei. In a few-shot setting [25], Chikusei is selected as the source dataset, and the other three datasets are selected as the target datasets.

- (a) **Chikusei dataset:** Acquired on 29 July 2014, in Chikusei, Ibaraki, Japan, the dataset was gathered using Hyperspectral Visible/Near-Infrared Cameras (Hyperspec-VNIRC). It encompasses 19 distinct classes and spans  $2517 \times 2335$  pixels, maintaining a spatial resolution of 2.5 m per pixel. Comprising 128 spectral bands spanning from 363 to 1018 nm, this dataset provides comprehensive spectral information. The pseudo-color composite image and the ground-truth map of Chikusei are depicted in Figure 4. Table 1 shows the land cover classes and the corresponding numbers of samples in the Chikusei dataset.

**Table 1.** Land cover classes and numbers of samples in Chikusei.

Class	Name	Pixels	Class	Name	Pixels
1	Water	2345	11	Row crops	5961
2	Bare soil (school)	2859	12	Plastic house	2193
3	Bare soil (park)	236	13	Manmade (non-dark)	1220
4	Bare soil (farmland)	48,525	14	Manmade (dark)	7664
5	Natural plants	4297	15	Manmade (blue)	431
6	Weeds in farmland	1108	16	Manmade (red)	222
7	Forest	20,516	17	Manmade grass	1040
8	Grass	6515	18	Asphalt	801
9	Rice field (grown)	13,369	19	Paved ground	145
10	Rice field (first stage)	1268		Total: 77,592	

**Figure 4.** Pseudo-color composite image and ground-truth map of Chikusei. (a) False-color image. (b) Ground truth. (c) Labels illustration.

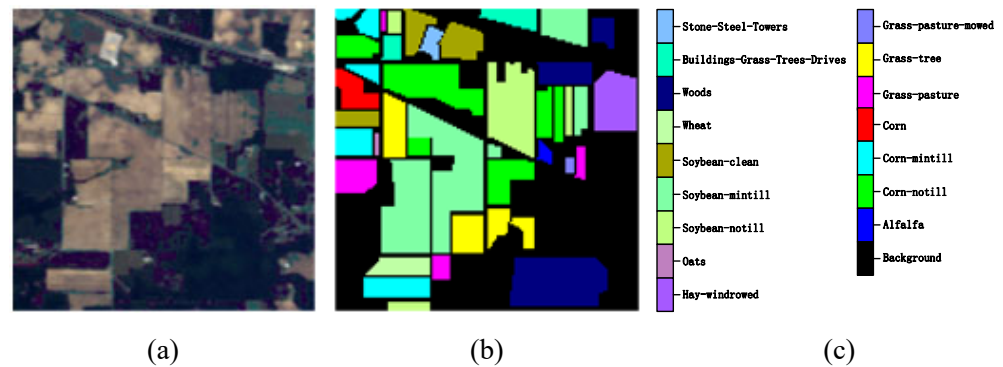
- (b) **IP dataset:** Acquired via the airborne visible infrared imaging spectrometer (AVIRIS) sensor at the Indiana Pine test site situated in northwestern Indiana, this dataset comprises  $145 \times 145$  pixels and initially included 224 spectral bands. However, 20 defective bands were removed, resulting in the utilization of the remaining 200 spectral bands for the experiment. With a spatial resolution of 20 m, this dataset encompasses 16 distinct classes. The pseudo-color composite image and the ground-truth map of IP are depicted in Figure 5. Table 2 shows the land cover classes and the corresponding numbers of samples in the IP dataset.

**Table 2.** Land cover classes and numbers of samples in IP.

Class	Name	Pixels	Class	Name	Pixels
1	Alfalfa	46	10	Soybean-notill	972
2	Corn-notill	1428	11	Soybean-mintill	2455
3	Corn-mintill	830	12	Soybean-clean	593
4	Corn	237	13	Wheat	205
5	Grass-pasture	483	14	Woods	1265
6	Grass-trees	730	15	Buildings-Grass-Trees-Drives	386
7	Grass-pasture-mowed	28	16	Stone-Steel-Towers	93
8	Hay-windrowed	478		Total: 10,249	
9	Oats	20			

- (c) **SA dataset:** Collected using a 224-band AVIRIS sensor over the scenic Salinas Valley in California, USA, the Salinas Valley dataset boasts high spatial resolution at 3.7 m

per pixel. Featuring an image encompassing 512 lines by 217 samples, this dataset underwent the removal of 20 abundant spectral bands, leaving behind 204 bands that effectively represent 16 diverse classes. The pseudo-color composite image and the ground-truth map of SA are depicted in Figure 6. Table 3 shows the land cover classes and the corresponding numbers of samples in the SA dataset.



**Figure 5.** Pseudo-color composite image and ground-truth map of IP. (a) False-color image. (b) Ground truth. (c) Labels illustration.

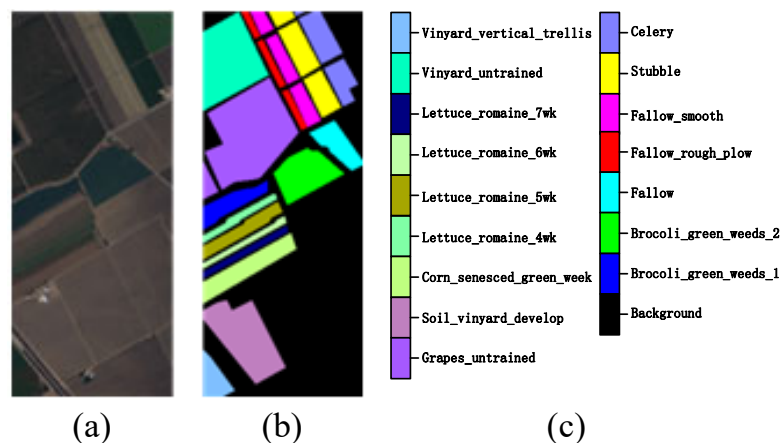
**Table 3.** Land cover classes and numbers of samples in SA.

Class	Name	Pixels	Class	Name	Pixels
1	Brocoli_green_weeds 1	2009	10	Corn_senesced_green_weeds	3278
2	Brocoli_green_weeds 2	3726	11	Lettuce_romaine_4wk	1068
3	Fallow	1976	12	Lettuce_romaine_5wk	1927
4	Fallow_rough_plow	1394	13	Lettuce_romaine_6wk	916
5	Fallow_smooth	2678	14	Lettuce_romaine_7wk	1070
6	Stubble	3959	15	Vinyard_untrained	7268
7	Celery	3579	16	Vinyard_vertical_trellis	1807
8	Grapes_untrained	11,271	Total: 54,129		
9	Soil_vinyard_develop	6203			

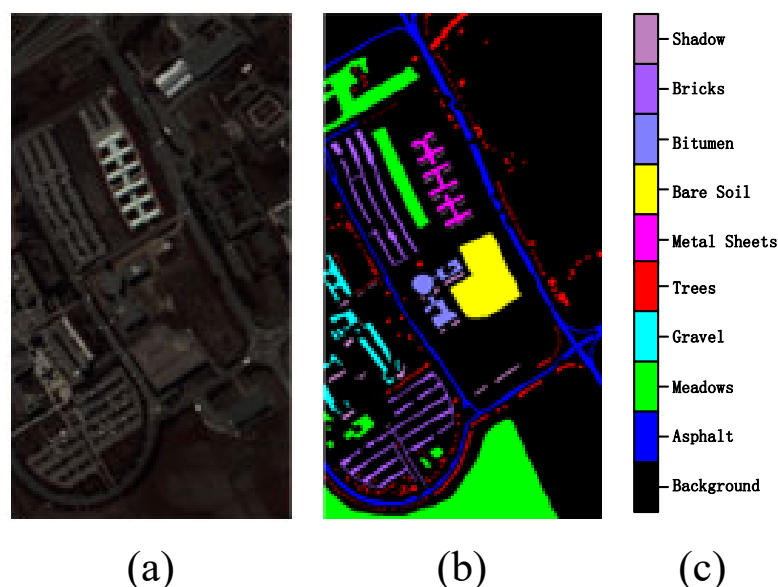
(d) **PU dataset:** Captured over the Pavia University, Italy, utilizing the reflective optics system imaging spectrometer (ROSIS-3), this image spans a size of  $610 \times 340$  pixels and maintains a spatial resolution of 1.3 m per pixel across 115 spectral bands. Following the removal of 12 noisy bands, subsequent experiments were carried out using the remaining 103 bands. The pseudo-color composite image and the ground-truth map of PU are depicted in Figure 7. Table 4 shows the land cover classes and the corresponding numbers of samples in the PU dataset.

**Table 4.** Land cover classes and numbers of samples in PU.

Class	Name	Pixels
1	Asphalt	6631
2	Meadows	18,649
3	Gravel	2099
4	Trees	3064
5	Metal sheets	1345
6	Bare soil	5029
7	Bitumen	1330
8	Bricks	3682
9	Shadows	947
Total		42,776



**Figure 6.** Pseudo-color composite image and ground-truth map of SA. (a) False-color image. (b) Ground truth. (c) Labels illustration.



**Figure 7.** Pseudo-color composite image and ground-truth map of PU. (a) False-color image. (b) Ground truth. (c) Labels illustration.

