Overcoming Granularity Mismatch in Knowledge Distillation for Few-Shot Hyperspectral Image Classification

Hao Wu, Zhaohui Xue, *Member, IEEE*, Shaoguang Zhou, and Hongjun Su, *Senior Member, IEEE*

***Abstract*— Hyperspectral image classification (HSIC) often struggles due to the scarcity of labeled samples. Knowledge distillation (KD), including self-distillation (SD) where a model learns from its own predictions, has emerged as a promising solution. However, existing distillation methods in HSIC face a "granularity mismatch" problem as they rely on coarse, patch-level data for fine-grained, pixel-level classification, which introduces label noise and causes misclassification. To overcome this issue, we propose central spectral self-distillation (CSSD), a framework that isolates pure spectral information at the patch center and leverages it for SD. CSSD consists of three main components. First, the backbone network separates spectral and spatial feature processing to extract pure central spectral features. Second, a spectral refiner module enhances these spectral features before integrating spatial context. Finally, an SD loss aligns the final predictions with the central spectral guidance, ensuring granularity matching at the pixel level. The experimental results on five hyperspectral datasets demonstrate the effectiveness of CSSD under few-shot conditions. The source code will be available online at https://github.com/ZhaohuiXue/CSSD.**

***Index Terms*— Few-shot learning, granularity mismatch, hyperspectral image (HSI), knowledge distillation (KD), self-distillation (SD).**

I. Introduction

HYPERSPECTRAL imaging captures the spectral characteristics of objects across different wavelengths, enabling precise identification of material composition and state [1], [2]. With advancements in remote sensing technology, hyperspectral image classification (HSIC) has become essential in fields such as agriculture [3], environmental monitoring [4], and resource surveying [5]. As artificial intelligence and remote sensing hardware continue to improve, its applications are expected to expand further, especially in large-scale, fine-grained surface-type surveys [6], [7].

However, HSIC faces significant challenges due to the labor-intensive and costly sample collection [8]. Each class typically requires a substantial number of labeled samples, but obtaining sufficient labeled data is often difficult in practice. This limitation reduces the effectiveness of traditional supervised learning methods, highlighting the urgent need for efficient few-shot learning strategies that can maintain high classification accuracy with limited samples [9].

This article addressing the few-shot problem mainly focuses on two approaches.

1) *External Sample Augmentation*: Methods such as semi-supervised learning [10], [11], active learning [12], [13], and transfer learning [14], [15] introduce additional unlabeled data or labeled data from other domains to mitigate the shortage of labeled samples.
2) *Internal Sample Optimization*: Techniques such as self-supervised learning [16], [17], contrastive learning [18], and meta-learning [19], [20], [21] enhance the model's ability to discriminate between similar and dissimilar samples, improving performance under few-shot conditions.

While these methods can improve classification performance to some extent, they have notable limitations. External sample augmentation must consider distribution discrepancies between internal and external samples, arising from variations in terrain, lighting, sensor parameters, and acquisition times—common issues in hyperspectral remote sensing [22], [23]. Internal sample optimization can suffer from sample selection bias; if the available labeled samples are unrepresentative—due to insufficient ground surveys or limited expert knowledge—the model may learn skewed or incomplete information [24].

Knowledge distillation (KD) offers a promising avenue for extracting richer discriminative information from limited labeled samples [25]. Initially developed for model compression, KD transfers knowledge from a large teacher model to a smaller student model, enabling the student to emulate the teacher's predictive behavior. Its effectiveness arises from leveraging the nonmaximum values in the teacher's predictions—often called "dark knowledge"—which provide richer supervisory signals and guide the model to learn inter-class relationships [26].

Beyond using soft labels, KD has evolved into various forms. Some methods transfer intermediate features from the teacher to the student, allowing the student to learn deeper feature representations [27]. Others distill the teacher's

Received 17 October 2024; revised 19 November 2024 and 19 December 2024; accepted 14 January 2025. Date of publication 16 January 2025; date of current version 4 February 2025. This work was supported in part by the National Natural Science Foundation of China under Grant 42271324 and in part by the Natural Science Foundation of Jiangsu Province under Grant BK20221506. *(Corresponding author: Zhaohui Xue.)*

Hao Wu and Shaoguang Zhou are with the School of Earth Sciences and Engineering, Hohai University, Nanjing 211100, China.

Zhaohui Xue and Hongjun Su are with the College of Geography and Remote Sensing, Hohai University, Nanjing 211100, China (e-mail: zhaohui.xue@hhu.edu.cn).

Digital Object Identifier 10.1109/TGRS.2025.3530614

attention mechanisms, helping the student focus on similar regions as the teacher [28]. Additionally, relational distillation conveys relationships between samples in the teacher model to the student, aiding in better understanding of the data structure [29]. These approaches extend KD beyond output soft labels to include feature-level, attention, and relation knowledge transfer.

Recently, KD has been applied to hyperspectral data analysis tasks such as image fusion [30], anomaly detection [31], target tracking [32], and HSIC [33]. In these applications, it addresses model complexity and computational load due to the high dimensionality of HSIs and improves performance, particularly in classification task [34], [35].

However, teacher–student architectures introduce additional computational and memory overhead. Self-distillation (SD), a specialized form of KD, addresses these issues by having the model simultaneously serves as both teacher and student [36], making it easier to integrate distillation into existing frameworks. This approach can be categorized into three types.

1) The model acts as both teacher and student at the same time: for example, Shang et al. [37] proposed a weighted SD method to tackle class imbalance problem in HSIC by focusing on small and hard samples.
2) Different training stages of the same model serve as teacher and student: Wang et al. [38] introduced an SD regularization that constrains the model to make consistent predictions after domain adaptation, addressing catastrophic forgetting.
3) Different layers of the model act as teacher and student: Qin et al. [39] used an SD strategy from deep to shallow layers, aligning outputs with Kullback–Leibler divergence to improve training stability and generalization in few-shot HSIC.

Despite these advances, we observe that when applying KD to HSIC, existing methods often avoid directly using classification predictions as knowledge sources. For instance, Qin et al. [39] perform distillation between embeddings without employing an auxiliary classifier like in [36], and Yue et al.'s [40] soft labels are derived from a custom spatial–spectral joint distance. The root cause is the "granularity mismatch" problem in KD applied to HSIC. Here, the model's predictions are based on coarse, patch-level data, which do not align with the detailed, pixel-level knowledge needed for accurate classification. As illustrated in Fig. 1(b) and (c), when models incorporate spatial context from neighboring pixels in deeper layers, the predictions become coarser and may misrepresent individual pixel classes. This mismatch leads to semantic inconsistencies between the provided knowledge (coarse, patch-level predictions) and the required knowledge (detailed, pixel-level classifications), introducing label noise during training and misguiding the learning process (Table I). Therefore, existing HSIC methods seldom use distillation strategies that utilize predictive results as knowledge sources, somewhat compromising intuitiveness and interpretability.

To address this issue, we propose a novel method called central spectral self-distillation (CSSD). CSSD mitigates the granularity mismatch problem by decoupling spectral and spatial feature processing and leveraging the uncontaminated spectral features of the central pixel for SD. Specifically, CSSD comprises the following.

1) *Backbone Model:* Shallow layers focus on per-pixel spectral feature extraction without incorporating spatial context, ensuring that the spectral features remain uncontaminated by neighboring pixels. Then, deep layers perform spatial analysis and classification. Between these stages, a spectral refiner module employs deep learnable dilation and attention mechanisms to optimize local spectral features.
2) *CSSD Loss:* We distill the classification results of central spectral features into the final classifier. This helps mitigate issues of scarce supervision and overfitting under few-shot conditions.

By processing spectral features per pixel and independently distilling the central spectral features, CSSD avoids interference from neighboring pixels, aligning the granularity of the knowledge provider and recipient. This ensures that the knowledge transferred during distillation is directly relevant to the pixel-level classification task.

Our main contributions are as follows.

1) *Identifying the Granularity Mismatch Problem:* We clarify the existence of granularity mismatch in patch-based HSIC KD and highlight the critical role of aligning knowledge granularity with pixel-level classification needs.

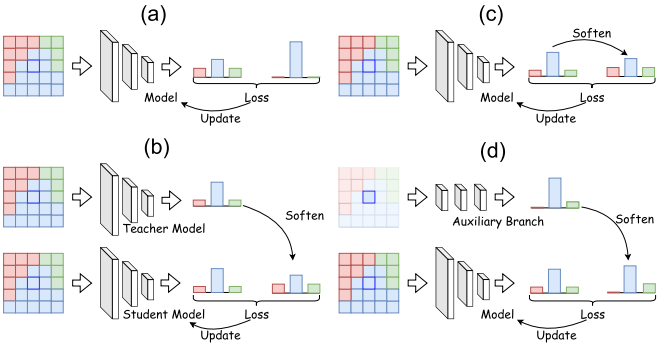


Fig. 1. Illustration of the "granularity mismatch" problem in KD for patch-based HSIC. (a) *Vanilla supervised learning:* direct pixel-level classification using ground-truth labels. (b) *KD:* using a teacher model trained on patches to guide pixel-level predictions introduces a mismatch between coarse-grained knowledge and fine-grained requirements, leading to label noise during training. (c) *Patch-based SD:* the model's own patch-based predictions serve as soft labels for pixel-level tasks, but the granularity mismatch persists. (d) *Center-based SD:* our method focuses on the central pixel's spectral features to provide accurate, pixel-level knowledge, effectively aligning granularity.

TABLE I
COMPARISON OF VANILLA SUPERVISED LEARNING AND DIFFERENT DISTILLATION SCHEMES FOR PATCH-BASED HSIC

Method†	A	B	C	D
Knowledge Provider *Granularity*	Ground Truth Pixel	Teacher Model Patch	Model Patch	Auxiliary Branch Pixel
Knowledge Recipient *Granularity*	Model Pixel	Student Model Pixel	Model Pixel	Model Pixel
Is Granularity Consistent	Yes	No	No	Yes

†Methods A, B, C, and D correspond to the approaches depicted in Fig. 1.

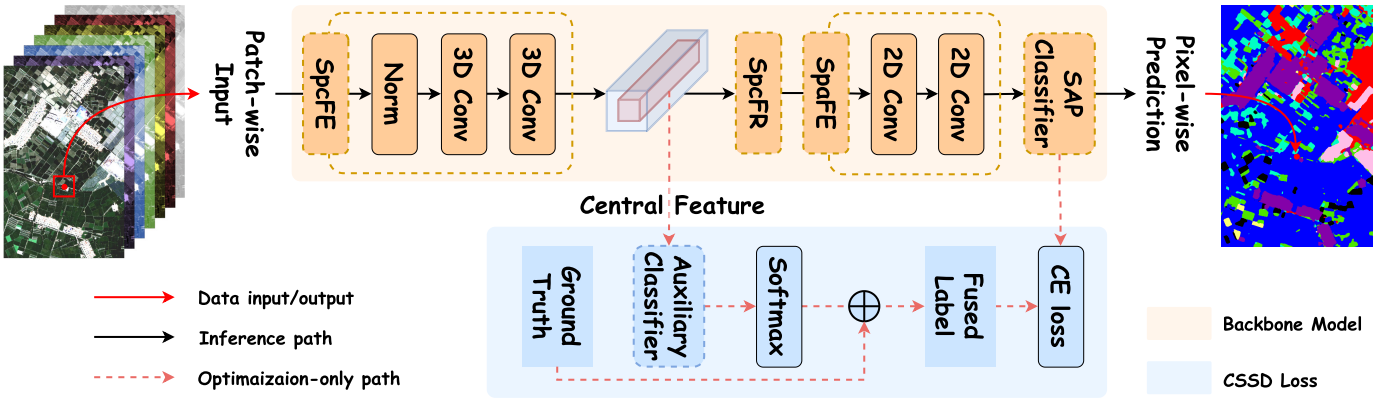


Fig. 2. Overview of the proposed method, including the backbone model and CSSD. The backbone model decouples spectral and spatial feature processing, providing the central pixel's uncontaminated spectral features to CSSD. CSSD distills the accurate "dark knowledge" from the central pixel through an auxiliary classifier, fusing it with the true labels as the model's final prediction target.

2) *Proposing the CSSD Method:* We introduce CSSD, using softened predictions of central spectral feature classifications to guide model updates, effectively resolving the granularity mismatch problem by ensuring semantic consistency and appropriate knowledge granularity.
3) *Designing a Supporting Backbone Model:* We develop a backbone model tailored for CSSD, where spectral features are processed pixel-wisely before spatial operations, enhanced by a spectral refiner module using deep learnable dilation and attention mechanisms.

II. Methodology

This section first provides a brief methodological overview. Then, it details the components of our approach: the backbone model and the center spectral SD. The overall structure is shown in Fig. 2.

A. Methodological Overview of the Proposed Method

Our proposed method focuses on extracting and leveraging the central pixel's spectral features to guide the training of the entire patch. To achieve this, we require a backbone model with a feature extractor that operates exclusively in the spectral domain, allowing us to capture the uncontaminated spectral feature of the central pixel.

To meet this requirement, we have developed a novel backbone model that includes a spectral feature extractor (SpeFE), a spectral feature refiner (SpeFR), a spatial feature extractor (SpaFE), and a final classifier. The SpeFE performs convolution solely along the spectral dimension of the input HSI cube, avoiding any operations in the spatial domain (ensured by setting the spatial kernel size to 1).

Since the proposed center spectral SD is conducted between different depths of the backbone, to ensure that the two levels of knowledge can be aligned meaningfully, we introduce an auxiliary classifier after the SpeFE. The auxiliary classifier accesses only the central spectral features to make predictions, thereby distilling central spectral knowledge into the label space.