#### 4.2. Experiment Setting

Since the research in this paper addresses the few-shot HSI classification with noisy labels, two types of state-of-the-art methods are compared in this paper. First, two latest few-shot HSI classification methods, Gia-FSL [12] and RPCL [13], are selected for comparison to indicate the impact of the noisy labels on few-shot HSI classification and the outperformance of the proposed method under such impact. Gia-FSL [12] combined graph information aggregation-based FSL with domain alignment to address domain bias. RPCL [13] integrated supervised contrastive learning into few-shot HSI classification to obtain better prototypes. Second, two typical methods which can handle HSI classification with noisy labels, SSRN [59] and DCRN [15], are also selected to indicate the superiority of the proposed method to address noisy labels in the task of few-shot HSI classification. SSRN designed an end-to-end spectral–spatial residual network for performing hyperspectral image classification. DCRN [15] employed a two-channel residual network structure and an anti-noise loss function to enhance the robustness of the model to noisy labels.

In the comparison, 200 labeled samples per class from the source dataset were randomly selected for training. For the target datasets, only five labeled samples per class were selected for training. These were augmented to 200 samples per class by cropping

and restoration. And the rest of the samples in the target datasets were used for testing. In each episode of the training process, a  $N$ -way  $K$ -shot task was constructed to form the support set, where  $N$  is the number of classes in the dataset (e.g.,  $N = 16$  for IP and SA, and 9 for PU) and  $K$  is set to 5. At the same time, the number of samples per class composing the query set is set to 15, i.e.,  $M = 15$ .

For the ablation experiment's purpose, we name the method that removes the weighted contrastive regularization function from the proposed method as WCPN-CL, to validate the contribution of the weighted contrastive regularization function.

Since the datasets used for evaluation do not contain noise themselves, we need to set the noisy labels manually to validate the effectiveness of the proposed method. To simulate the real-world noise situation, we choose the paired label swap method [60] which swaps the labels of two samples selected randomly. This is in accordance with the premise that the number of classes in the datasets is determined. We selected the support set and query set from the dataset to which noise has been added, and the number of noise samples for each task has randomness due to random sampling, which is more in line with real noise-construction scenarios. For the proportion of samples over the datasets, we set it consistent with most of the methods [46,48], including 0% (i.e., no noise), 20%, 40%, and 60%, so as to better verify the model's robustness at different noise levels. Excessive noise such as noise rate greater than 80% is often discarded in real scenarios, so we do not take it into account.

In the meta-testing phase, the testing samples from the target datasets are fed into the trained mapping module and feature extractor to obtain discriminative features, and then a KNN classifier is used to classify the unlabeled samples. Overall accuracy (OA), average accuracy (AA), and kappa coefficient were used to evaluate the performance of different methods. We randomly ran each experiment 10 times and report the average results. All experiments were conducted on a Pytorch computer with a 56 MHz CPU, Tesla T4 GPU, and 64 GB of RAM. Adam was used as the optimizer and the number of training iterations was set to 3000. The learning rate was set to 0.001. The window size of the input patches was set to  $9 \times 9$ .

#### 4.3. Comparing with Other Methods

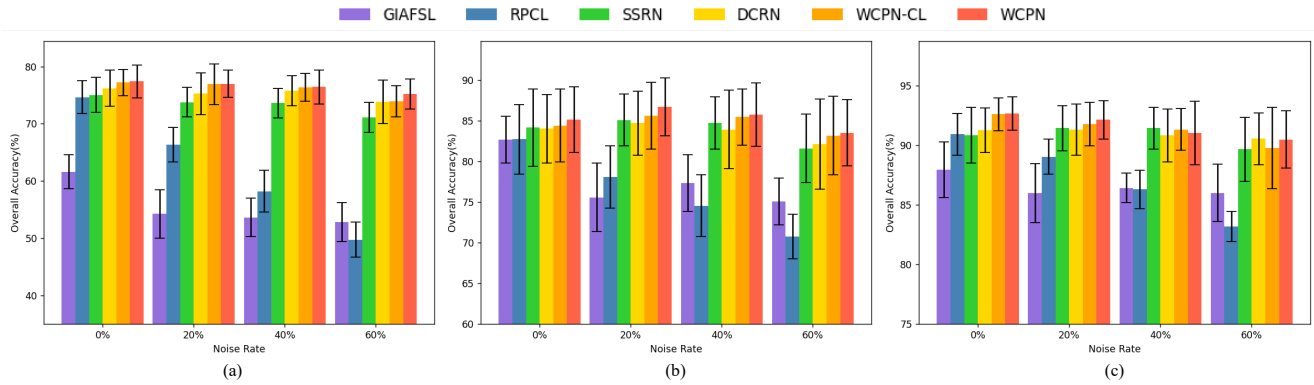
Tables 5 and 6 show the classification performance of the different methods on the IP dataset with different noise settings. For the few-shot methods for HSI classification, Gia-FSL and RPCL are very sensitive to noise. As the noise rate goes from 0% to 60%, the performance of Gia-FSL and RPCL decreases by 8.79% and 24.9%, respectively. At the same time, the performance of SSRN and DCRN are relatively stable with increasing noise rates. However, their accuracy rates are lower than WPCN, which shows the effect of the calibration brought by weighted samples. For WCPN and WCPN-CL, they perform relatively well at all noise rates, especially at noise levels of 20% and 40%. The WPCN exhibits better robustness, and the changes in noise do not have a significant effect on it. Figure 8a visualizes the classification effect of different methods on the IP dataset, and the advantages of the WCPN-CL and WCPN become more obvious as the noise rate increases.

Tables 7 and 8 show the classification performance of the different methods on the SA dataset. Compared with the performance on IP dataset, all methods are less affected by noise on this dataset. This may be because the HSI images of this dataset are easy to classify (the OA accuracy of most methods exceed 90%). WPCN only achieved an advantage with the noise rate of 0% and 20%, indicating that the role of sample weights for decreasing the impact of noisy labels is not significant for datasets that are easy to classify. It may be because a lot of useful information is filtered out when using the weights for calibration. Figure 8b visualizes the classification effectiveness of the different methods on the SA dataset, which also confirms the superiority of the proposed method.

Tables 9 and 10 show the classification performance of the different methods on the PU dataset. WPCN achieves optimal performance over different levels of noise rate, outperforming the mainstream method DCRN by about 1.5% on average. Its noise immunity



is also well demonstrated on the PU dataset. Although the increase in noise rate significantly affects the classification accuracy, the presence of weighted contrastive learning suppresses this effect to some extent due to the large differences between the noisy samples of different classes. Figure 8c visualizes the classification effect of different methods on the PU dataset. And the advantages of WCPN-CL and WCPN can be also found from the results in this figure.



**Figure 8.** OA results for GIAFSL, RPCL,SSRN, DCRN, WCPN-CL, and WCPN methods at different noise rates: (a) IP, (b) PU, and (c) SA.

**Table 5.** Classification performance of the GIAFSL, RPCL,SSRN, DCRN, WCPN-CL, and WCPN methods on the IP dataset with noise rate 0% and 20%.

Class	Noise Rate = 0%						Noise Rate = 20%					
	GIAFSL	RPCL	SSRN	DCRN	WCPN-CL	WCPN	GIAFSL	RPCL	SSRN	DCRN	WCPN-CL	WCPN
1	89.76	<b>99.76</b>	99.51	<b>99.76</b>	99.02	<b>99.76</b>	88.54	96.83	99.27	<b>99.76</b>	99.51	98.54
2	41.33	64.27	65.04	65.4	<b>70.71</b>	68.4	33.46	48.38	60.87	63.94	<b>69.69</b>	65.73
3	44.06	61.99	61.31	<b>67.85</b>	64.27	64.36	31.61	51.09	<b>67.41</b>	61.61	67.15	66.81
4	72.46	88.36	90.52	<b>92.84</b>	89.61	90.04	48.32	76.55	88.92	<b>92.33</b>	90.26	92.2
5	69.67	79.85	79.25	80.17	80.9	<b>80.94</b>	64.31	73.08	77.2	<b>79.81</b>	78.83	79.29
6	76.23	90.69	89.85	<b>92.79</b>	90.76	89.26	69.61	86.19	91.23	<b>91.77</b>	91.13	90.83
7	99.13	<b>100</b>	<b>100</b>	<b>100</b>	<b>100</b>	<b>100</b>	90.43	99.13	99.57	<b>100</b>	98.7	<b>100</b>
8	88.44	91.54	96.28	94.8	98.08	<b>98.84</b>	85.58	89.92	94.31	<b>94.33</b>	94.06	94.29
9	98	<b>100</b>	<b>100</b>	99.33	<b>100</b>	<b>100</b>	96.67	<b>100</b>	<b>100</b>	<b>100</b>	<b>100</b>	<b>100</b>
10	57.39	69.19	70.56	71.95	68	<b>72.09</b>	53.07	64.98	<b>71.34</b>	68.84	70.8	69.98
11	58.88	67.36	66.22	65.8	70.4	<b>70.42</b>	54.19	58.46	62.89	67.68	67.98	<b>72.7</b>
12	44	60.43	68.11	<b>68.57</b>	66.6	66.12	32.65	42.33	61.89	63.08	65.94	<b>67.23</b>
13	96.95	<b>98</b>	96.25	97.7	96.55	96.4	94.8	97.2	97.95	96.5	<b>98.6</b>	98
14	76.58	89.29	89.3	89.25	89.9	<b>91.8</b>	70.79	88.88	90	90.38	<b>91.48</b>	88.03
15	69.71	85.72	83.39	<b>90.52</b>	88.48	87.09	47.74	72.07	81.36	<b>88.11</b>	87.17	83.96
16	<b>99.2</b>	96.93	97.39	98.18	95	95.68	92.95	97.39	97.73	<b>98.3</b>	<b>98.3</b>	97.39
OA	61.62 ± 2.98	74.64 ± 2.88	75.05 ± 3.05	76.22 ± 3.15	77.22 ± 2.28	<b>77.4</b> ± 2.83	54.24 ± 4.26	66.37 ± 3.01	73.79 ± 2.59	75.26 ± 3.66	76.92 ± 3.53	<b>76.96</b> ± 2.38
AA	73.86 ± 1.68	83.96 ± 1.63	84.56 ± 1.43	<b>85.93</b> ± 1.7	85.52 ± 1.32	85.7 ± 1.55	65.92 ± 2.59	77.66 ± 1.72	83.87 ± 1.27	84.78 ± 2.0	<b>85.6</b> ± 1.47	85.31 ± 1.62
Kappa	0.56 ± 0.03	0.71 ± 0.03	0.71 ± 0.03	0.73 ± 0.03	<b>0.74</b> ± 0.03	<b>0.74</b> ± 0.03	0.48 ± 0.05	0.62 ± 0.03	0.70 ± 0.03	0.72 ± 0.04	<b>0.74</b> ± 0.04	<b>0.74</b> ± 0.03

**Table 6.** Classification performance of the GIAFSL, RPCL, SSRN, DCRN, WCPN-CL, and WCPN methods on the IP dataset with noise rate 40% and 60%.

Class	Noise Rate = 40%						Noise Rate = 60%					
	GIAFSL	RPCL	SSRN	DCRN	WCPN-CL	WCPN	GIAFSL	RPCL	SSRN	DCRN	WCPN-CL	WCPN
1	89.27	92.2	99.51	<b>99.76</b>	99.27	99.51	84.39	67.56	98.29	<b>99.76</b>	98.54	98.78
2	32.24	37.24	59.95	62.75	62.42	<b>65.43</b>	29.99	26.44	54.88	54.48	60.02	<b>62.45</b>
3	29.28	43.66	54.12	<b>65.14</b>	61.83	63.61	30.22	29.49	60.52	63.81	<b>64.72</b>	60.42
4	56.16	49.14	84.91	<b>91.25</b>	89.87	89.27	46.64	26.03	82.07	87.89	<b>88.58</b>	86.38
5	61.97	65.21	76.32	79.62	80.04	<b>80.13</b>	57.47	59.14	71.44	<b>80.52</b>	80.08	79.54
6	65.67	76.91	89.34	92.61	<b>93.56</b>	91.52	66.23	73.39	88.94	<b>92.87</b>	90.87	92.8
7	96.52	97.39	<b>100</b>	99.57	<b>100</b>	<b>100</b>	92.61	91.3	<b>100</b>	<b>100</b>	<b>100</b>	<b>100</b>
8	82.14	81.25	92.01	95.67	94.59	<b>95.98</b>	85.29	69.77	87.76	92.16	92.18	<b>94.84</b>
9	96	98	<b>100</b>	<b>100</b>	<b>100</b>	<b>100</b>	98	90.67	<b>100</b>	<b>100</b>	<b>100</b>	<b>100</b>
10	52.83	56.53	67.38	70.72	70.02	<b>71.81</b>	52.15	50.36	67.31	64.59	71.19	<b>72.7</b>
11	52.6	56.76	69.06	67.19	<b>71.28</b>	68.7	53.18	52.13	64.02	<b>68.36</b>	63.98	66.98
12	32.77	31.11	59.49	<b>66.43</b>	63.44	65.36	31.24	21.9	59.88	63.42	66.05	<b>66.77</b>
13	96.15	96.05	98.4	98.15	<b>98.8</b>	97.7	93.25	90.25	<b>97.85</b>	97.8	97	96.5
14	74.36	80.84	92.13	90.08	92.63	<b>92.75</b>	73.24	72.52	88.61	91.02	<b>91.06</b>	90.81
15	47.38	47.45	85.83	<b>89.5</b>	84.65	84.12	46.59	32.44	81.78	<b>86.06</b>	78.85	83.94
16	96.02	84.2	<b>98.98</b>	97.16	98.18	97.5	92.73	68.18	<b>98.64</b>	97.95	97.61	97.16
OA	53.61 ± 3.36	58.2 ± 3.67	73.64 ± 2.58	75.76 ± 2.64	76.35 ± 2.42	<b>76.45</b> ± 2.97	52.83 ± 3.37	49.74 ± 3.07	71.12 ± 2.66	73.83 ± 3.76	73.97 ± 2.74	<b>75.2</b> ± 2.64
AA	66.34 ± 3.53	68.37 ± 2.51	82.96 ± 1.32	<b>85.35</b> ± 1.74	85.04 ± 1.52	85.21 ± 1.92	64.58 ± 2.91	57.6 ± 3.56	81.37 ± 1.31	83.79 ± 2.2	83.8 ± 1.65	<b>84.38</b> ± 1.82
Kappa	0.48 ± 0.04	0.53 ± 0.04	0.70 ± 0.03	<b>0.73</b> ± 0.03	<b>0.73</b> ± 0.03	<b>0.73</b> ± 0.03	0.47 ± 0.04	0.43 ± 0.03	0.68 ± 0.03	0.71 ± 0.04	0.71 ± 0.03	<b>0.72</b> ± 0.03

**Table 7.** Classification performance of the GIAFSL, RPCL, SSRN, DCRN, WCPN-CL, and WCPN methods on the SA dataset with noise rates of 0% and 20%.

Class	Noise Rate = 0%						Noise Rate = 20%					
	GIAFSL	RPCL	SSRN	DCRN	WCPN-CL	WCPN	GIAFSL	RPCL	SSRN	DCRN	WCPN-CL	WCPN
1	98.62	99.39	99.24	97.78	99.47	<b>99.78</b>	96.35	98.29	<b>99.44</b>	99.39	98.36	98.7
2	99.42	99.94	99.38	99.17	<b>99.96</b>	99.87	97.77	99.55	99.44	99.82	<b>99.93</b>	99.56
3	90.7	90.59	90.55	91.73	94.16	<b>95.31</b>	87.97	91.22	92.2	<b>94.61</b>	94.04	90.63
4	98.81	99.18	99.03	98.81	98.83	<b>99.63</b>	99.16	99.48	99.58	<b>99.59</b>	99.52	99.55
5	89.62	90.96	91.83	92.6	93.12	<b>94.34</b>	92.54	91.71	94.44	95.35	94.55	<b>96.23</b>
6	<b>99</b>	98.81	98.24	98.84	98.77	98.75	98.88	98.99	98.91	<b>99.14</b>	98.92	98.51
7	98.59	<b>99.55</b>	96.79	98.98	<b>99.55</b>	99.43	98.55	99.23	97.31	98.14	99.17	<b>99.47</b>
8	74.85	80.37	81.43	80.45	83.69	<b>84.82</b>	69.85	72.95	78.68	76.66	<b>82.12</b>	81.03
9	97.43	99.73	99	99.15	99.52	<b>99.9</b>	97.88	99.84	99.18	<b>99.85</b>	99.41	99.73
10	79.93	86.66	86.03	87.71	<b>91.06</b>	90.93	72.17	85.46	89.18	90.93	88.22	<b>92.13</b>
11	96.62	98.89	98.54	98.67	99.14	<b>99.63</b>	95.96	98.28	99.54	99.18	99.44	<b>99.69</b>
12	<b>99.12</b>	98.82	98.89	98.46	96.47	98.22	97.81	99.61	<b>99.84</b>	99.37	98.68	98.76
13	98.16	<b>99.25</b>	99.07	99.08	99.17	96.98	99.42	99.2	99.24	<b>99.53</b>	99.17	99.34
14	98.14	97.91	98.51	<b>99.46</b>	97.2	99.03	98.23	98.57	98.44	<b>99.5</b>	95.97	97.43
15	73.18	79.23	78.53	81.48	<b>82.59</b>	79.57	71.91	76.79	<b>83.82</b>	82.42	79.66	82.15
16	89.33	91.59	95.77	93.61	<b>96.94</b>	96.47	84.42	92.32	92.9	95.06	<b>96.41</b>	96.34
OA	87.98 ± 2.34	90.93 ± 1.76	90.84 ± 2.34	91.28 ± 1.89	92.62 ± 1.37	<b>92.67</b> ± 1.4	86 ± 2.49	89.04 ± 1.47	91.45 ± 1.88	91.33 ± 2.16	91.79 ± 1.82	<b>92.15</b> ± 1.61
AA	92.59 ± 1.35	94.43 ± 1.2	94.43 ± 1.61	94.75 ± 1.28	95.6 ± 0.95	<b>95.79</b> ± 1.08	91.18 ± 1.37	93.84 ± 0.74	95.13 ± 1.13	95.54 ± 0.96	95.22 ± 1.15	<b>95.58</b> ± 1.18
Kappa	0.87 ± 0.03	0.90 ± 0.02	0.90 ± 0.03	0.90 ± 0.02	<b>0.92</b> ± 0.02	<b>0.92</b> ± 0.02	0.84 ± 0.03	0.88 ± 0.02	<b>0.91</b> ± 0.02	0.90 ± 0.02	<b>0.91</b> ± 0.02	<b>0.91</b> ± 0.02

**Table 8.** Classification performance of the GIAFSL, RPCL, SSRN, DCRN, WCPN-CL, and WCPN methods on the SA dataset with noise rates of 40% and 60%.