Finally, the "dark knowledge" learned by the auxiliary classifier can be transferred to the final classifier. We present detailed explanations of each component in Sections II-B and II-C.

B. Backbone Model

The proposed backbone model employs a spectral-then-spatial design to achieve SD of spectral features. The HSI data undergo four stages: global spectral feature extraction, refined spectral feature processing, spatial feature extraction, and prediction.

1) Spectral Feature Extractor: The SpeFE is designed to handle the high dimensionality and redundancy across bands in raw hyperspectral data. To address this, we use batch normalization (BN), dilated convolution, and wide convolution (convolution with wide kernels), accomplishing global spectral feature extraction.

First, we introduce an additional channel dimension to the input HSI cubes, forming a tensor with the shape Channel $\times$ Spectral $\times$ Width $\times$ Height $(C \times S \times W \times H$, where $C = 1)$. Then, a BN layer is applied to standardize the tensor along the channel dimension

$$\underset{1\times S\times W\times H}{x_{\text{bn}}} = \text{BN}(\underset{1\times S\times W\times H}{x}). \tag{1}$$

Next, we apply a dilated convolutional layer with a $3 \times 1 \times 1$ kernel and a dilation rate of 3 in the spectral dimension. The dilated convolution compresses the spectral dimension from S to approximately $S/3$, using a stride of $(3, 1, 1)$ and padding of $(2, 1, 1)$. At the same time, we increase the channel dimension from 1 to 64 to preserve the information capacity

$$\underset{64\times\lfloor S/3\rfloor\times W\times H}{x_{\text{dilate}}} = \text{Conv}_{3\times1\times1}(x_{\text{bn}}, 64, \text{dilation} = (3, 1, 1)). \tag{2}$$

Then, a convolutional layer is applied with a wide kernel size of $\lfloor S/3 \rfloor \times 1 \times 1$ to further compress the spectral dimension to a single value and then squeeze it. The channel dimension is expanded to 256. To reduce parameters and computational load, we use the group convolution setting with two groups, halving the number of parameters

$$\underset{256\times W\times H}{x_{\text{SpeFE}}} = \text{Conv}_{\lfloor S/3\rfloor\times1\times1}(x_{\text{dilate}}, 256, \text{groups} = 2). \tag{3}$$

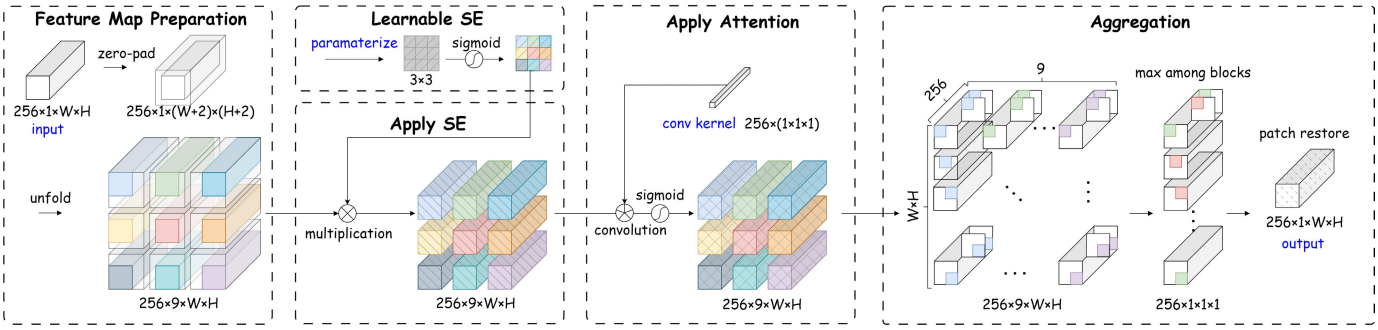


Fig. 3. Workflow of the SpeFR. For better visualization, a structuring element size of 3 × 3 is shown as an illustrative example.

With just two convolutional layers, the SpeFE module achieves local-to-global spectral feature processing, resulting in a highly optimized spectral feature representation.

2) Spectral Feature Refiner: To further enhance model performance, we introduce an innovative SpeFR after SpeFE. The SpeFR is designed to refine spectral features by incorporating spatial context in a learnable and adaptive manner. The SpeFR achieves this by combining two key techniques: *1) deep dilation*: extends traditional morphological dilation to deep feature maps, allowing the model to capture and emphasize important spatial context and *2) attention mechanism*: introduces flexibility by enabling the model to focus on the most relevant parts within each spatial neighborhood.

To provide a comprehensive understanding of SpeFR, we first offer an explanation of deep dilation. Imagine the feature map as an image where each "pixel" contains a high-dimensional spectral feature vector. The deep dilation operation slides a window [structure element (SE)] over this feature map. For each position, it crops a neighborhood around a central pixel and aggregates information within these neighborhoods, highlighting prominent spatial context.

The SpeFR consists of the following steps, as illustrated in Fig. 3 and detailed in Algorithm 1.

1) *Learnable SE:* We define a learnable **SE** of size $w \times h$, representing the spatial neighborhood around each pixel. Then, applying *Sigmoid* activation σ to **SE** ensures that the weights are positive

$$\mathbf{SE}^* = \sigma(\mathbf{SE}). \tag{4}$$

2) *Feature Map Preparation:* We zero-pad the input feature map x_{SpeFE} to accommodate the size of **SE**

$$x_{\text{padded}} = \text{Zero-pad}(x_{\text{SpeFE}}, \lfloor w/2 \rfloor, \lfloor h/2 \rfloor). \tag{5}$$

Then, we extract sliding blocks from the padded feature map by *unfold* operation, resulting in x_{block}. Each block corresponds to a spatial neighborhood of size $w \times h$ around a pixel

$$x_{\text{block}}(i, j) = x_{\text{padded}}(i : i + w - 1, j : j + h - 1). \tag{6}$$

3) *Applying SE:* For each spatial position i in **SE**, we multiply the corresponding block $x_{\text{block}}(:, i, :, :)$ by the learned weight $\mathbf{SE}^*(i)$ to achieve pixel-level weighting

$$x_{\text{dilated}}(:, i, :, :) = x_{\text{block}}(:, i, :, :) \times \mathbf{SE}^*(i). \tag{7}$$

Algorithm 1 Spectral Feature Refiner

Input: $\mathrm{x}_{\text{SpeFE}}$: Input feature map
$\boldsymbol{BB}$: Bounding box for **SE**
Output: $\mathrm{x}_{\text{SpeFR}}$: Refined feature map
Function `SpeFR`$(\mathrm{x}_{\text{SpeFR}}, \boldsymbol{BB})$:
  $\mathbf{SE}^* \leftarrow \sigma(\boldsymbol{BB})$;
  $\mathrm{x}_{\text{padded}} \leftarrow \text{Zero-pad}(\mathrm{x}_{\text{SpeFE}}, \lfloor w/2 \rfloor, \lfloor h/2 \rfloor)$;
  $\mathrm{x}_{\text{block}} \leftarrow \text{Unfold}(\mathrm{x}_{\text{padded}}, w, h)$;
  for $i \leftarrow 1$ **to** $w \times h$ **do**
    $\mathrm{x}_{\text{dilated}} \leftarrow \mathrm{x}_{\text{block}}(:, i, :, :) \cdot \mathbf{SE}^*(i)$;
    $\mathrm{x}_{\text{attention}} \leftarrow \mathrm{x}_{\text{dilated}} \cdot \sigma(\text{Conv}_{1\times1\times1}(\mathrm{x}_{\text{dilated}}))$;
  $\mathrm{x}_{\text{SpeFR}} \leftarrow \max_i(\mathrm{x}_{\text{attention}})$;
  return $\mathrm{x}_{\text{SpeFR}}$

4) *Applying Attention:* To further highlight the most discriminative blocks, we apply a block-level attention mechanism using a 1 × 1 × 1 convolution followed by a Sigmoid activation σ

$$x_{\text{attention}} = x_{\text{dilated}} \cdot \sigma(\text{Conv}_{1\times1\times1}(x_{\text{dilated}})). \tag{8}$$

5) *Aggregation:* We perform the deep dilation by taking the elementwise maximum across all blocks, consistent with multichannel image dilation, which emphasizes the most discriminative responses from each block

$$\underset{256\times W\times H}{x_{\text{SpeFR}}} = \max_i \{x_{\text{attention}}(:, i, :, :)\}. \tag{9}$$

The refined feature map x_{SpeFR} is with the same dimensions as the input feature map x_{SpeFE}.

3) Spatial Feature Extractor: While the SpeFR enhances spectral features by incorporating spatial context, it does not explicitly model spatial patterns. To effectively learn joint spectral–spatial representations, we design a SpaFE. Specifically, a 3 × 3 convolutional layer is applied sequentially to reduce the number of channels from 256 to 64

$$\underset{64\times W\times H}{x_{\text{SpaFE}}} = \text{Conv}_{3\times3}(x_{\text{SpeFR}}, 64). \tag{10}$$

4) SAP Classifier: Following the SpaFE, another 3 × 3 convolution is applied to reduce the number of channels from 64 to the number of classes K:

$$\underset{K\times W\times H}{x_{\text{Cls}}} = \text{Conv}_{3\times3}(x_{\text{SpaFE}}, K). \tag{11}$$

Typically, global average pooling (GAP) is then applied along the spatial dimensions to obtain the average category of x_{SpaFE}, realizing the final classification of the central pixel

$$\underset{K}{\hat{y}_{\text{GAP}}} = \frac{1}{W \times H} \sum_{i,j} \mathbf{x}_{\text{Cls}}(i, j). \tag{12}$$

GAP generally yields good results for contiguous ground objects; however, it can suppress spectral variation within a patch, leading to misclassification at boundaries and loss of small or elongated objects.

To address this problem, we propose a spectral-aware pooling (SAP) approach. We use the output from the SpeFE, which is spatially uncontaminated, to construct a weighted average pooling. Specifically, we compute the cosine similarity between every pixel in the patch and the central pixel using x_{SpeFE}. The weight matrix W is

$$W_{i,j} = \frac{x(i, j) \cdot x(\boldsymbol{c})}{\|x(i, j)\| \|x(\boldsymbol{c})\|}. \tag{13}$$

Here, $x(i, j)$ is the feature vector of the (i, j)th pixel in x_{SpeFE}, and $x(\boldsymbol{c})$ is the central pixel's feature vector. W reflects the spectral similarity of each pixel relative to the central pixel. Before average pooling, the patch's category prediction results are multiplied by W channelwise, so the final prediction is an ensemble vote among spectrally similar pixels

$$\underset{K}{\hat{y}} = \frac{1}{W \times H} \sum_{i,j} W(i, j) \cdot x_{\text{Cls}}(i, j). \tag{14}$$

In this way, the SAP classifier reduces the impact of highly heterogeneous pixels on the final prediction, optimizing mapping details and improving the association between patch features and the central pixel's prediction.

C. Center Spectral SD

KD is effective because it provides the student model with more informative learning targets derived from a knowledgeable source (e.g., an experienced teacher model). In our proposed center spectral SD, we harness this principle by using an auxiliary classifier that processes the uncontaminated spectral features of the central pixel to generate accurate and informative predictions. These granularity-consistent predictions serve as softened learning targets that guide the optimization of the model.

1) Auxiliary Classifier: The auxiliary classifier maps the 256-D global spectral features of the central pixel $x(\boldsymbol{c})$ to the category dimension K using a linear layer (implemented with a 1×1 convolutional layer)

$$\underset{K}{y_{\text{aux}}} = \text{Conv}_{1\times1}(x_{\text{SpeFE}}(\boldsymbol{c}), K). \tag{15}$$

The prediction logits y_{aux} reflect the category tendencies of the central pixel, providing a supervisory signal beyond the ground truth.

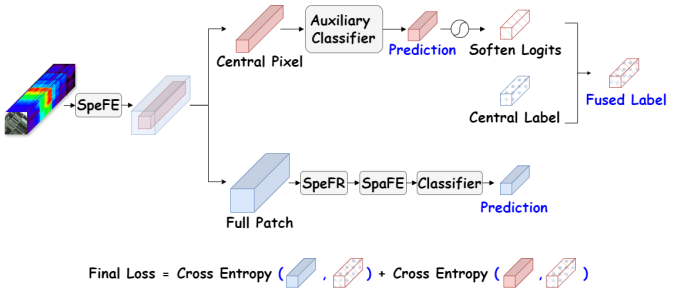


Fig. 4. Illustration of the CSSD loss.

2) CSSD Loss: Before using y_{aux} for model optimization, we must acknowledge that its prediction accuracy is not guaranteed when relying solely on spectral features. Therefore, we prioritize the correctness of the target while gradually incorporating "dark knowledge." Specifically, we apply a weighted fusion to introduce prediction logits into the optimization target with a smaller weight (Fig. 4).