Class	Noise Rate = 40%						Noise Rate = 60%					
	GIAFSL	RPCL	SSRN	DCRN	WCPN-CL	WCPN	GIAFSL	RPCL	SSRN	DCRN	WCPN-CL	WCPN
1	98.17	98.31	98.84	<b>99.6</b>	99.34	98.84	95.78	93.41	98.32	99.44	98.82	<b>99.5</b>
2	96.17	99.04	99.2	<b>99.52</b>	99.11	98.4	97.82	97.02	99.14	<b>99.54</b>	99.47	99.2
3	87.84	90.59	<b>93.64</b>	92.57	90.67	93.27	85.78	87.52	87.5	<b>89.64</b>	88.24	89.1
4	99.6	99.34	99.68	99.53	99.52	<b>99.69</b>	99.42	99.01	<b>99.78</b>	99.62	99.59	99.6
5	91.93	91.28	94.61	<b>95.72</b>	94.07	92.96	91.37	90.62	93.49	<b>94.9</b>	93.92	94.52
6	99.27	98.81	99.46	<b>99.59</b>	99.31	99.3	99.14	98.13	98.98	<b>99.75</b>	99.42	99.29
7	99.06	98.46	<b>99.14</b>	98.36	97.64	93.88	99.5	98.24	99.49	98.9	<b>99.64</b>	97.42
8	68.73	68.29	78.36	74.62	80.71	<b>80.76</b>	70.88	65.68	73.18	74.84	75.13	<b>76.1</b>
9	97.72	99.71	99.53	<b>99.78</b>	99.39	99.56	98.17	97.34	<b>99.66</b>	99.5	99.08	99.53
10	75.35	77.58	89	<b>91.02</b>	89.47	90.97	78.17	69.11	85.62	<b>89.74</b>	85.41	87.54
11	97.59	97.37	<b>99.53</b>	99.46	99.06	99.01	96.58	92.8	98.65	98.76	99.08	<b>99.29</b>
12	99.59	99.31	99.67	99.43	<b>99.84</b>	99.33	99	95.85	99.01	99.32	98.85	<b>99.83</b>
13	99.31	98.91	<b>99.7</b>	99.6	99.59	98.99	99.09	98.88	99.42	<b>99.86</b>	99.45	99.77
14	97.62	97.19	98.85	<b>99.59</b>	99.34	98.78	97.42	96.05	98.78	<b>99.3</b>	98.89	98.19
15	74.12	69.57	81.9	<b>81.92</b>	79.01	78.14	67.21	63.35	81.09	81.59	79.35	<b>81.62</b>
16	87.65	89.71	95.99	95.97	96.2	<b>96.8</b>	86.25	84.61	<b>94.66</b>	94	91.63	93.75
OA	86.42 ± 1.24	86.31 ± 1.61	<b>91.45</b> ± 1.77	90.85 ± 2.23	91.34 ± 1.75	91.05 ± 2.66	86 ± 2.41	83.2 ± 1.25	89.67 ± 2.69	<b>90.56</b> ± 2.17	89.8 ± 3.41	90.5 ± 2.4
AA	91.86 ± 1.35	92.09 ± 0.89	<b>95.45</b> ± 1.09	95.39 ± 1.24	95.14 ± 0.92	94.92 ± 1.82	91.35 ± 1.66	89.23 ± 1.11	94.17 ± 2.1	<b>94.92</b> ± 1.32	94.12 ± 2.55	94.64 ± 1.71
Kappa	0.85 ± 0.01	0.85 ± 0.02	<b>0.91</b> ± 0.02	0.90 ± 0.02	0.90 ± 0.02	0.90 ± 0.03	0.84 ± 0.03	0.81 ± 0.01	0.89 ± 0.03	<b>0.90</b> ± 0.02	0.89 ± 0.04	0.89 ± 0.03

**Table 9.** Classification performance of the GIAFSL, RPCL, SSRN, DCRN, WCPN-CL, and WCPN methods on the PU dataset with noise rates of 0% and 20%.

Class	Noise Rate = 0%						Noise Rate = 20%					
	GIAFSL	RPCL	SSRN	DCRN	WCPN-CL	WCPN	GIAFSL	RPCL	SSRN	DCRN	WCPN-CL	WCPN
1	79.88	85.4	<b>91.51</b>	89.92	88.59	86.82	73.06	80.5	87.77	87.93	88.94	<b>89.81</b>
2	<b>87.7</b>	79.99	79.69	78.35	81.44	82.57	80.18	77.57	83.57	80.54	83.49	<b>86.39</b>
3	56.24	65.54	76.4	76.1	79.37	<b>80.6</b>	55.98	67.09	78.51	77.13	<b>82.35</b>	77.79
4	<b>93.11</b>	91.89	92	92.27	86.95	85.72	89.6	<b>91.45</b>	89.64	90.76	87.21	88.17
5	98.82	<b>99.39</b>	98.58	98.69	99.19	99.09	94.17	98.44	99.71	99.19	99.6	<b>99.72</b>
6	74.88	82.68	75.26	<b>84.07</b>	82.13	81.92	62.21	70.7	77.91	<b>83.84</b>	79.47	79.64
7	75.46	84.29	92.83	89.15	92.12	<b>93.18</b>	74.26	79.69	94.31	<b>94.62</b>	93	91.09
8	72.3	83.47	<b>92.08</b>	89.68	84.55	90.84	63.01	68.58	86.28	<b>88.82</b>	88.43	87.84
9	97.28	<b>97.75</b>	96.85	95.68	96.66	96.5	96.6	98.12	96.13	95.99	<b>98.15</b>	96.82
OA	82.68 ± 2.88	82.72 ± 4.26	84.17 ± 4.75	84.03 ± 4.23	84.42 ± 4.5	<b>85.15</b> ± 4.07	75.59 ± 4.24	78.1 ± 3.85	85.09 ± 3.19	84.71 ± 3.95	85.62 ± 4.11	<b>86.75</b> ± 3.56
AA	81.74 ± 1.21	85.6 ± 3.11	88.36 ± 2.2	88.21 ± 2.63	87.89 ± 2.88	<b>88.58</b> ± 2.42	76.56 ± 1.74	81.35 ± 2.35	88.2 ± 2.84	88.76 ± 2.59	<b>88.96</b> ± 2.57	88.58 ± 3.12
Kappa	0.77 ± 0.03	0.78 ± 0.05	0.80 ± 0.05	0.80 ± 0.05	0.80 ± 0.05	<b>0.81</b> ± 0.05	0.69 ± 0.05	0.72 ± 0.04	0.81 ± 0.04	0.80 ± 0.05	0.81 ± 0.05	<b>0.83</b> ± 0.04

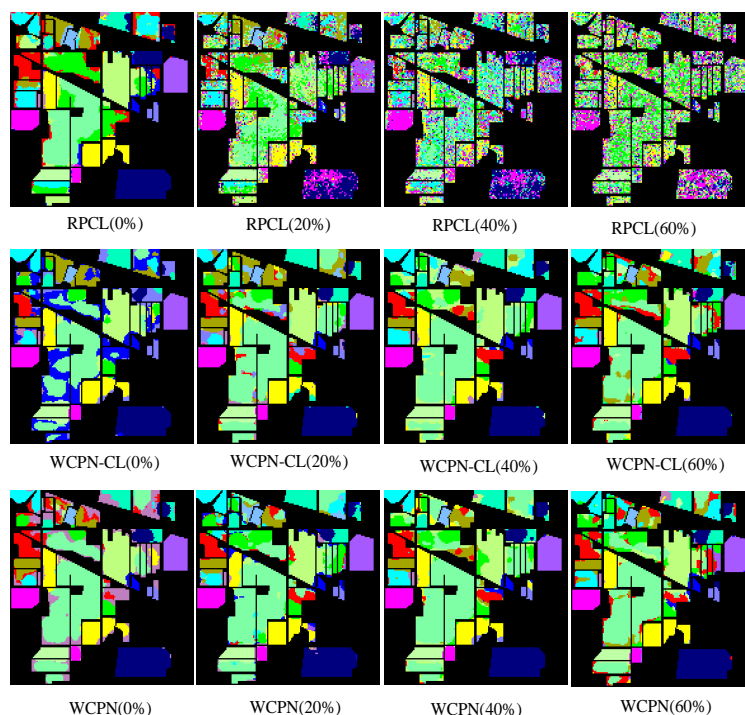
**Table 10.** Classification performance of the GIAFSL, RPCL, SSRN, DCRN, WCPN-CL, and WCPN methods on the PU dataset with noise rates of 40% and 60%.

Class	Noise Rate = 40%						Noise Rate = 60%					
	GIAFSL	RPCL	SSRN	DCRN	WCPN-CL	WCPN	GIAFSL	RPCL	SSRN	DCRN	WCPN-CL	WCPN
1	71.69	75.13	89.23	87.88	<b>90.12</b>	89.66	70.57	68.25	86.31	86.99	<b>87.96</b>	86.45
2	<b>84.64</b>	76.15	81.7	79.39	83.32	82.93	<b>82.04</b>	77.54	77.95	79.36	79.47	80.5
3	60.3	56.93	74.22	73.23	77.75	<b>78.25</b>	52.94	55.66	75.45	68.22	<b>79.38</b>	74.28
4	89.24	89.9	<b>92.14</b>	91.59	91.38	89.67	89.61	73.44	90.67	<b>92.68</b>	90.29	90.05
5	95.53	95.49	99.49	98.97	99.18	<b>99.81</b>	95.77	88.11	<b>99.73</b>	97.68	99.59	99.45
6	59.53	64.48	79.77	<b>85.52</b>	77.8	81.35	54.43	60.23	75.46	77.52	78.58	<b>81.66</b>
7	81.15	79.9	95.08	91.68	<b>95.74</b>	93.41	79.83	78.74	92.32	90.69	<b>92.95</b>	92.58
8	61.58	61.17	86.38	86.5	86.24	<b>89.43</b>	61.82	46.6	81.64	81.14	82.71	<b>83.05</b>
9	96.97	96.86	97.1	94.7	97.81	<b>97.87</b>	96.68	93.97	97.41	94.43	98.05	<b>98.81</b>
OA	77.34 ± 3.48	74.55 ± 3.79	84.74 ± 3.23	83.94 ± 4.83	85.48 ± 3.47	<b>85.78</b> ± 3.89	75.08 ± 2.88	70.77 ± 2.72	81.62 ± 4.21	82.14 ± 5.52	83.19 ± 4.81	<b>83.53</b> ± 4.09
AA	77.85 ± 1.61	77.34 ± 1.31	88.35 ± 1.75	87.72 ± 2.14	88.81 ± 1.82	<b>89.15</b> ± 1.83	75.97 ± 1.54	71.39 ± 2.42	86.33 ± 2.74	85.41 ± 1.99	<b>87.67</b> ± 2.08	87.43 ± 1.77
Kappa	0.70 ± 0.04	0.67 ± 0.04	0.80 ± 0.04	0.80 ± 0.05	0.81 ± 0.04	<b>0.82</b> ± 0.05	0.68 ± 0.03	0.62 ± 0.03	0.77 ± 0.05	0.77 ± 0.06	<b>0.79</b> ± 0.05	<b>0.79</b> ± 0.05

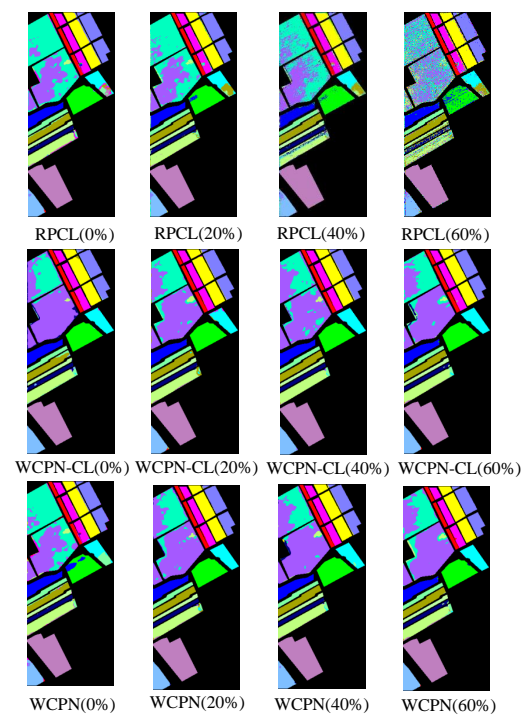
#### 4.4. Classification Visualization

In order to further demonstrate the noise immunity and the accuracy of the proposed method in the presence of increasing noise, we show the classification maps and 2D feature visualization of the different methods on the IP, SA, and PU datasets. However, due to space constraints, we only show RPCL [13] which is the latest few-shot method for HSI classification, and our WCPN-CL and WCPN methods. Figures 9–11 are the classification maps, and Figures 12–14 are the 2D feature visualizations. The results in these figures also confirm the advantages of the proposed method.

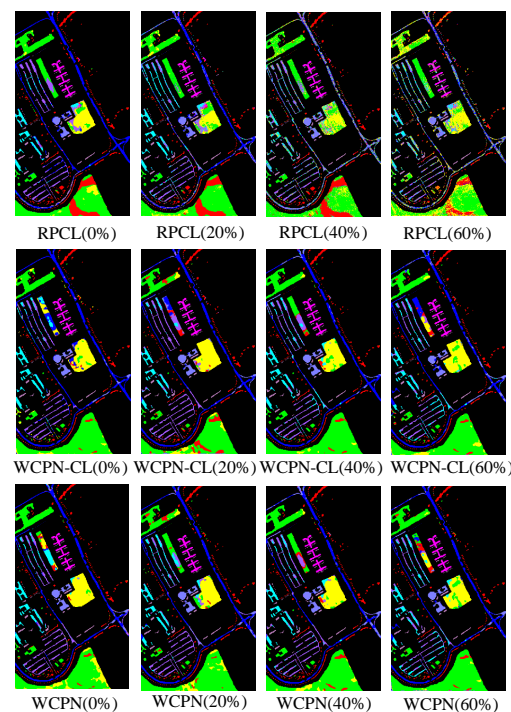
Figure 9 shows the classification map on the IP dataset. For the different methods in the presence of a gradually increasing noise rate, the degree of confusion of the classification also gradually increases. However, it can be clearly seen that RPCL is much more disorganized in the presence of noise than our WCPN-CL and WCPN methods. The WCPN achieves a good noise-resistant classification for both class 7 (Grass-pasture-mowed) on the left and class 8 (Hay-windrowed) on the right. Figure 12 shows the 2D feature visualization on the IP dataset. The clustering of each class keeps getting worse with the increase in noise, and our WCPN method shows significant advantages when noise rate is greater than 40%. Figure 10 shows the classification map on the SA dataset, where the WCPN achieves a better classification for the two classes located at the top left (Broccoli-green-weeds 2), even similar to the noiseless case. Figure 13 shows the 2D feature visualization on the SA dataset; the clustering of each class does not differ much when the noise rate increases, but there is an indistinguishable case in RPCL in orange color (class 8) and blue color (class 15), which coincides with the results in Tables 7 and 8. Figure 11 shows the classification map on the PU dataset, where the WCPN achieves a better classification for class 8 (Bricks) located in the center. Figure 14 shows the 2D feature visualization on PU dataset, where the class 8 in orange color also gets better 2D clustering.



**Figure 9.** Classification map on the IP dataset. RPCL (0%), 74.64%; RPCL (20%), 66.37%; RPCL (40%), 58.2%; RPCL (60%), 49.74%; WCPN-CL (0%), 77.22%; WCPN-CL (20%), 76.92%; WCPN-CL (40%), 76.35%; WCPN-CL (60%), 73.97%; WCPN (0%), 77.4%; WCPN (20%), 76.96%; WCPN (40%), 76.45%; WCPN (60%), 75.2%.



**Figure 10.** Classification map on the SA dataset. RPCL (0%), 90.93%; RPCL (20%), 89.04%; RPCL (40%), 86.31%; RPCL (60%), 83.2%; WCPN-CL (0%), 92.62%; WCPN-CL (20%), 91.79%; WCPN-CL (40%), 91.34%; WCPN-CL (60%), 89.8%; WCPN (0%), 92.67%; WCPN (20%), 92.39%; WCPN (40%), 91.39%; WCPN (60%), 90.5%.



**Figure 11.** Classification map on the PU dataset. RPCL (0%), 82.72%; RPCL (20%), 78.1%; RPCL (40%), 74.55%; RPCL (60%), 70.77%; WCPN-CL (0%), 84.42%; WCPN-CL (20%), 85.62%; WCPN-CL (40%), 85.48%; WCPN-CL (60%), 83.19%; WCPN (0%), 85.15%; WCPN (20%), 86.75%; WCPN (40%), 85.78%; WCPN (60%), 83.53%.



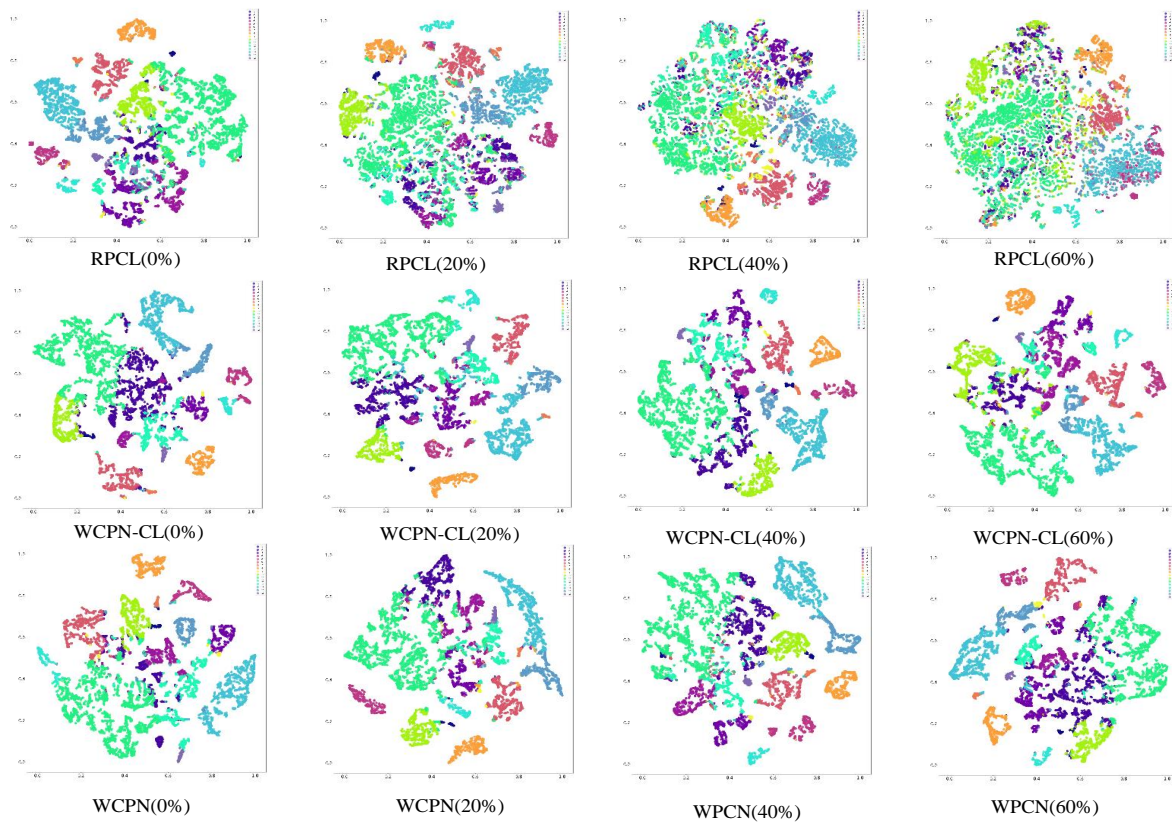


Figure 12. 2-D feature visualization on IP.