First, a temperature coefficient τ is introduced to adjust the smoothing intensity of the softmax function

$$\bar{y}_{\text{aux}}(\boldsymbol{c}) = \frac{\exp(y_{\text{aux}}(\boldsymbol{c})/\tau)}{\sum_{k=1}^{K} \exp(y_{\text{aux}}(k)/\tau)}. \tag{16}$$

Then, the softened logits $\bar{y}_{\text{aux}}$ is combined with the true label y using a proportion coefficient ϵ to obtain the fused learning target

$$\tilde{y} = \epsilon \cdot \bar{y}_{\text{aux}} + (1 - \epsilon) \cdot y. \tag{17}$$

Finally, we compute the cross-entropy loss between the final classifier's prediction and the learning target, defining the CSSD loss

$$\mathcal{L}_{\text{cssd}} = -\sum_{c=1}^{K} \tilde{y}(\boldsymbol{c}) \log \hat{y}(\boldsymbol{c}). \tag{18}$$

3) Final Loss: Since both the auxiliary classifier and the model classifier share the same label space, the distilled learning target $\tilde{y}$ can also serve as the target for the auxiliary classifier. The loss for the auxiliary classifier is

$$\mathcal{L}_{\text{aux}} = -\sum_{c=1}^{K} \tilde{y}(\boldsymbol{c}) \log y_{\text{aux}}(\boldsymbol{c}). \tag{19}$$

Thus, the final loss combines $\mathcal{L}_{\text{cssd}}$ (CSSD loss) and $\mathcal{L}_{\text{aux}}$ (auxiliary classifier loss). Since both have the same structure, we treat them as parallel batches

$$\begin{aligned} \mathcal{L}_{\text{final}} &= \mathcal{L}_{\text{cssd}} + \mathcal{L}_{\text{aux}} \\ &= -\sum_{c=1}^{K} \tilde{y}(\boldsymbol{c}) \log \hat{y}(\boldsymbol{c}) - \sum_{c=1}^{K} \tilde{y}(\boldsymbol{c}) \log y_{\text{aux}}(\boldsymbol{c}) \\ &= -\sum_{c=1}^{K} [\tilde{y}, \tilde{y}](\boldsymbol{c}) \log[\hat{y}, y_{\text{aux}}](\boldsymbol{c}). \end{aligned} \tag{20}$$

In this way, no extra loss-balancing hyperparameters are required during optimization.

TABLE II
SPECIFICATIONS AND LAND COVER INFORMATION OF EXPERIMENTAL HYPERSPECTRAL DATASETS

Dataset	Pavia University (PU)		Huanghekou (HHK)		Dafeng Natural Reserve (DF)		Heihe Watershed (HH)		Houston University (HU)	
Sensor	ROSIS-3		AHSI ZY1-02D		AHSI GF-5		CASI/SASI		CASI	
Platform	Airborne		Spaceborne		Spaceborne		Airborne		Airborne	
Loc.&Time	Italy, 2001		China, 2019		China, 2020		China, 2012		America, 2017	
GSD	1.3m		30.0m		30.0m		2.4m		1.0m	
Wavelength	$0.43\mu m$ - $0.84\mu m$		$0.39\mu m$ - $2.5\mu m$		$0.39\mu m$ - $2.51\mu m$		$0.38\mu m$ - $2.45\mu m$		$0.38\mu m$ - $1.05\mu m$	
Data Size	610×340×103		1147×1600×119		986×632×256		667×417×149		601×2384×48	
Class No.	Class Name	#	Class Name	#	Class Name	#	Class Name	#	Class Name	#
1	Asphalt	6631	Reed	310	Sea	1020	Corn	43036	Healthy grass	9799
2	Meadows	18649	Spartina alterniflora	187	Mud flat	528	Fragrant-flowered garlic	917	Stressed grass	32502
3	Gravel	2099	Salt filter pond	247	Road	150	Cauliflower	7629	Artificial turf	684
4	Trees	3064	Salt evaporation pond	300	Building	89	Bell pepper	11644	Evergreen trees	13588
5	Painted metal sheets	1345	Dry pond	140	Farmland	918	Potato	4072	Deciduous trees	5048
6	Bare soil	5029	Tamarisk	127	River	276	Endive sprout	803	Bare earth	4516
7	Bitumen	1330	Salt pan	306	Trees	724	Watermelon	2040	Water	266
8	Self-blocking bricks	3682	Seep weed	218	Aquaculture	746	Artificial surfaces	29035	Residential buildings	39762
9	Shadows	947	River	584	Lake	79	-	-	Non-residential buildings	223684
10	-	-	Sea	4694	-	-	-	-	Roads	45810
11	-	-	Mudbank	14	-	-	-	-	Sidewalks	34002
12	-	-	Tidal creek	67	-	-	-	-	Crosswalks	1516
13	-	-	Fallow land	459	-	-	-	-	Major thoroughfares	46358
14	-	-	Ecological restoration pond	310	-	-	-	-	Highways	9849
15	-	-	Robinia	111	-	-	-	-	Railways	6937
16	-	-	Fishpond	124	-	-	-	-	Paved parking lots	11475
17	-	-	Pit pond	128	-	-	-	-	Unpaved parking lots	149
18	-	-	Building	398	-	-	-	-	Cars	6578
19	-	-	Bare land	87	-	-	-	-	Trains	5365
20	-	-	Paddyfield	508	-	-	-	-	Stadium seats	6824
21	-	-	Cotton	332	-	-	-	-	-	
22	-	-	Soybean	71	-	-	-	-	-	
23	-	-	Corn	103	-	-	-	-	-	
Total	-	42776	-	9825	-	4530	-	99176	-	504712

III. EXPERIMENTAL RESULTS

A. Experimental Settings

1) Hyperspectral Datasets: To evaluate the effectiveness of the proposed method, we selected five diverse hyperspectral datasets: Pavia University (PU), Huanghekou (HHK), Dafeng Natural Reserve (DF), Heihe Watershed (HH), and Houston University (HU). These datasets were chosen for their spectral and spatial diversities, as well as their representation of different land cover classes. Detailed information about each dataset is provided in Table II.

2) Sample Selection: To assess performance under limited labeled data, we employed an N-shot classification approach, using only N labeled samples per class. The remaining labeled samples were used for testing. Unless stated otherwise, we used five labeled samples per class for the PU, DF, and HH datasets. For the HHK and HU datasets, which have much more classes and are more challenging, we increased this to ten samples per class to ensure effective learning.

3) Compared Methods: To validate the effectiveness of our method, we conducted comparison experiments with several popular and state-of-the-art models. These include five backbone models: SSRN [41], SSFTT [42], MATA [43], DBCTNet [44], and MiM [45]; and five few-shot learning methods: CMFSL [19], FSCF-SSL [16], DM-MRN [20], SCFormer [21], and SPFormer [17]. For all compared methods, hyperparameters were set to their recommended values. Except for FSCF-SSL and SCFormer, which fix the patch size at 33 and 7, respectively, all other methods used a patch size of 9.

4) Implementation Details: For CSSD, we used the ADAM optimizer with $\beta_1 = 0.9$, $\beta_2 = 0.999$, and a weight decay of 0.0001. The model was trained for 150 epochs with an initial learning rate of 0.001, which decayed according to a cosine schedule. The batch size was set to $5 \times K$, where K is the number of classes.

All experiments were conducted using Python 3.8, PyTorch 1.13.1, and cuDNN 11.7.1 on a workstation equipped with a GeForce RTX 3080 (10 GB) GPU.

5) Evaluation and Metrics: To ensure a fair comparison, each method underwent ten independent experiments using randomly selected samples, with results averaged across these runs. Performance was evaluated using classwise accuracy (CA), overall accuracy (OA), average accuracy (AA), and the kappa coefficient (κ). We also include computational metrics such as model parameter size (Param), training time (T_{train}), and test time (T_{test}).

B. Parameter Sensitive Analysis

1) Patch Size and SE Size: In our backbone model, the patch size and the size of the SE are two hyperparameters. The patch size determines the spatial context captured, while the SE size affects the level of spatial abstraction during feature refinement. We investigated how these parameters influence classification accuracy.

As shown in Fig. 5, the optimal patch size varies among datasets. For PU, HHK, and HU, performance generally improves with larger patch sizes up to 9. In contrast, for DF and HH, performance peaks between patch sizes of 5 and 9 and then declines.

Regarding the SE size, the impact appears consistent across datasets. The most significant improvement is observed when increasing the SE size from 1 to 3. This suggests that a

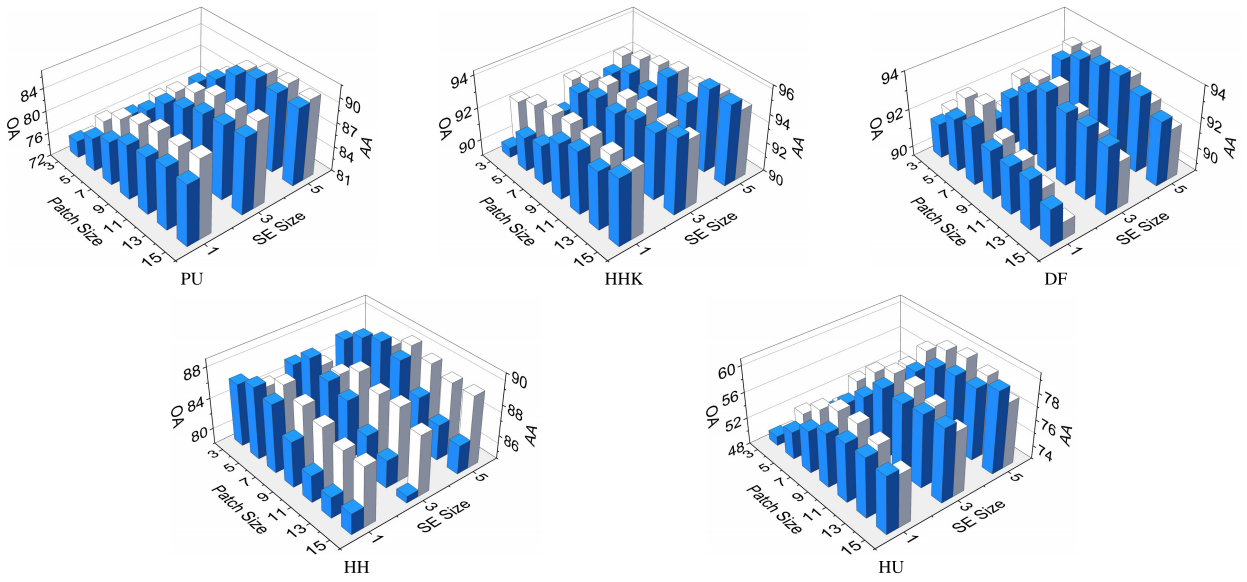


Fig. 5. Parameter sensitive analysis of the size of HSI patch and SE. Blue bars represent OAs, and gray bars denote AAs.

TABLE III
CLASSIFICATION RESULTS ON THE PU DATASET, WITH FIVE SAMPLES PER CLASS. THE BEST OVERALL RESULTS ARE HIGHLIGHTED

Method	Non-Few-shot Methods					Few-shot Methods					
	SSRN [41]	SSFTT [42]	MATA [43]	DBCTNet [44]	MiM [45]	CMFSL [19]	FSCF-SSL [16]	DM-MRN [20]	SCFormer [21]	SPFormer [17]	Ours
Asphalt	84.40±8.87	71.80±7.09	73.21±11.75	76.48±12.09	79.68±5.68	75.33±9.93	73.90±11.56	77.80±7.70	66.29±11.72	**85.23±9.97**	84.06±6.93
Meadows	69.71±12.78	77.36±10.65	78.12±11.20	75.42±14.52	75.98±12.75	78.76±12.91	80.61±7.85	**82.28±7.32**	86.25±7.32	70.79±8.38	79.91±11.54
Gravel	85.74±10.16	74.67±16.99	76.78±13.85	84.55±9.19	83.49±11.08	75.71±14.19	86.08±12.41	84.83±11.10	65.69±20.99	83.79±13.62	**91.54±7.94**
Trees	90.09±7.77	**96.75±2.08**	90.85±9.14	88.29±8.86	86.32±8.22	92.61±3.28	94.62±3.51	91.31±8.96	85.41±14.87	91.91±5.44	90.55±8.23
Painted metal sheets	99.67±0.54	99.72±0.68	99.40±0.79	99.75±0.36	99.71±0.46	99.51±0.84	99.98±0.03	99.87±0.20	99.24±1.18	**100.00±0.00**	**100.00±0.00**
Bare soil	82.83±8.79	78.95±12.21	71.88±9.55	87.19±9.32	90.52±12.21	86.30±7.42	88.47±9.88	78.37±13.72	60.13±2.98	**95.13±4.11**	89.04±13.39
Bitumen	97.15±2.06	89.21±4.72	92.86±4.85	95.48±4.57	96.92±2.75	86.81±7.76	97.09±5.87	98.11±1.38	79.22±14.91	**99.67±0.50**	99.47±0.92
Self-blocking bricks	81.24±12.83	61.97±18.70	71.71±16.16	76.71±11.96	72.95±8.41	71.65±11.93	**91.31±10.81**	55.27±14.55	63.27±17.98	81.45±12.81	80.46±11.03
Shadows	98.35±2.77	98.97±1.62	**99.72±0.33**	96.40±4.09	95.52±2.92	97.02±4.07	97.57±4.53	99.42±0.68	99.04±1.19	97.76±2.49	99.67±0.67
OA (%)	79.19±5.52	78.16±4.73	78.52±5.93	80.29±6.06	80.93±4.78	80.65±5.60	84.18±3.88	80.99±2.91	77.51±3.15	81.36±3.52	**84.68±3.66**
AA (%)	87.69±2.08	83.26±2.58	83.84±2.14	86.70±2.40	86.79±2.07	84.85±1.94	89.96±2.99	85.25±2.09	78.28±1.69	89.53±2.18	**90.52±1.79**
κ	0.74±0.06	0.72±0.06	0.73±0.07	0.75±0.07	0.76±0.05	0.75±0.07	0.80±0.05	0.76±0.04	0.70±0.04	0.77±0.04	**0.81±0.04**
Params (K)	396.99	489.15	691.93	**16.36**	79.52	614.01	3156.23	1817.82	7234.02	41.9	435.63
T_{train} (s)	47.9	4.28	5.5	**3.35**	21.38	460.62	262.18	57.34	2148.36	4.55	6.43
T_{test} (s)	4.41	1.17	1.6	**0.6**	39.93	1.85	13.8	27.88	6.66	2.78	2.42

dot-shaped SE (size 1) does not provide sufficient refinement because it fails to capture information from neighboring pixels to enhance the features. Larger SE sizes beyond 3 offer moderate gains but at the cost of increased memory consumption during the *unfold* operation.