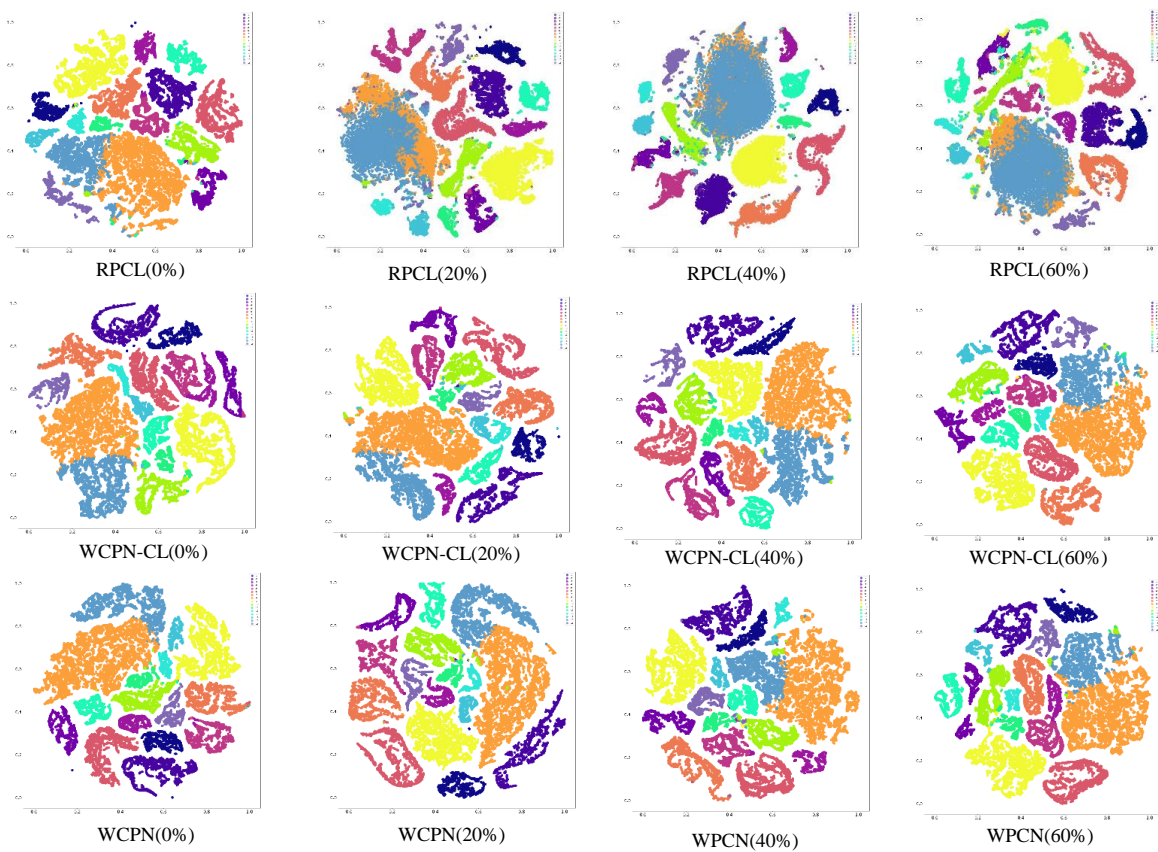
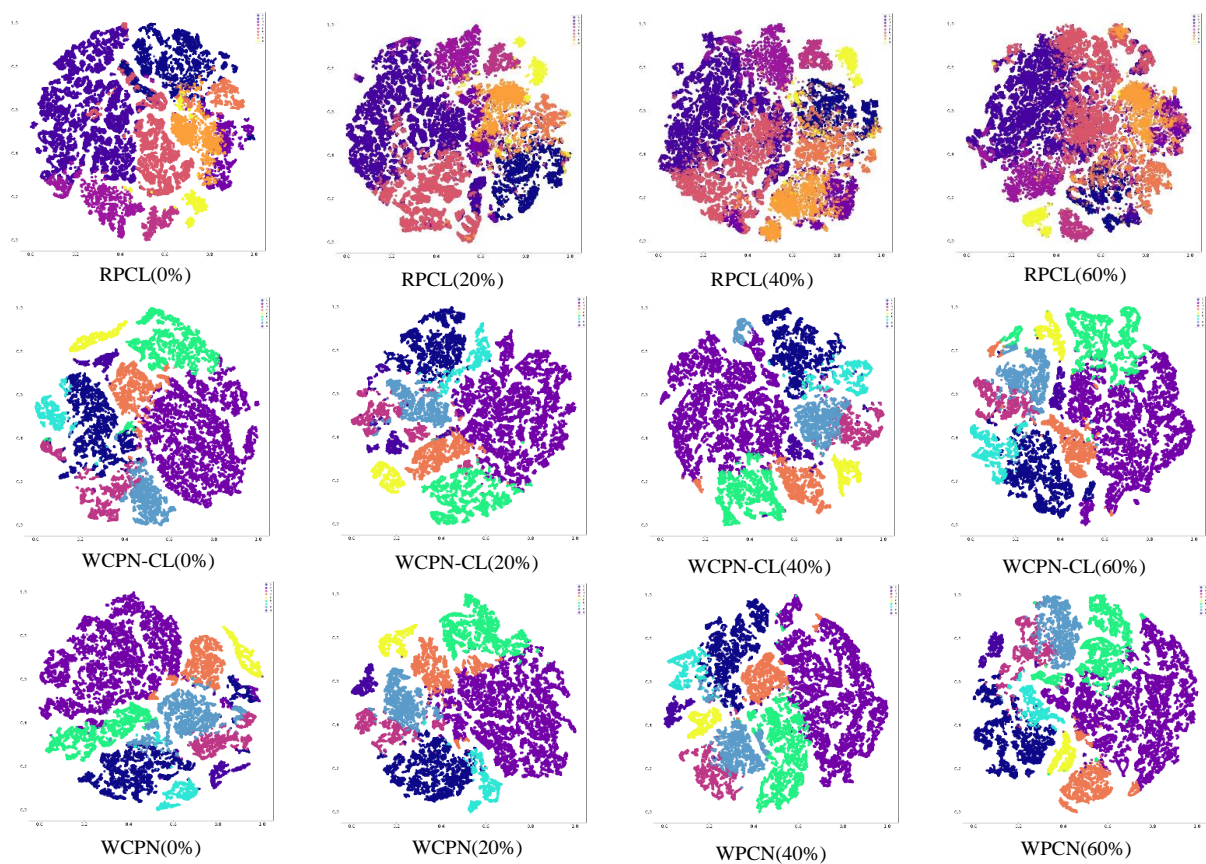


Figure 13. 2-D feature visualization on SA.

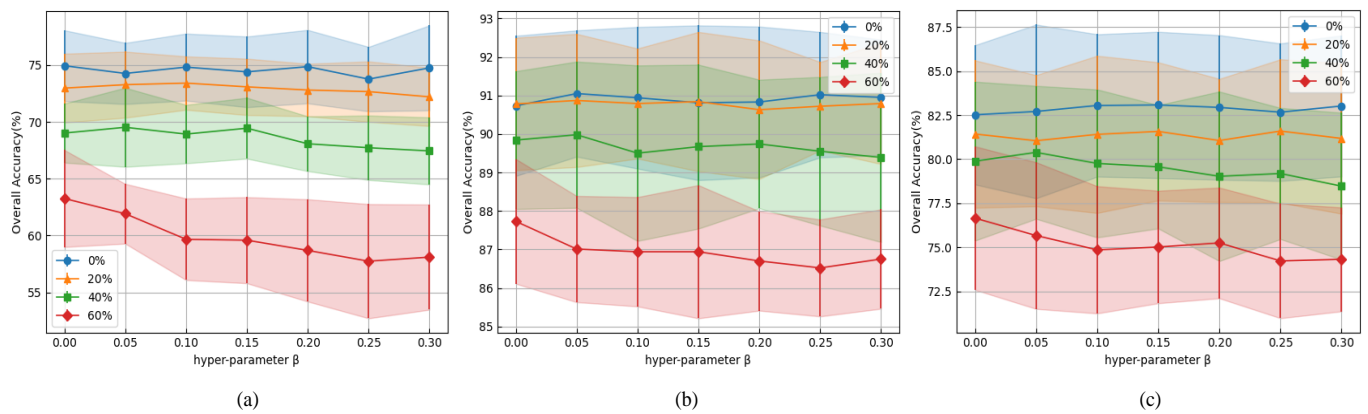


**Figure 14.** 2-D feature visualization on PU.

#### 4.5. Sensitive Analysis of Parameters

As shown in Equation (11) above, there is a hyperparameter  $\beta$  in the weighted contrastive regularization function, which indicates how tolerant the regularization function is to the noise weights below the mean. In our experiments, we set  $\beta \in [0, 0.3]$ , and changed it with a step size of 0.05 to see how sensitive the proposed method is to the change of  $\beta$ .

Figure 15 shows the classification results of the WCPN when changing the value of  $\beta$  under different noise rates. From the sub-figures (a–c), it can be seen that the sensitivity of the proposed method to the hyperparameter increases when the noise rate is greater than 40% on different datasets. When the noise rate is less than 20%, the change in the hyperparameter  $\beta$  does not have a significant impact on the proposed method. The reasons are that when  $\beta = 0$ , the criteria for determining whether it is a clean or a noisy positive sample are strict, and when  $\beta$  is larger, the criteria are more lenient. When the noise rate is large, there are many noisy samples, and a smaller  $\beta$  will filter more noisy samples and improve the credibility of positive samples. There will be better classification performance. Meanwhile, when the noise rate is small, there are fewer noisy samples, and either harsh or lenient criteria will have less impact on the classification performance.



**Figure 15.** WCPN classification results for  $\beta$  at different noise rates: (a) IP, (b) PU, and (c) SA.

## 5. Discussion

In this study, we propose WCPN to address the challenges posed by noisy labels in few-shot hyperspectral image classification. While our method demonstrates promising performance in mitigating the impact of noisy labels, several points warrant further discussion.

Firstly, in our experiments, noise was introduced by swapping the labels of part of the data in one dataset with those of another part. This method simulates label errors but does not necessarily reflect the arbitrary nature of real-world label noise, which might not be readily identifiable. Future work could explore more realistic noise models to better simulate the variability and complexity of label errors encountered in practical scenarios. Secondly, we observed that WCPN shows higher noise resistance in datasets that are relatively easy to classify. However, its effectiveness is less pronounced in more challenging datasets. This suggests that our method may need further refinement to handle complex data more robustly. Finally, incorporating contrastive learning into the WCPN results in increased training time. While contrastive learning enhances feature representation, its computational cost is non-trivial. Future research should address how to optimize the training process to reduce computational expenses while maintaining or improving the method's efficacy.

## 6. Conclusions

In this paper, we propose a WCPN to address the challenges posed by noisy labels in few-shot HSI classification. By utilizing the similarity between the samples from same classes to obtain the sample weights and applying it to the prototype calculation of support and query sets, the WCPN demonstrates some self-calibration ability in the presence of noisy samples. In addition, we introduce a weighted contrastive regularization function to enhance the degree of data aggregation and improve the differentiation of prototypes. Experiments on artificially generated noisy labels on several HSI datasets are conducted, and the results show that the WCPN has excellent performance in mitigating the impact of noisy labels. This shows that for HSI classification, the proposed method can better adapt to noisy environments where there are small amounts of labeled data available. But it is worth noting that the proposed method will cost more time because of the computation of the weights of the samples.

**Author Contributions:** Methodology, C.L.; Validation, C.L. and Z.H.; Formal analysis, D.Z., Y.R. and C.L.; Investigation, D.Z. and Z.H.; Resources, Z.H.; Data curation, Y.R. and Z.H.; Writing—original draft, Y.R.; Writing—review & editing, C.L. and J.W.; Visualization, Y.R.; Supervision, D.Z., C.L. and J.W.; Project administration, D.Z.; Funding acquisition, D.Z. All authors have read and agreed to the published version of the manuscript.



**Funding:** This work was funded by National Science and Technology Major Project for High Resolution Earth Observation System grant number 80-Y50G19-9001-22/23, and Henan Province Key Research and Development Special Project grant number 241111210300.

**Data Availability Statement:** All the codes are available at github <https://github.com/12ian/WCPN-FSL>. The original contributions presented in the study are included in the article, further inquiries can be directed to the corresponding author.

**Conflicts of Interest:** The authors declare no conflicts of interest.

## References

1. Boggs, J.L.; Tsegaye, T.; Coleman, T.; Reddy, K.; Fahsi, A. Relationship between hyperspectral reflectance, soil nitrate-nitrogen, cotton leaf chlorophyll, and cotton yield: A step toward precision agriculture. *J. Sustain. Agric.* **2003**, *22*, 5–16. [[CrossRef](#)]
2. Lee, M.A.; Huang, Y.; Yao, H.; Thomson, S.J.; Bruce, L.M. Determining the effects of storage on cotton and soybean leaf samples for hyperspectral analysis. *IEEE J. Sel. Top. Appl. Earth Observ. Remote Sens.* **2014**, *7*, 2562–2570. [[CrossRef](#)]
3. Pontius, J.; Martin, M.; Plourde, L.; Hallett, R. Ash decline assessment in emerald ash borer-infested regions: A test of tree-level, hyperspectral technologies. *Remote Sens. Environ.* **2008**, *112*, 2665–2676. [[CrossRef](#)]
4. Dalponte, M.; Ørka, H.O.; Gobakken, T.; Gianelle, D.; Næsset, E. Tree species classification in boreal forests with hyperspectral data. *IEEE Trans. Geosci. Remote Sens.* **2012**, *51*, 2632–2645. [[CrossRef](#)]
5. Chi, J.; Crawford, M.M. Spectral unmixing-based crop residue estimation using hyperspectral remote sensing data: A case study at Purdue university. *IEEE J. Sel. Top. Appl. Earth Observ. Remote Sens.* **2014**, *7*, 2531–2539. [[CrossRef](#)]
6. Yuan, Y.; Wang, Q.; Zhu, G. Fast hyperspectral anomaly detection via high-order 2-D crossing filter. *IEEE Trans. Geosci. Remote Sens.* **2014**, *53*, 620–630. [[CrossRef](#)]
7. Paoletti, M.; Haut, J.; Plaza, J.; Plaza, A. Deep learning classifiers for hyperspectral imaging: A review. *ISPRS J. Photogramm. Remote Sens.* **2019**, *158*, 279–317. [[CrossRef](#)]
8. Kang, J.; Zhang, Y.; Liu, X.; Cheng, Z. Hyperspectral Image Classification Using Spectral–Spatial Double-Branch Attention Mechanism. *Remote Sens.* **2024**, *16*, 193. [[CrossRef](#)]
9. Snell, J.; Swersky, K.; Zemel, R. Prototypical networks for few-shot learning. In Proceedings of the Advances in Neural Information Processing Systems, Long Beach, CA, USA, 4–9 December 2017.
10. Sung, F.; Yang, Y.; Zhang, L.; Xiang, T.; Torr, P.H.; Hospedales, T.M. Learning to compare: Relation network for few-shot learning. In Proceedings of the IEEE Conference On Computer Vision And Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018; pp. 1199–1208.
11. Li, Z.; Liu, M.; Chen, Y.; Xu, Y.; Du, Q. Deep Cross-Domain Few-Shot Learning for Hyperspectral Image Classification. *IEEE Trans. Geosci. Remote Sens.* **2021**, *60*, 5501618. [[CrossRef](#)]
12. Zhang, Y.; Li, W.; Zhang, M.; Wang, S.; Tao, R.; Du, Q. Graph information aggregation cross-domain few-shot learning for hyperspectral image classification. *IEEE Trans. Neural Netw. Learn. Syst.* **2022**, *35*, 1912–1925. [[CrossRef](#)]
13. Liu, Q.; Peng, J.; Ning, Y.; Chen, N.; Sun, W.; Du, Q.; Zhou, Y. Refined Prototypical Contrastive Learning for Few-Shot Hyperspectral Image Classification. *IEEE Trans. Geosci. Remote Sens.* **2023**, *61*, 5506214. [[CrossRef](#)]
14. Tu, B.; Zhou, C.; He, D.; Huang, S.; Plaza, A. Hyperspectral classification with noisy label detection via superpixel-to-pixel weighting distance. *IEEE Trans. Geosci. Remote Sens.* **2020**, *58*, 4116–4131. [[CrossRef](#)]
15. Xu, Y.; Li, Z.; Li, W.; Du, Q.; Liu, C.; Fang, Z.; Zhai, L. Dual-channel residual network for hyperspectral image classification with noisy labels. *IEEE Trans. Geosci. Remote Sens.* **2021**, *60*, 5502511. [[CrossRef](#)]
16. Lee, K.H.; He, X.; Zhang, L.; Yang, L. Cleannet: Transfer learning for scalable image classifier training with label noise. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018; pp. 5447–5456.
17. Zhang, W.; Wang, Y.; Qiao, Y. MetaCleaner: Learning to Hallucinate Clean Representations for Noisy-Labeled Visual Recognition. In Proceedings of the 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Long Beach, CA, USA, 15–19 June 2019.
18. Han, J.; Luo, P.; Wang, X. Deep self-learning from noisy labels. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Long Beach, CA, USA, 15–19 June 2019; pp. 5138–5147.
19. Ma, X.; Huang, H.; Wang, Y.; Romano, S.; Erfani, S.; Bailey, J. Normalized loss functions for deep learning with noisy labels. In Proceedings of the International Conference on Machine Learning, Virtual, 13–18 July 2020; pp. 6543–6553.
20. Tu, B.; Zhang, X.; Kang, X.; Zhang, G.; Li, S. Density peak-based noisy label detection for hyperspectral image classification. *IEEE Trans. Geosci. Remote Sens.* **2018**, *57*, 1573–1584. [[CrossRef](#)]
21. Bateni, P.; Goyal, R.; Masrani, V.; Wood, F.; Sigal, L. Improved few-shot visual classification. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, 13–19 June 2020; pp. 14493–14502.
22. Bendre, N.; Marín, H.T.; Najafirad, P. Learning from few samples: A survey. *arXiv* **2020**, arXiv:2007.15484.
23. Koch, G.; Zemel, R.; Salakhutdinov, R. Siamese neural networks for one-shot image recognition. In Proceedings of the ICML Deep Learning Workshop, Lille, France, 6–11 July 2015; Volume 2.