To balance performance and computational efficiency, we selected a patch size of 9 and an SE size of 3 for all datasets in subsequent experiments.

2) Temperature Coefficient and Epsilon Coefficient: We further analysis the impact of the temperature coefficient (τ) and the proportion coefficient (ϵ), which are two hyperparameters with respect to the proposed CSSD loss. τ fine-tunes the smoothing intensity of the softmax output to incorporate "dark knowledge," while ϵ controls how much of this "dark knowledge" is fused with the true label.

As illustrated in Fig. 6, increasing τ generally improves performance across all datasets, with optimal values typically between 2 and 8. However, excessively high τ values can lead to slight accuracy declines, especially for DF and HH.

For ϵ, choosing smaller generally produces better results by effectively integrating auxiliary predictions without overshadowing the true labels. Setting ϵ above 0.2 can lead to a noticeable decrease in accuracy, especially for the PU, DF, and HHK datasets. However, the HH dataset is an exception and benefits from ϵ values as high as 0.4.

In summary, moderate softening (τ between 2 and 4) and a smaller proportion of "dark knowledge" (ϵ between 0.1 and 0.2) offer the best tradeoff between stability and accuracy. In subsequent experiments, we set τ to 4 for PU, HHK, HU, and DF, and 2 for HH. The ϵ values were set to 0.1 for PU, HHK, and HU, 0.2 for DF, and 0.3 for HH.

C. Comparison of Classification Performance

Tables III–VII present detailed classification results for the five datasets. Corresponding classification maps are shown in Figs. 7–11.

1) Accuracy: Generally, few-shot methods outperform the non-few-shot counterparts, with our method standing out as

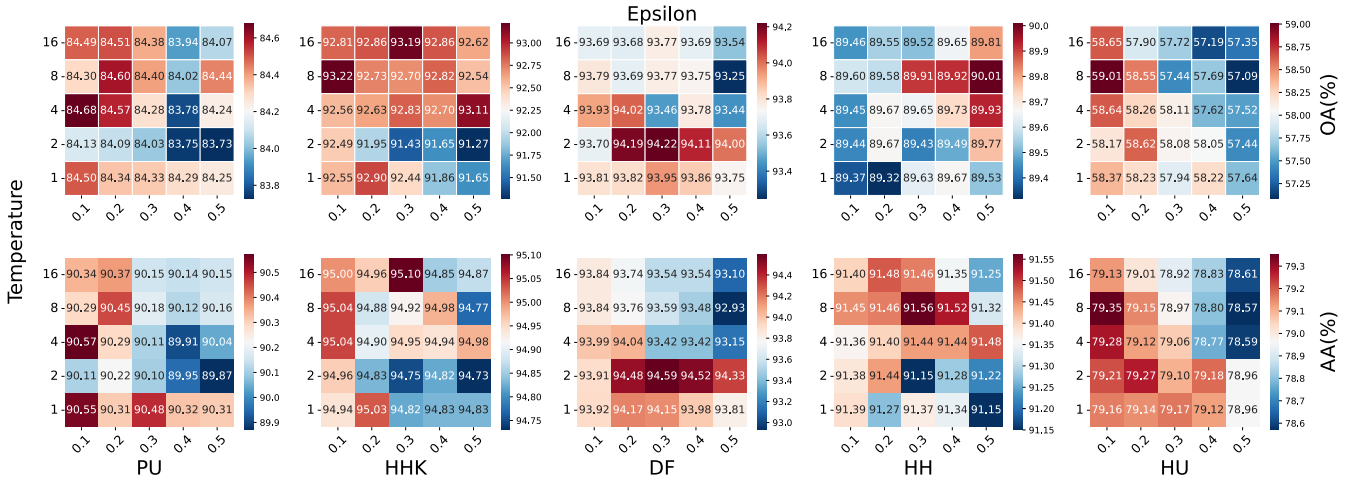


Fig. 6. Parameter sensitive analysis of CSSD with respect to the temperature coefficient (Temperature, τ) and the proportion coefficient (Epsilon, ϵ).

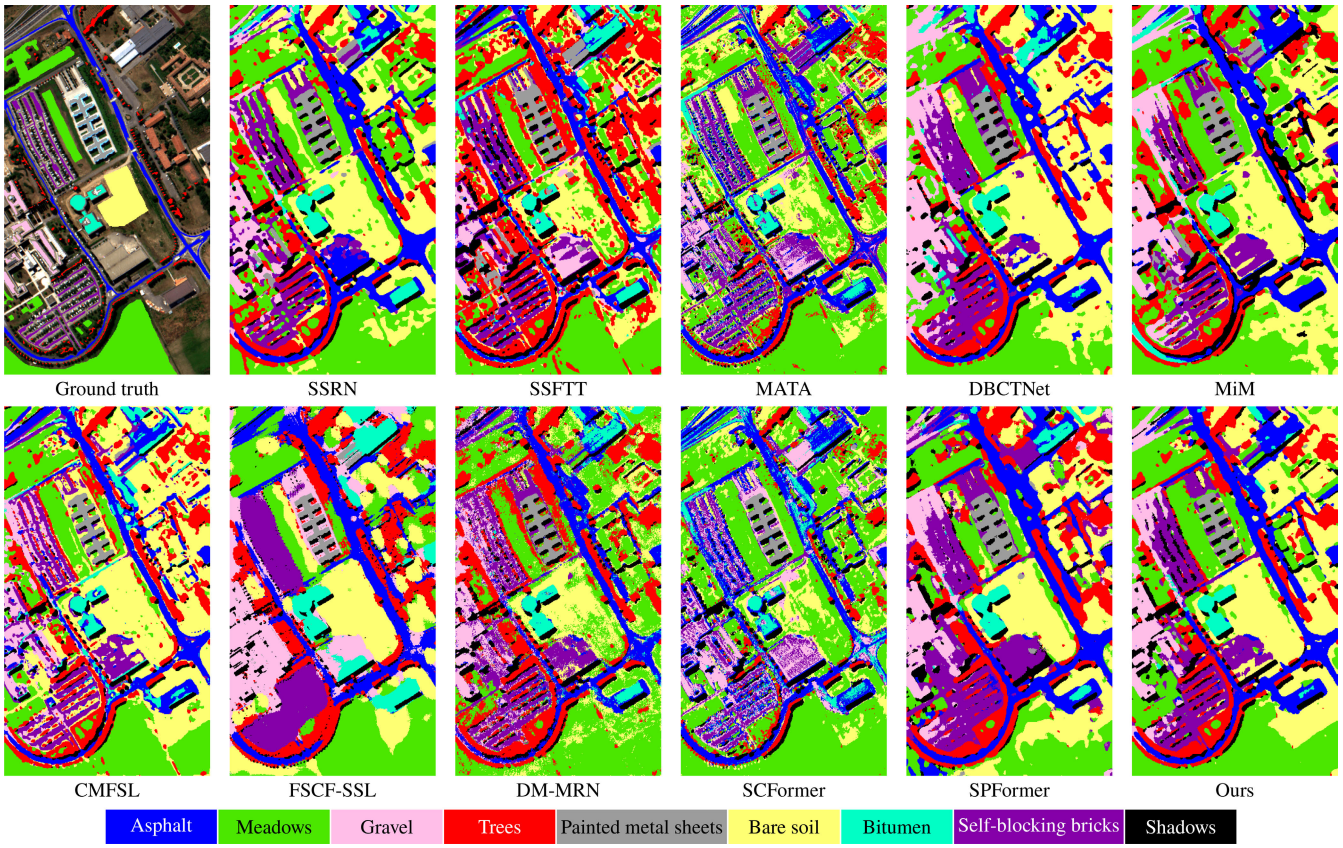


Fig. 7. False-color image with ground truth and predictions maps of the PU dataset obtained by compared methods.

the best across five different datasets. Specifically, among non-few-shot methods, CSSD shows minimum OA advantages of 3.75%, 1.16%, 2.48%, 0.77%, and 3.57% for PU, HHK, DF, HH, and HU, respectively. Against other few-shot methods, CSSD maintains OA advantages of at least 0.5%, 0.6%, 1.03%, 0.38%, and 3.77%. These results clearly show the benefits of few-shot methods in our testing scenarios. Additionally, our approach produces top results in AA and kappa coefficient (κ), confirming its effectiveness.

Regarding CA, our method achieved the highest accuracy for half of the categories in the HU dataset. It shows notable CA with structured objects such as *residential buildings*, *nonresidential buildings*, *crosswalks*, and *paved parking lots*, showing minimum CA advantages of 3.23%, 2.09%, 6.59%, and 6.52%, respectively. This performance enhancement can be attributed to our method's focus on central pixels, which enhances predictions near object edges. For the other datasets, while our method did not consistently achieve noticeable CA, its overall lead in AA underscores that the interclass insights

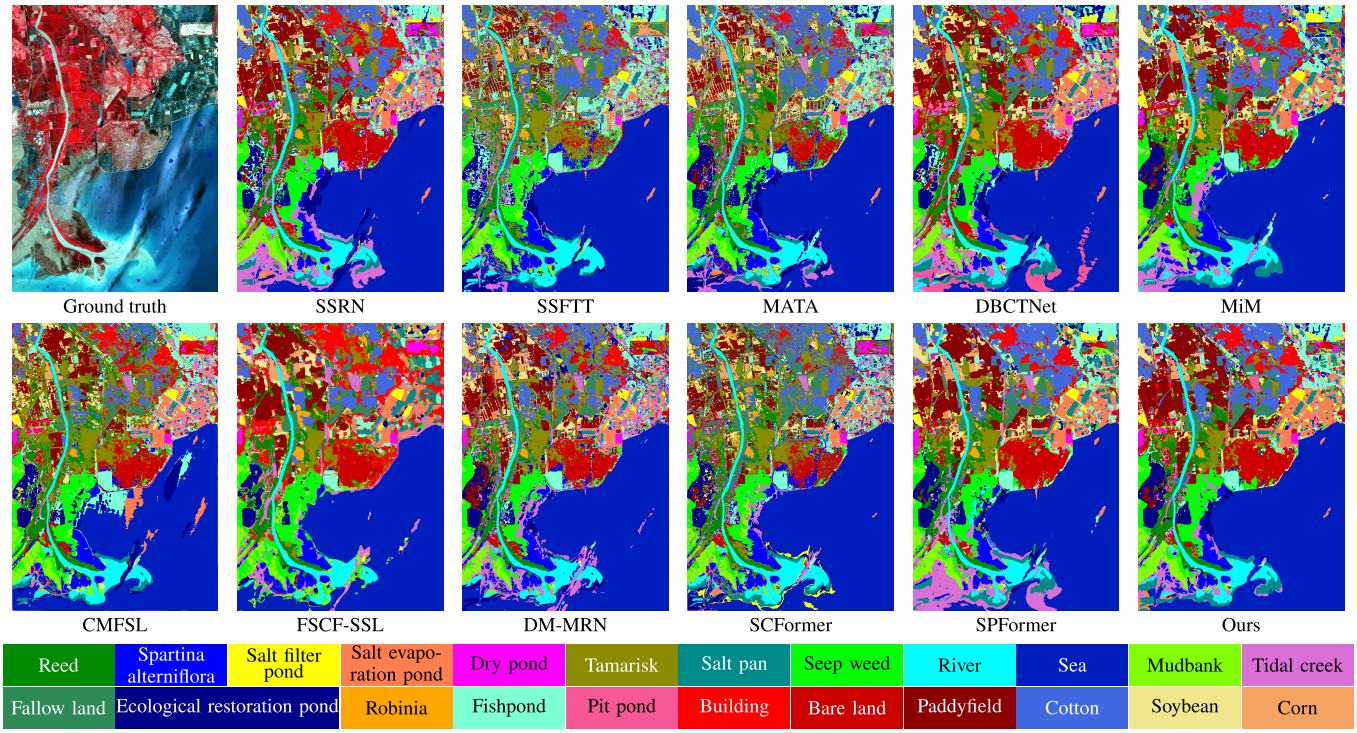


Fig. 8. False-color image with ground truth and predictions maps of the HHK dataset obtained by compared methods.

TABLE IV

CLASSIFICATION RESULTS ON THE HHK DATASET, WITH TEN SAMPLES PER CLASS. THE BEST OVERALL RESULTS ARE HIGHLIGHTED

Method	Non-Few-shot Methods					Few-shot Methods					
	SSRN [41]	SSFTT [42]	MATA [43]	DBCTNet [44]	MiM [45]	CMFSL [19]	FSCF-SSL [16]	DM-MRN [20]	SCFormer [21]	SPFormer [17]	Ours
Reed	67.77±9.33	63.77±12.17	67.60±9.25	57.40±9.78	72.70±11.88	68.33±6.82	59.87±9.01	67.97±7.13	48.87±9.72	**77.23±10.42**	71.97±9.34
Spartina alterniflora	95.76±4.02	96.78±3.16	**98.25±2.93**	95.25±3.77	97.01±2.37	96.89±1.52	95.82±3.20	98.08±2.77	93.05±11.24	93.73±1.86	97.68±2.80
Salt filter pond	98.52±1.88	98.40±2.67	98.23±5.18	95.78±11.58	98.90±2.20	99.62±1.14	97.13±5.39	99.16±1.27	93.38±12.21	**100.00±0.00**	98.78±3.67
Salt evaporation pond	98.10±4.43	99.10±2.13	95.03±7.29	**99.66±0.93**	95.28±6.84	98.45±2.10	91.83±7.32	97.10±3.94	91.07±19.68	96.93±5.00	99.21±2.06
Dry pond	98.15±2.82	93.31±12.00	97.54±3.01	98.00±2.76	96.31±3.31	97.85±2.70	91.16±4.76	97.54±3.01	91.85±15.99	97.54±2.87	97.23±2.90
Tamarisk	92.05±3.06	95.64±2.11	97.78±1.88	86.75±2.10	88.46±2.76	91.97±3.57	88.21±6.03	95.21±3.40	84.70±7.73	88.21±1.42	89.23±2.24
Salt pan	**100.00±0.00**	99.80±0.34	99.29±0.57	**100.00±0.00**	99.66±1.01	99.97±0.10	98.34±1.65	99.19±1.71	96.08±11.09	**100.00±0.00**	**100.00±0.00**
Seep weed	93.22±9.95	85.29±25.40	95.43±9.47	97.26±2.34	92.50±11.01	95.15±9.43	97.65±2.46	**98.70±1.05**	95.44±7.98	89.47±11.36	95.24±9.83
River	99.74±0.73	99.67±0.72	**100.00±0.00**	**100.00±0.00**	**100.00±0.00**	99.95±0.16	98.73±1.66	**100.00±0.00**	99.70±0.73	99.81±0.57	**100.00±0.00**
Sea	87.56±11.05	90.31±13.13	89.87±10.70	84.16±11.49	89.27±6.88	88.45±10.48	**97.70±1.50**	90.72±5.53	89.47±12.05	79.10±8.13	91.22±9.90
Mudbank	**100.00±0.00**	**100.00±0.00**	**100.00±0.00**	**100.00±0.00**	95.71±12.86	**100.00±0.00**	**100.00±0.00**	**100.00±0.00**	97.14±8.57	94.29±17.14	**100.00±0.00**
Tidal creek	91.58±6.32	90.35±7.78	88.07±10.95	90.53±7.16	88.77±6.29	88.77±8.60	85.79±8.61	**95.61±5.22**	78.25±17.76	88.42±5.72	91.23±6.23
Fallow land	95.86±5.42	90.91±9.03	96.10±4.60	92.49±6.38	96.82±5.08	93.52±7.18	84.77±5.30	95.03±4.49	94.83±4.28	**96.86±5.48**	96.06±5.50
Ecological restoration pond	90.43±7.86	88.23±8.13	85.20±8.45	92.83±7.88	93.73±5.57	91.43±7.53	77.43±9.49	91.37±3.94	78.53±18.65	93.47±5.71	**93.60±6.37**
Robinia	99.90±0.30	98.22±3.67	97.92±3.05	98.32±4.43	99.21±2.38	98.42±1.99	95.45±3.92	99.11±1.50	90.40±5.48	99.70±0.89	**100.00±0.00**
Fishpond	96.84±3.73	93.60±5.71	92.63±5.01	93.77±6.22	99.21±1.59	92.63±4.58	90.61±6.09	97.63±2.42	86.67±18.94	**99.91±0.26**	99.39±1.36
Pit pond	95.17±6.43	92.46±7.03	93.14±6.16	86.10±28.83	94.24±6.19	93.31±5.72	81.19±10.59	**95.34±4.24**	84.07±12.00	94.32±5.48	94.58±6.28
Building	93.84±4.59	88.58±4.90	89.25±5.76	91.16±2.66	94.59±3.90	92.96±5.12	78.20±10.64	**97.71±2.69**	80.18±19.59	95.49±3.16	94.95±3.77
Bare land	99.09±1.31	99.87±0.39	**100.00±0.00**	98.70±2.60	95.58±6.01	99.74±0.52	98.31±2.25	99.35±1.56	94.81±8.44	99.48±0.86	99.09±2.73
Paddyfield	89.62±8.27	85.60±8.43	86.16±7.71	85.52±5.15	89.82±4.65	85.88±6.01	85.74±6.76	**92.15±3.00**	78.82±14.29	90.24±6.34	90.34±6.46
Cotton	80.84±8.45	76.86±13.83	81.37±8.86	77.89±13.70	90.34±3.92	81.49±8.44	69.53±18.14	77.17±4.19	74.41±10.52	86.27±6.04	**88.42±6.32**
Soybean	98.36±2.64	95.41±6.30	98.69±1.77	96.23±5.87	98.03±3.34	95.57±4.21	85.90±10.06	96.56±4.89	89.34±6.23	**100.00±0.00**	99.84±0.49
Corn	**98.92±0.00**	**98.92±0.00**	98.17±1.93	**98.92±0.00**	98.71±0.65	98.81±0.32	93.87±3.26	**98.92±0.00**	93.76±14.11	**98.92±0.00**	**98.92±0.00**
OA (%)	90.36±5.73	90.46±7.82	90.92±5.26	87.70±6.01	91.69±3.49	90.52±5.26	92.09±1.38	92.25±2.39	87.74±9.77	86.80±4.07	**92.85±4.96**
AA (%)	93.97±1.40	92.22±2.89	93.29±1.22	92.03±2.12	94.12±1.80	93.44±1.20	88.84±1.29	94.77±0.46	87.17±9.82	93.89±1.76	**95.09±1.36**
κ	0.88±0.07	0.88±0.09	0.88±0.06	0.84±0.07	0.89±0.04	0.88±0.06	0.90±0.02	0.90±0.03	0.84±0.12	0.83±0.05	**0.91±0.06**
Params (K)	453.85	563.79	724.01	18.9	80.43	614.01	3181.75	1830.62	7239.92	**42.36**	488.3
T_{train} (s)	215.48	7.27	14.35	9.71	79.05	528.04	466.21	928.87	2418.22	**10.02**	25.81
T_{test} (s)	1.23	0.3	0.39	0.67	9.66	0.46	3.34	41.79	1.77	**0.8**	0.61

gained from KD help achieve a good balance across different classes.

2) Complexity: We also evaluated the space and computational complexity of each method. The space complexity was measured by the model's parameter size. For the proposed method, parameter size mainly relates to the spectral dimensions (S) due to the wide convolutional layer in SpeFE module [see (3)]. For most cases, our model's parameter size is generally comparable to SSRN, placing it on the lower end among the methods we analyzed. A relatively lower parameter size helps prevent overfitting with limited training data while maintaining model effectiveness.

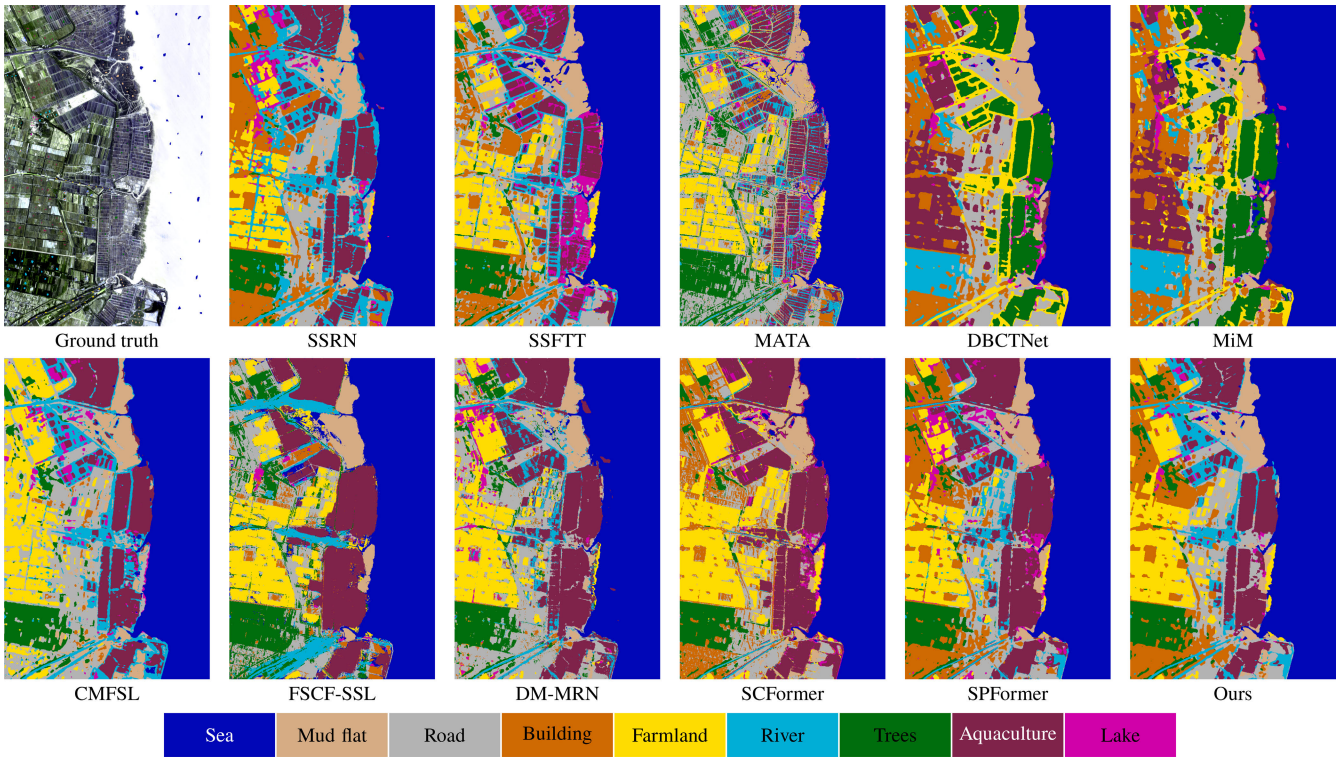


Fig. 9. False-color image with ground truth and predictions maps of the DF dataset obtained by compared methods.

TABLE V

CLASSIFICATION RESULTS ON THE DF DATASET, WITH FIVE SAMPLES PER CLASS. THE BEST OVERALL RESULTS ARE HIGHLIGHTED

Method	Non-Few-shot Methods					Few-shot Methods					
	SSRN [41]	SSFTT [42]	MATA [43]	DBCTNet [44]	MiM [45]	CMFSL [19]	FSCF-SSL [16]	DM-MRN [20]	SCFormer [21]	SPFormer [17]	Ours
Sea	**100.00±0.00**	**100.00±0.00**	**100.00±0.00**	**100.00±0.00**	99.64±1.09	**100.00±0.00**	**100.00±0.00**	**100.00±0.00**	**100.00±0.00**	**100.00±0.00**	**100.00±0.00**
Mudflat	86.92±11.53	87.92±13.75	83.71±18.26	88.97±10.67	80.11±20.75	91.74±8.53	89.16±7.49	**95.77±6.44**	94.74±7.71	90.33±8.02	93.82±7.82
Road	72.14±16.95	**90.00±13.11**	85.10±5.93	76.76±16.72	63.38±12.57	77.52±14.26	80.90±10.04	88.41±7.48	74.00±17.39	76.07±13.48	84.48±15.27
Building	94.29±5.16	**99.88±0.36**	95.60±5.61	93.10±4.57	89.88±5.00	95.00±4.99	98.21±2.83	98.21±2.83	95.95±4.59	94.64±4.43	99.17±1.69
Aquaculture	77.95±10.72	83.14±8.10	78.83±12.19	85.81±5.15	70.93±9.85	89.09±5.13	79.98±9.04	87.03±9.00	87.45±5.81	79.39±8.11	**91.52±3.99**
Farmland	91.81±7.04	96.83±5.42	81.92±11.59	95.83±5.01	88.78±8.46	96.79±3.65	89.04±10.72	93.80±6.05	86.13±10.77	95.65±4.96	**99.89±0.17**
River	85.16±5.78	86.37±10.58	88.64±7.39	86.68±4.48	84.66±8.18	90.21±2.08	**93.00±6.32**	89.78±2.69	90.58±2.13	85.59±4.72	89.01±4.42
Trees	94.41±3.88	94.43±3.63	85.92±6.78	95.64±3.19	97.98±2.85	95.05±4.74	**98.53±1.79**	92.85±4.62	97.80±1.55	97.85±2.57	94.13±3.02
Lake	98.92±1.89	86.22±12.11	91.76±9.62	97.16±4.62	**100.00±0.00**	88.92±10.04	95.41±7.36	98.24±2.77	82.30±9.83	99.86±0.41	98.38±4.86
OA (%)	89.16±2.01	91.31±3.99	87.85±2.11	91.79±2.62	86.84±3.15	91.59±1.76	91.91±2.06	93.24±2.28	92.91±2.00	90.87±2.02	**94.27±1.76**
AA (%)	89.07±2.24	91.64±4.02	87.94±2.21	91.10±3.26	86.15±2.70	93.23±1.52	91.58±1.51	93.79±1.66	89.88±2.83	91.04±2.29	**94.49±2.62**
κ	0.87±0.02	0.90±0.05	0.86±0.02	0.90±0.03	0.84±0.04	0.92±0.02	0.90±0.02	0.92±0.03	0.91±0.02	0.89±0.02	**0.93±0.02**
Params (K)	933.25	1194.18	926.94	**38.25**	79.52	614.01	3390.16	1820.42	7249.32	41.9	853.42
T_{train} (s)	61.22	**2.98**	4.92	5.06	14.9	273.41	586.83	533.11	2134.51	3.15	21.34
T_{test} (s)	14.65	0.29	0.3	0.74	4.29	**0.2**	2.77	51.9	0.8	0.4	0.54