24. Rao, M.; Tang, P.; Zhang, Z. A Developed Siamese CNN with 3D Adaptive Spatial-Spectral Pyramid Pooling for Hyperspectral Image Classification. *Remote Sens.* **2020**, *12*, 1964 [CrossRef]
25. Liu, B.; Yu, X.; Yu, A.; Zhang, P.; Wan, G.; Wang, R. Deep few-shot learning for hyperspectral image classification. *IEEE Trans. Geosci. Remote Sens.* **2018**, *57*, 2290–2304. [CrossRef]
26. Tang, H.; Li, Y.; Han, X.; Huang, Q.; Xie, W. A spatial–spectral prototypical network for hyperspectral remote sensing image. *IEEE Geosci. Remote Sens. Lett.* **2019**, *17*, 167–171. [CrossRef]
27. Sun, J.; Shen, X.; Sun, Q. Hyperspectral Image Few-Shot Classification Network Based on the Earth Mover’s Distance. *IEEE Trans. Geosci. Remote Sens.* **2022**, *60*, 1–14. [CrossRef]
28. Gao, K.; Liu, B.; Yu, X.; Qin, J.; Zhang, P.; Tan, X. Deep relation network for hyperspectral image few-shot classification. *Remote Sens.* **2020**, *12*, 923. [CrossRef]
29. Ma, X.; Ji, S.; Wang, J.; Geng, J.; Wang, H. Hyperspectral image classification based on two-phase relation learning network. *IEEE Trans. Geosci. Remote Sens.* **2019**, *57*, 10398–10409. [CrossRef]
30. Bai, J.; Huang, S.; Xiao, Z.; Li, X.; Zhu, Y.; Regan, A.C.; Jiao, L. Few-shot hyperspectral image classification based on adaptive subspaces and feature transformation. *IEEE Trans. Geosci. Remote Sens.* **2022**, *60*, 1–17. [CrossRef]
31. Xi, B.; Li, J.; Li, Y.; Song, R.; Hong, D.; Chanussot, J. Few-shot learning with class-covariance metric for hyperspectral image classification. *IEEE Trans. Image Process.* **2022**, *31*, 5079–5092. [CrossRef] [PubMed]
32. Zhang, C.; Yue, J.; Qin, Q. Global prototypical network for few-shot hyperspectral image classification. *IEEE J. Sel. Top. Appl. Earth Observ. Remote Sens.* **2020**, *13*, 4748–4759. [CrossRef]
33. Liu, C.; Yang, L.; Li, Z.; Yang, W.; Han, Z.; Guo, J.; Yu, J. Multi-level relation learning for cross-domain few-shot hyperspectral image classification. *Appl. Intell.* **2024**, *54*, 4392–4410. [CrossRef]
34. Cheng, H.; Zhu, Z.; Li, X.; Gong, Y.; Sun, X.; Liu, Y. Learning with instance-dependent label noise: A sample sieve approach. *arXiv* **2020**, arXiv:2010.02347.
35. Wei, T.; Shi, J.X.; Tu, W.W.; Li, Y.F. Robust long-tailed learning under label noise. *arXiv* **2021**, arXiv:2108.11569.
36. Cheng, D.; Liu, T.; Ning, Y.; Wang, N.; Han, B.; Niu, G.; Gao, X.; Sugiyama, M. Instance-dependent label-noise learning with manifold-regularized transition matrix estimation. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, New Orleans, LA, USA, 18–24 June 2022; pp. 16630–16639.
37. Xia, X.; Liu, T.; Han, B.; Wang, N.; Gong, M.; Liu, H.; Niu, G.; Tao, D.; Sugiyama, M. Part-dependent label noise: Towards instance-dependent label noise. *Adv. Neural Inf. Process. Syst.* **2020**, *33*, 7597–7610.
38. Chen, L.H.; Li, H.; Zhang, W.; Huang, J.; Ma, X.; Cui, J.; Li, N.; Yoo, J. Anomman: Detect anomaly on multi-view attributed networks. *arXiv* **2022**, arXiv:2201.02822. [CrossRef]
39. Li, S.; Ge, S.; Hua, Y.; Zhang, C.; Wen, H.; Liu, T.; Wang, W. Coupled-view deep classifier learning from multiple noisy annotators. In Proceedings of the AAAI Conference on Artificial Intelligence, New York, NY, USA, 7–12 February 2020; Volume 34, pp. 4667–4674.
40. Shu, J.; Xie, Q.; Yi, L.; Zhao, Q.; Zhou, S.; Xu, Z.; Meng, D. Meta-weight-net: Learning an explicit mapping for sample weighting. In Proceedings of the Advances in Neural Information Processing Systems Vancouver, BC, Canada, 8–14 December 2019.
41. Wang, R.; Mou, S.; Wang, X.; Xiao, W.; Ju, Q.; Shi, C.; Xie, X. Graph structure estimation neural networks. In Proceedings of the Web Conference 2021, Ljubljana, Slovenia, 19–23 April 2021; pp. 342–353.
42. Chen, P.; Chen, G.; Ye, J.; Heng, P.A.; others. Noise against noise: stochastic label noise helps combat inherent label noise. In Proceedings of the International Conference on Learning Representations, Addis Ababa, Ethiopia, 30 April 2020.
43. Hu, W.; Li, Z.; Yu, D. Simple and effective regularization methods for training on noisily labeled data with generalization guarantee. *arXiv* **2019**, arXiv:1905.11368.
44. Li, S.; Liu, T.; Tan, J.; Zeng, D.; Ge, S. Trustable Co-Label Learning From Multiple Noisy Annotators. *IEEE Trans. Multimed.* **2023**, *25*, 1045–1057. [CrossRef]
45. Zhang, Y.; Zheng, S.; Wu, P.; Goswami, M.; Chen, C. Learning with feature-dependent label noise: A progressive approach. *arXiv* **2021**, arXiv:2103.07756.
46. Mazumder, P.; Singh, P.; Nambodiri, V.P. Rnnp: A robust few-shot learning approach. In Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision, Virtual, 5–9 January 2021; pp. 2664–2673.
47. Lu, J.; Jin, S.; Liang, J.; Zhang, C. Robust few-shot learning for user-provided data. *IEEE Trans. Neural Netw. Learn. Syst.* **2020**, *32*, 1433–1447. [CrossRef] [PubMed]
48. Liang, K.J.; Rangrej, S.B.; Petrovic, V.; Hassner, T. Few-shot learning with noisy labels. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, New Orleans, LA, USA, 18–24 June 2022; pp. 9089–9098.
49. Jiang, J.; Ma, J.; Wang, Z.; Chen, C.; Liu, X. Hyperspectral image classification in the presence of noisy labels. *IEEE Trans. Geosci. Remote Sens.* **2018**, *57*, 851–865. [CrossRef]
50. Tu, B.; Zhang, X.; Kang, X.; Wang, J.; Benediktsson, J.A. Spatial density peak clustering for hyperspectral image classification with noisy labels. *IEEE Trans. Geosci. Remote Sens.* **2019**, *57*, 5085–5097. [CrossRef]
51. Jiang, J.; Ma, J.; Liu, X. Multilayer spectral–spatial graphs for label noisy robust hyperspectral image classification. *IEEE Trans. Neural Netw. Learn. Syst.* **2020**, *33*, 839–852. [CrossRef]
52. Chen, T.; Kornblith, S.; Norouzi, M.; Hinton, G. A simple framework for contrastive learning of visual representations. In Proceedings of the International Conference on Machine Learning, Online, 13–18 July 2020; pp. 1597–1607.



53. He, K.; Fan, H.; Wu, Y.; Xie, S.; Girshick, R. Momentum contrast for unsupervised visual representation learning. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, 13–19 June 2020; pp. 9729–9738.
54. Khosla, P.; Teterwak, P.; Wang, C.; Sarna, A.; Tian, Y.; Isola, P.; Maschinot, A.; Liu, C.; Krishnan, D. Supervised contrastive learning. In Proceedings of the Advances in Neural Information Processing Systems, Virtual, 6–12 December 2020.
55. Li, J.; Xiong, C.; Hoi, S.C. Learning from noisy data with robust representation learning. In Proceedings of the IEEE/CVF International Conference on Computer Vision, 11–17 October 2021; pp. 9485–9494.
56. Wu, Z.F.; Wei, T.; Jiang, J.; Mao, C.; Tang, M.; Li, Y.F. Ngc: A unified framework for learning with open-world noisy data. In Proceedings of the IEEE/CVF International Conference on Computer Vision, 11–17 October 2021; pp. 62–71.
57. Li, S.; Xia, X.; Ge, S.; Liu, T. Selective-supervised contrastive learning with noisy labels. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, New Orleans, LA, USA, 18–24 June 2022; pp. 316–325.
58. Yi, L.; Liu, S.; She, Q.; McLeod, A.I.; Wang, B. On learning contrastive representations for learning with noisy labels. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, New Orleans, LA, USA, 18–24 June 2022; pp. 16682–16691.
59. Zhong, Z.; Li, J.; Luo, Z.; Chapman, M. Spectral–spatial residual network for hyperspectral image classification: A 3-D deep learning framework. *IEEE Trans. Geosci. Remote Sens.* **2017**, *56*, 847–858. [[CrossRef](#)]
60. Han, B.; Yao, Q.; Yu, X.; Niu, G.; Xu, M.; Hu, W.; Tsang, I.; Sugiyama, M. Co-teaching: Robust training of deep neural networks with extremely noisy labels. In Proceedings of the Advances in Neural Information Processing Systems, Montreal, QC, Canada, 3–8 December 2018.

**Disclaimer/Publisher’s Note:** The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.