TABLE VI

CLASSIFICATION RESULTS ON THE HH DATASET, WITH FIVE SAMPLES PER CLASS. THE BEST OVERALL RESULTS ARE HIGHLIGHTED

Method	Non-Few-shot Methods					Few-shot Methods					
	SSRN [41]	SSFTT [42]	MATA [43]	DBCTNet [44]	MiM [45]	CMFSL [19]	FSCF-SSL [16]	DM-MRN [20]	SCFormer [21]	SPFormer [17]	Ours
Corn	68.82±9.38	91.12±4.51	88.30±3.79	78.94±8.90	73.52±7.27	89.83±3.38	84.11±8.83	86.23±5.27	**91.94±2.51**	59.45±9.94	87.95±7.14
Fragrant-flowered Garlic	85.86±9.78	78.18±10.79	70.57±11.29	87.86±8.25	90.31±3.46	80.57±6.23	65.58±14.44	91.24±7.92	60.82±16.72	**93.89±4.52**	88.72±7.77
Cauliflower	85.47±10.66	81.14±9.23	68.72±17.37	84.10±11.46	79.99±11.36	76.52±8.24	78.59±10.32	81.16±6.89	71.48±15.84	84.58±9.16	**86.41±10.04**
Bell pepper	93.01±8.21	90.82±9.49	86.76±6.42	**96.83±2.44**	96.44±1.40	94.06±2.06	95.60±2.63	93.30±6.75	86.95±4.01	94.28±5.44	96.82±1.03
Potato	86.48±11.79	91.27±8.59	89.38±6.58	87.45±9.29	90.45±6.58	95.02±4.46	81.14±5.41	**96.68±2.49**	90.06±3.98	85.32±12.12	94.22±4.26
Endive sprout	98.73±1.27	99.11±1.01	94.79±2.89	98.72±1.83	**99.54±0.57**	97.34±2.45	91.52±9.56	99.21±0.49	91.20±7.00	99.51±0.62	99.46±0.29
Watermelon	84.24±9.79	83.92±5.34	81.71±9.09	75.97±17.96	73.83±13.40	86.90±4.14	85.48±9.70	84.96±6.69	76.98±7.50	81.58±14.92	**89.55±6.31**
Artificial surfaces	89.05±7.13	86.88±6.11	78.64±7.07	89.79±6.88	81.20±11.42	86.00±6.85	89.31±5.59	**93.21±5.34**	85.19±5.89	86.36±5.37	89.27±6.63
OA (%)	80.30±5.07	88.88±3.40	83.58±3.61	85.14±4.43	80.03±5.54	88.28±2.02	86.35±3.76	89.27±2.38	87.13±2.94	75.51±4.10	**89.65±3.63**
AA (%)	86.46±2.85	87.80±2.34	82.36±2.31	87.46±2.46	85.66±2.13	88.31±3.06	83.92±2.22	90.75±1.29	81.83±2.63	85.62±2.08	**91.55±1.25**
κ	0.74±0.06	0.85±0.05	0.78±0.05	0.80±0.05	0.74±0.07	0.84±0.04	0.82±0.05	0.85±0.03	0.82±0.04	0.69±0.05	**0.86±0.05**
Params (K)	163.63	559.25	701.06	980.04	762.05	614.01	3226.96	1814.55	7238.31	**41.86**	557.68
T_{train} (s / 100 it.)	1.75	27.62	**0.85**	1.34	1.32	3.11	35.12	53.86	22.33	2.09	4.77
T_{test} (s)	5.47	**1.2**	4.04	6.27	4.93	4.22	63.24	81.78	17.25	9.13	7.27

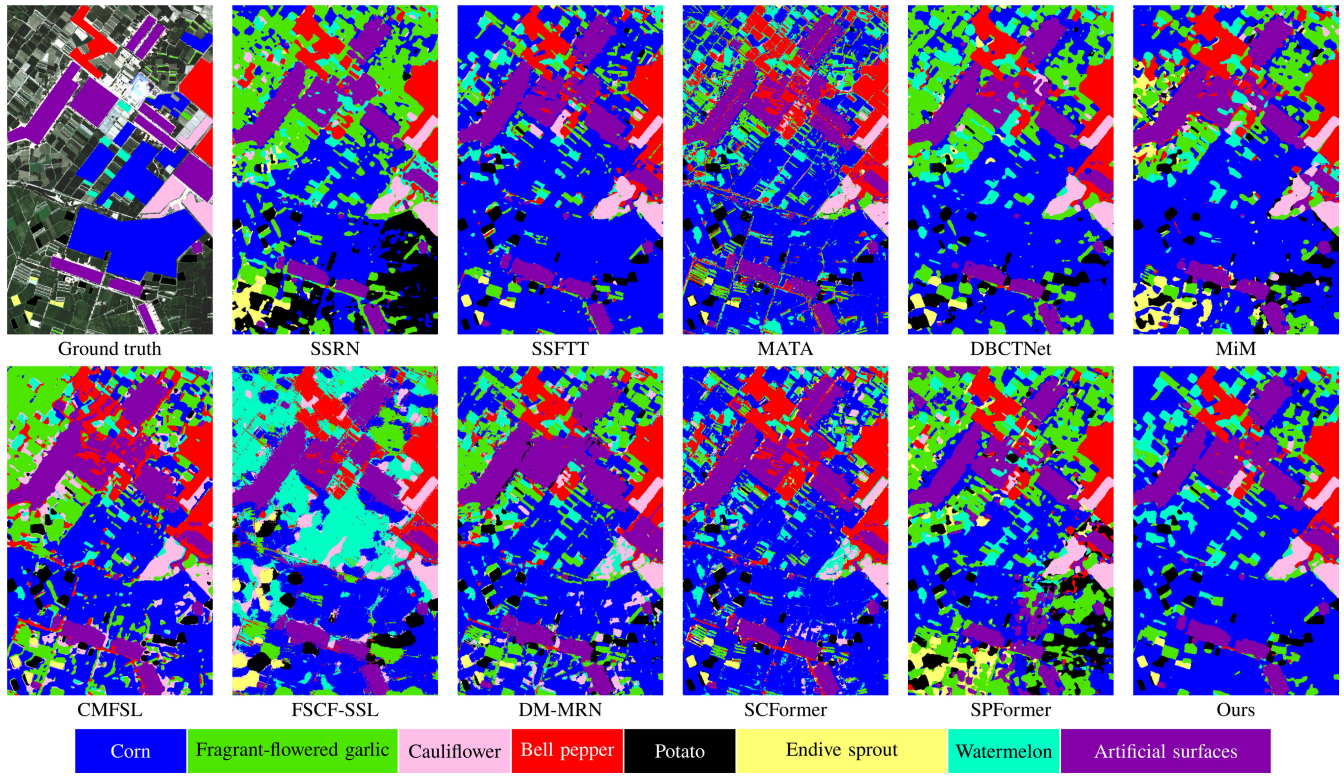


Fig. 10. False-color image with ground truth and predictions maps of the HH dataset obtained by compared methods.

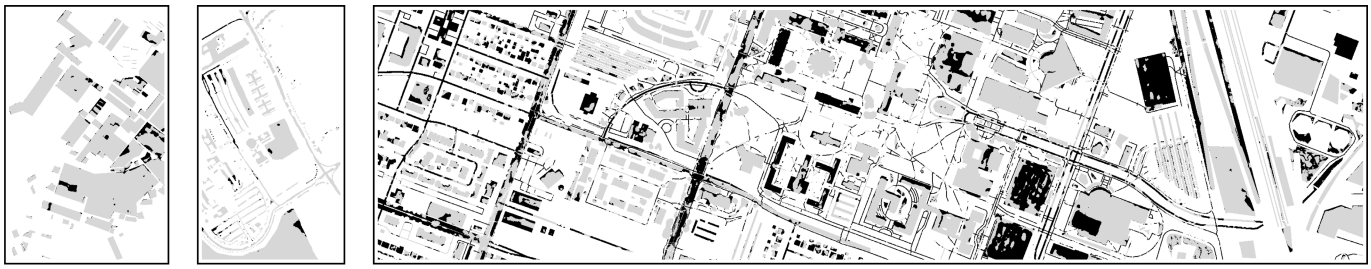
Fig. 11. False-color image with ground truth and predictions of the HU dataset obtained by compared methods.

For computational complexity, we evaluated training and testing times. For fairness, we reported training times over 100 iterations (it.) for all methods. SSRN and our approach, both being convolutional neural network (CNN)-based methods, exhibited similar inference speeds. Transformer-based methods such as SSFTT, MATA, DBCTNet, and SPFormer were the fastest among all. Although MiM employs four types of Mamba-Cross-Scan operations—which significantly increase its computational costs—it is still much faster than episodic training-based few-shot methods such as CMFSL, FSCF-SSL, DM-MRN, and SCFormer. Overall, our method demonstrates lower time complexity compared to others, particularly because it converges in just 150 epochs.

TABLE VII

CLASSIFICATION RESULTS ON THE HU DATASET, WITH TEN SAMPLES PER CLASS. THE BEST OVERALL RESULTS ARE HIGHLIGHTED

Method	Non-Few-shot Methods					Few-shot Methods					
	SSRN [41]	SSFTT [42]	MATA [43]	DBCTNet [44]	MiM [45]	CMFSL [19]	FSCF-SSL [16]	DM-MRN [20]	SCFormer [21]	SPFormer [17]	Ours
Healthy grass	**88.00±7.23**	84.09±14.42	87.79±7.89	87.65±8.09	80.78±7.73	87.28±8.55	81.86±8.12	83.11±5.92	84.32±9.37	78.99±6.41	84.80±7.91
Stressed grass	74.72±6.28	62.55±19.96	**77.10±5.07**	74.34±5.43	68.25±6.42	76.00±4.91	61.06±14.74	72.25±5.04	75.65±6.04	66.76±7.76	74.28±5.58
Artificial turf	**100.00±0.00**	95.77±11.31	99.97±0.09	99.87±0.15	99.91±0.19	98.32±2.25	99.81±0.45	99.99±0.04	94.36±6.83	**100.00±0.00**	**100.00±0.00**
Evergreen trees	93.23±3.23	86.55±8.83	92.52±4.21	92.77±3.96	90.83±4.56	94.01±4.21	94.48±3.32	92.92±2.99	90.96±1.83	92.64±3.37	**94.88±3.28**
Deciduous trees	76.34±5.30	79.31±5.92	82.38±2.79	79.53±4.44	78.24±5.84	74.81±3.01	60.45±13.09	80.55±3.16	71.46±5.42	79.74±4.34	**84.93±3.05**
Bareearth	91.15±6.15	78.06±13.89	91.35±5.66	85.45±11.75	**95.59±3.16**	89.12±6.78	91.19±5.78	95.04±4.14	88.12±4.56	91.91±5.64	92.63±4.89
Water	**100.00±0.00**	99.77±0.26	99.38±0.26	99.61±0.39	99.57±0.59	99.61±0.30	99.88±0.25	99.80±0.59	89.26±7.99	99.88±0.18	**100.00±0.00**
Residential buildings	67.90±6.17	63.95±14.48	71.23±3.23	72.16±6.82	68.33±5.38	72.68±8.01	71.65±14.74	69.52±6.50	58.78±9.37	70.27±5.71	**75.91±3.27**
Non-residential buildings	47.69±8.92	38.87±6.39	37.75±6.57	36.48±5.10	46.71±6.91	51.02±5.02	46.76±21.10	46.45±8.86	46.16±11.38	43.81±6.78	**53.11±4.87**
Roads	**33.57±7.98**	24.09±13.58	24.48±6.84	20.64±9.17	30.57±6.36	30.39±6.43	14.14±12.17	25.79±8.10	26.34±15.30	27.59±5.56	31.85±6.31
Sidewalks	38.50±5.05	35.50±12.04	34.13±4.83	24.16±7.33	36.40±5.17	35.20±4.68	19.33±20.84	18.56±4.23	29.56±5.54	**41.98±5.32**	38.88±5.46
Crosswalks	51.56±6.52	43.49±9.28	48.30±6.64	36.33±11.60	47.03±8.31	43.78±3.87	49.65±20.89	44.44±7.11	32.08±8.91	50.42±10.09	**58.15±5.72**
Major thoroughfares	44.37±7.96	34.96±19.26	35.33±6.38	28.73±16.42	45.49±6.16	33.63±11.12	39.07±23.02	30.25±7.98	33.64±15.22	**49.71±5.68**	47.91±5.01
Highways	92.17±4.42	70.72±18.95	87.41±5.21	85.30±15.98	88.62±3.20	81.47±7.38	93.00±5.96	**93.52±4.07**	76.56±12.71	91.03±4.88	92.37±5.55
Railways	95.78±7.13	72.66±36.20	96.01±3.38	97.59±1.39	92.56±8.28	96.08±2.76	87.71±10.98	**98.07±2.17**	93.84±3.27	94.70±8.43	95.73±7.79
Paved parking lots	79.10±5.37	62.58±18.46	80.55±5.52	71.40±13.69	77.40±6.75	72.98±6.80	70.77±16.80	79.27±4.73	60.56±2.53	77.36±7.35	**87.07±3.16**
Unpaved parking lots	**100.00±0.00**	94.39±13.29	**100.00±0.00**	**100.00±0.00**	**100.00±0.00**	99.93±0.22	**100.00±0.00**	**100.00±0.00**	97.72±1.85	**100.00±0.00**	**100.00±0.00**
Cars	88.45±4.41	87.22±8.62	75.63±4.73	80.99±14.86	85.40±3.78	80.64±9.30	84.41±8.42	87.55±6.25	53.74±13.90	88.48±5.51	**90.09±4.15**
Trains	88.98±7.62	82.44±9.04	83.00±11.71	79.98±10.37	85.40±10.76	80.59±9.67	90.30±5.10	**94.77±3.16**	71.64±10.76	88.98±6.88	91.73±8.13
Stadium seats	90.38±7.43	84.95±10.34	87.81±9.39	89.30±7.72	92.55±6.00	86.41±7.19	91.22±7.01	**94.37±2.96**	81.05±10.03	89.68±6.09	93.03±6.85
OA (%)	55.41±4.14	46.83±2.97	49.17±3.28	46.60±2.63	53.92±3.09	55.21±2.59	50.31±7.86	51.60±3.55	50.03±5.19	53.31±3.65	**58.98±2.04**
AA (%)	77.09±1.31	69.10±5.14	74.61±1.42	72.11±1.94	75.48±1.15	74.20±1.46	72.34±1.88	75.31±0.75	67.79±1.92	76.20±1.31	**79.37±0.97**
κ	0.49±0.04	0.40±0.03	0.43±0.03	0.40±0.02	0.47±0.03	0.47±0.02	0.43±0.06	0.45±0.03	0.42±0.05	0.46±0.04	**0.52±0.02**
Params (K)	199.74	236.43	613.34	**8.48**	80.24	300.66	3070.75	1825.12	7231.9	42.26	297.37
T_{train} (s / 100 it.)	52.33	5.79	12.73	**3.26**	67.55	851.42	1111.81	389.65	3238.84	5.07	7.25
T_{test} (s)	16.67	**8.42**	13.9	10.71	472.74	24.64	1133.26	821.02	89.94	27.08	19.23



HH PU HU

Fig. 12. Difference maps between ground truth and CSSD predictions for the PU, HH, and HU datasets. White regions □ indicate unlabeled samples, gray regions ▒ represent correctly classified pixels, and black regions ■ denote prediction errors compared to the ground truth.

3) Classification Maps: The classification maps reveal differences in mapping characteristics among the methods. MATA and SCFormer not only create more detailed maps, but also tend to produce salt-and-pepper noise. On the other hand, many convolutional methods, such as SSRN and FSCF-SSL, yield smoother outputs, which often obscure details, like *self-blocking bricks* in the PU dataset. Our method finds a good balance; for example, it accurately captures river channels in both the HHK and DF datasets and also effectively identifies different vegetable patches in the HH dataset.

To assess the effectiveness of our method in handling edge classification, we present difference maps between the ground truth and our predictions for the HH, PU, and HU datasets in Fig. 12. These maps enable a visual evaluation of edge pixel classification accuracy using our methods. It is important to note that meaningful edge analysis for the HHK and DF datasets is not feasible due to their fragmented labeling styles, which make edge-based evaluations challenging.

In the HH dataset, which represents agricultural fields with features annotated as rectangular blocks, CSSD achieves relatively good edge segmentation performance. Although some errors are present, the overall edge misclassification is minimal. For the PU and HU datasets, which primarily depict urban environments, the annotations include both large homogeneous regions (such as buildings and grasslands) and narrow linear features like roads. In the PU dataset, CSSD performs well in handling the edges of large features, partly because the annotations avoid labeling exact edges during the annotation process. However, in the HU dataset, there are some narrow misclassifications along building edges. When it comes to road edges, CSSD shows good performance in the PU dataset, except for some systematic misclassifications in the upper left portion of the image. In contrast, CSSD faces challenges with road edges and surfaces in the HU dataset. For example, there is significant misclassification of *major thoroughfares* in the lower part of the image and widespread errors on smaller roads throughout the entire image. In summary, CSSD performs well in handling edges of regular or large homogeneous regions but shows limitations in accurately classifying edges of narrow linear features such as roads.

D. Impact of Training Sample Size

To evaluate how different methods perform with varying training sample sizes, we tested the 11 methods using

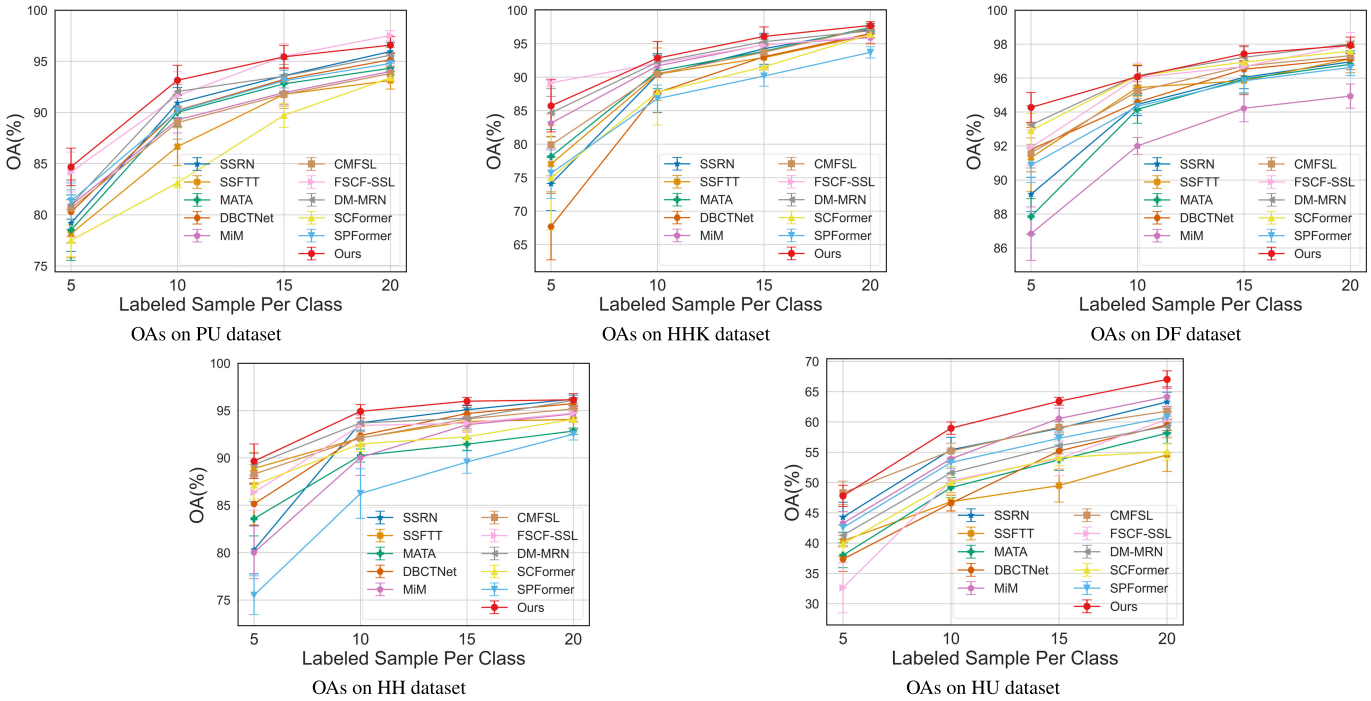


Fig. 13. Classification performance of different methods on five datasets with varying amounts of training samples.

additional few-shot configurations of 10, 15, and 20 samples per class. For classes with limited labeled data, we used a bisected split (for instance, *Mudband* in HHK, 14 samples split as [5, 9] for five shots and [7, 7] for 10, 15, and 20 shots).

As depicted in Fig. 13, OA across the datasets generally improves with more training samples. Few-shot methods such as CMFSL, FSCF-SSL, DM-MRN, SCFormer, SPFormer, and our CSSD method tend to achieve higher performance as the sample size increases. CSSD often attains the highest accuracy. For instance, in the ten-shot case on the PU dataset, CSSD achieves an OA of 93.10%, surpassing FSCF-SSL's of 91.63% and DM-MRN's of 92.05%.

However, in the five-shot case of HHK, FSCF-SSL largely outperforms CSSD and others. This is primarily due to FSCF-SSL's high performance on the dominant class *Sea*, facilitated by FSCF-SSL's large patch size (33) and the contiguous nature of the *Sea* class. When increasing the labeled sample per class, most methods improved considerably, but FSCF-SSL's accuracy only marginally increased due to the performance saturation of the dominant class.

Methods not specifically designed for few-shot tasks, such as SSRN, SSFTT, MATA, DBCTNet, and MiM, generally perform worse, especially with smaller sample sizes. For example, in the HHK dataset with just five training samples, DBCTNet's OA is 67.70%, while CSSD achieves 85.17%, a gap of 17.47%. When the number of training samples increases to 20, DBCTNet's OA rises to 96.53%, slightly below CSSD's of 97.73%. This indicates that methods not optimized for few-shot learning are significantly affected by limited labeled data, underscoring the importance of few-shot algorithms.

IV. Discussion

A. Ablation Analysis

To assess the individual and combined contributions of each module in the proposed method, we conducted a comprehensive ablation study. The results are presented in Table VIII, where each row incrementally incorporates different modules into the baseline model. Specifically, we evaluate the impact of the following modules.

1) *Baseline:* The basic version of our model that includes only foundational components (SpeFE + SpaFE + GAP Classifier).
2) *SpeFR:* Enhances spectral features by refining them.
3) *SAP:* Implements a pooling mechanism that leverages spectral similarity.
4) *SD:* Applies SD solely at the final classification layer.
5) *Patch Spectral Self-Distillation (PSSD):* Utilizes spectral features from the entire patch for auxiliary classification and SD.
6) *CSSD:* Employs spectral features of only the central pixel for auxiliary classification and SD.
7) *Center Spectral Auxiliary Classification (CSAC):* Performs auxiliary classification without applying SD.

In the following, we delve into the specific contributions and interactions of the modules.

1) Baseline: The baseline model serves as our reference point. Across all datasets, the baseline achieves moderate performance, reflecting the solid groundwork of our initial architecture.

2) SpeFR: Introducing SpeFR brings a noticeable performance uplift. On average, we observe increases of 1.08% in OA and 0.38% in AA. The improvement is particularly

TABLE VIII

ABLATION COMPARISON OF EACH VARIANT OF OUR METHOD IN TERMS OF OA AND AA. THE BEST AND SECOND-BEST RESULTS ARE HIGHLIGHTED IN BOLD AND UNDERLINED, RESPECTIVELY. FOR CLARITY, THE STD VALUES ARE OMITTED

Methods	PU		HHK		DF		HH		HU	
	OA (%)	AA (%)	OA (%)	AA (%)	OA (%)	AA (%)	OA (%)	AA (%)	OA (%)	AA (%)
Baseline	80.80	89.03	92.60	94.75	91.89	92.29	87.83	90.80	56.45	78.16
Baseline + **SpeFR**	82.71	89.03	92.80	94.79	93.09	93.16	88.06	90.85	58.29	79.11
Baseline + SpeFR + **SAP**	83.11	89.72	**93.23**	94.98	93.59	93.43	88.96	91.17	58.62	79.24
Baseline + SpeFR + SAP + **SD**	83.34	89.92	92.68	94.80	93.40	92.75	89.13	91.20	58.10	79.30
Baseline + SpeFR + SAP + **PSSD**	83.03	89.90	92.65	94.80	93.42	92.97	88.97	91.11	58.24	79.27
Baseline + SpeFR + SAP + **CSSD**	**84.68**	**90.52**	92.85	**95.09**	**94.27**	**94.49**	**89.65**	**91.55**	**58.98**	**79.37**
Baseline + SpeFR + SAP + **CSAC**	84.23	89.99	92.81	94.95	93.91	93.93	89.53	91.35	58.26	79.24

evident in the DF dataset, where OA jumps from 91.89% to 93.09%, and AA rises from 92.29% to 93.16%. These results underscore SpeFR's effectiveness in refining spectral features, enabling the model to capture more representative information inherent in hyperspectral data.

3) SAP: Switching from GAP to SAP further enhances performance, raising the average OA by 0.51% and AA by 0.32%. For instance, in the HHK dataset, OA improves from 92.80% to 93.23% and AA from 94.79% to 94.98%. These improvements demonstrate that by reducing the influence of heterogeneous pixels, SAP can perform more effective pooling.

4) SD: Incorporating SD, which applies SD only at the final classification layer, results in a slight decline in performance. On average, OA decreases by 0.17% and AA by 0.11%. For example, in the DF dataset, in the DF dataset, OA drops from 93.59% to 93.40% and AA from 93.43% to 92.75%. Similarly, the HU dataset experiences a minor OA decline from 58.62% to 58.10%. These subtle reductions suggest that SD at the final layer may introduce misleading supervisory information, slightly hindering performance.

5) PSSD: Utilizing spectral features from the entire patch, PSSD also results in slightly adverse effects, resulting in an average decrease of 0.07% in OA and 0.03% in AA [negligible differences considering the standard deviation (STD)]. These negative changes indicate that leveraging the entire patch's spectral information may also introduce irrelevant or noisy information.

6) CSSD: The CSSD module stands out as the most impactful enhancement. By focusing exclusively on the central pixel's spectral features for auxiliary classification, CSSD achieves an average OA increase of 0.62% and an AA boost of 0.59%. This is clearly illustrated in the DF dataset, where OA rises from 93.40% to 94.27% and AA from 92.75% to 94.49%. The PU dataset also benefits significantly, with OA rising by 1.57% from 83.34% to 84.68% and AA increasing from 89.92% to 90.52%. These substantial gains across multiple datasets validate CSSD's effectiveness in distilling knowledge from the most relevant and uncontaminated spectral information, thereby significantly enhancing the model's discriminative capabilities.

7) CSAC: To isolate the impact of the auxiliary classifier, we removed the SD from CSSD, creating the CSAC variant. CSAC achieved performance between CSSD and the other variants. For example, in the PU dataset, CSAC increased the OA from 83.11% (Baseline + SpeFR + SAP) to 84.23% though it did not reach the 84.68% achieved by CSSD. Essentially, auxiliary classification forms a multitask learning strategy, where the model simultaneously optimizes for both the main classification task and the auxiliary task of classifying the central pixel's spectral features, thereby enhancing the learning of spectral features, which can partly explain the performance gains observed in CSSD.

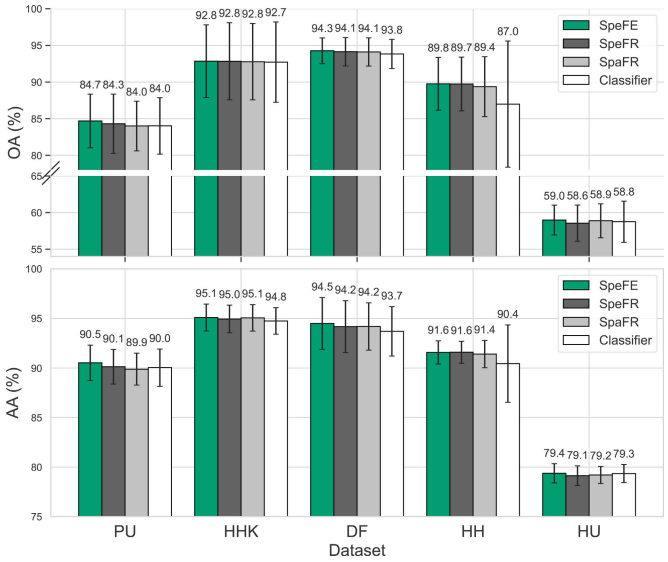


Fig. 14. Classification results of CSSD utilizing various features in SAP.

B. Impact of Feature Selection in SAP

To provide a more comprehensive analysis, we investigated which features are most effective for weight calculation in the SAP. Our experiment considered four features: x_{SpeFE} (**SpeFE**), x_{SpeFR} (**SpeFR**), x_{SpaFE} (**SpaFE**), and x_{Cls} (**Classifier**). The experimental results are shown in Fig. 14.

The results clearly demonstrate that using the output features of SpeFE for SAP weighting significantly improves model performance across all datasets. For instance, in the PU dataset, using SpeFE features for weighting results in an OA of 84.68% and an AA of 90.52%, outperforming results when using features from SpeFR, SpaFE, or Classifier. This improvement is mainly because SpeFE extracts

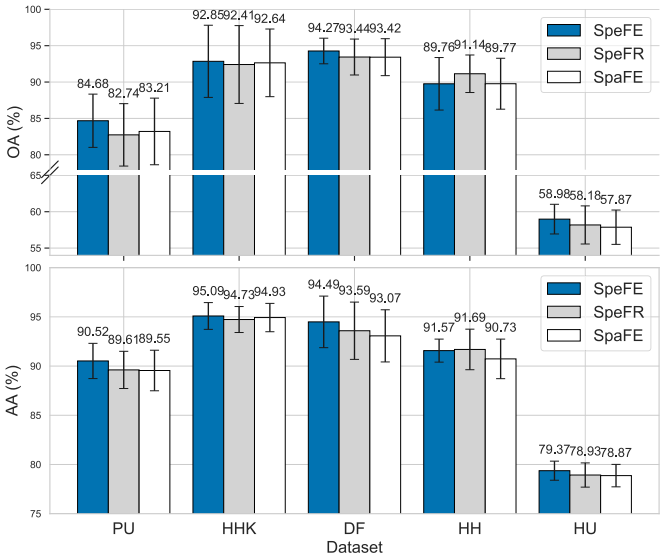


Fig. 15. Classification results of CSSD with different knowledge providers.

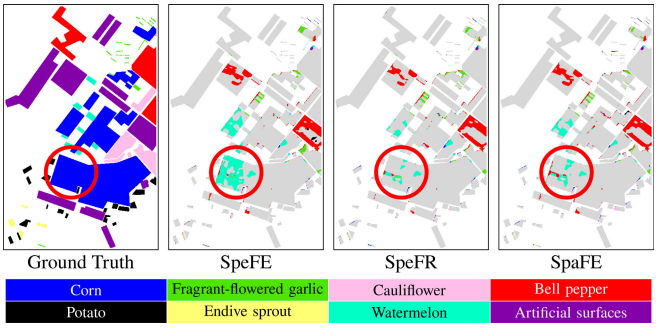


Fig. 16. Difference maps between ground truth and predictions for the HH dataset in the sixth randomized experiment. Distillation after SpeFE (middle left) reveals notable misclassifications, as emphasized by the red circle.

pure spectral features without contamination from neighboring pixels, ensuring accuracy when calculating spatial correlations.

C. Optimal Knowledge Provider

We also investigated which module within our backbone model serves as the optimal knowledge provider for distillation. Specifically, we tested three potential providers: the SpeFE, the SpeFR, and the SpaFE. As shown in Fig. 15, distilling from SpeFE generally yields the best performance across most datasets. This is because SpeFE captures central spectral characteristics without spatial contamination, thereby ensuring consistent granularity between the knowledge provider and the recipient.

However, there was a twist with the HH dataset, where distilling after SpeFR was better than after SpeFE. Looking closer, we found that in one of the ten randomized experiments, using SpeFE led to a significant drop in accuracy due to misclassifications in the first class (*Corn*), as shown in Fig. 16. Specifically, two out of the five *Corn* samples were flagged as outliers because their spectral signatures fell outside the 10th to 90th percentile ranges (see Fig. 17, left). These outliers could confuse the model, as the knowledge distilled from them would be skewed.

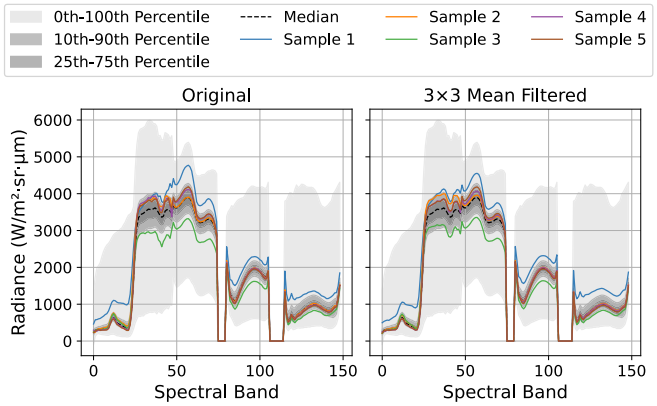


Fig. 17. Spectral signatures of random samples from the first class in the HH dataset. (Left) Original spectral signatures. (Right) Spectral signatures after applying a 3 × 3 mean filter.

To figure out why SpeFR did better here, we applied a 3 × 3 mean filter to smooth the spectral features, which is like adding some spatial context. The smoothed spectral signatures show that the outliers became closer to the median (see Fig. 17, right), suggesting that the spatial operations in SpeFR helped deal with these outliers by using information from nearby pixels.

To sum up, while SpeFE generally serves as an effective knowledge provider due to its pure spectral features, it can also magnify the influence of noise or outliers.

V. Conclusion

In this work, we proposed the CSSD method as a novel approach for few-shot HSIC. CSSD effectively addresses the granularity mismatch problem by decoupling spectral and spatial feature processing and using the central pixel's uncontaminated spectral features for SD, ensuring that the knowledge provided and required are aligned at the pixel level, enabling accurate and effective knowledge transfer. Extensive experiments on five diverse hyperspectral datasets demonstrate that CSSD outperforms state-of-the-art methods under few-shot conditions, validating its effectiveness in mitigating the challenges posed by limited labeled samples and granularity mismatch in HSIC.

While CSSD shows promising results, its performance can be influenced by the quality and representativeness of the training samples. For future work, we plan to explore the integration of active learning techniques to enhance the representativeness and robustness of samples, aiming to reduce the impact of outliers and improve performance under even more challenging conditions. Additionally, extending the CSSD framework to other domains, such as hyperspectral unmixing tasks, and exploring its applicability in self-supervised or semi-supervised settings could further broaden its impact.

Acknowledgment

The authors would like to thank Dr. P. Gamba for providing the Reflective Optics Spectrographic Imaging System data over Pavia, Italy, Dr. W. Sun at Ningbo University, China, for providing the GF-5 DF dataset and ZY-1 02D Yellow

River Delta dataset, and the IEEE Geoscience and Remote Sensing Society (GRSS) Data Fusion Technical Committee for providing the HU 2018 dataset. They would also like to thank Dr. Z. Zhong for sharing the code of SSRN, Dr. L. Sun for SSFTT, Dr. H. Liu for MATA, Dr. X. Dong for DBCTNet, Dr. W. Zhou for MiM, Dr. B. Xi for CMFSL, Dr. Z. Li for FSCF-SSL, and Dr. J.Li for SCFormer, respectively.

REFERENCES

[1] Q. Tong, Y. Xue, and L. Zhang, "Progress in hyperspectral remote sensing science and technology in China over the past three decades," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 7, no. 1, pp. 70–91, Jan. 2014.

[2] S.-E. Qian, "Hyperspectral satellites, evolution, and development history," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 14, pp. 7032–7056, 2021.

[3] C. Wang et al., "A review of deep learning used in the hyperspectral image analysis for agriculture," *Artif. Intell. Rev.*, vol. 54, no. 7, pp. 5205–5253, May 2021.

[4] G. Lassalle, M. P. Ferreira, L. E. C. L. Rosa, R. D. M. Scafutto, and C. R. de S. Filho, "Advances in multi- and hyperspectral remote sensing of mangrove species: A synthesis and study case on airborne and multisource spaceborne imagery," *ISPRS J. Photogramm. Remote Sens.*, vol. 195, pp. 298–312, Jan. 2023.

[5] H. Shirmard, E. Farahbakhsh, R. D. Müller, and R. Chandra, "A review of machine learning in processing remote sensing data for mineral exploration," *Remote Sens. Environ.*, vol. 268, Jan. 2022, Art. no. 112750.

[6] Y.-N. Liu et al., "The advanced hyperspectral imager: Aboard China's GaoFen-5 satellite," *IEEE Geosci. Remote Sens. Mag.*, vol. 7, no. 4, pp. 23–32, Dec. 2019.

[7] S. Chabrillat et al., "The EnMAP spaceborne imaging spectroscopy mission: Initial scientific results two years after launch," *Remote Sens. Environ.*, vol. 315, Dec. 2024, Art. no. 114379.

[8] 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.

[9] L. Zhang and L. Zhang, "Artificial intelligence for remote sensing data analysis: A review of challenges and opportunities," *IEEE Geosci. Remote Sens. Mag. (replaces Newsletter)*, vol. 10, no. 2, pp. 270–294, Jun. 2022.

[10] S. Jia, S. Jiang, S. Zhang, M. Xu, and X. Jia, "Graph-in-graph convolutional network for hyperspectral image classification," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 35, no. 1, pp. 1157–1171, Jan. 2024.

[11] T. Lu, Y. Fang, W. Fu, K. Ding, and X. Kang, "Dual-stream class-adaptive network for semi-supervised hyperspectral image classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 62, 2024, Art. no. 5507511.

[12] C. Deng, Y. Xue, X. Liu, C. Li, and D. Tao, "Active transfer learning network: A unified deep joint spectral–spatial feature learning model for hyperspectral image classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 57, no. 3, pp. 1741–1754, Mar. 2019.

[13] R. Thoreau, V. Achard, L. Risser, B. Berthelot, and X. Briottet, "Active learning for hyperspectral image classification: A comparative review," *IEEE Geosci. Remote Sens. Mag.*, vol. 10, no. 3, pp. 256–278, May 2022.

[14] H. Lee, S. Eum, and H. Kwon, "Exploring cross-domain pretrained model for hyperspectral image classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 60, 2022, Art. no. 5526812.

[15] J. Zhang, W. Li, W. Sun, Y. Zhang, and R. Tao, "Locality robust domain adaptation for cross-scene hyperspectral image classification," *Expert Syst. Appl.*, vol. 238, Mar. 2024, Art. no. 121822.

[16] Z. Li et al., "Few-shot hyperspectral image classification with self-supervised learning," *IEEE Trans. Geosci. Remote Sens.*, vol. 61, 2023, Art. no. 5517917.

[17] Z. Li, Z. Xue, Q. Xu, L. Zhang, T. Zhu, and M. Zhang, "SPFormer: Self-pooling transformer for few-shot hyperspectral image classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 62, 2024, Art. no. 5502019.

[18] 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.

[19] 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.

[20] J. Zeng, Z. Xue, L. Zhang, Q. Lan, and M. Zhang, "Multistage relation network with dual-metric for few-shot hyperspectral image classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 61, 2023, Art. no. 5510017.

[21] J. Li, Z. Zhang, R. Song, Y. Li, and Q. Du, "SCFormer: Spectral coordinate transformer for cross-domain few-shot hyperspectral image classification," *IEEE Trans. Image Process.*, vol. 33, pp. 840–855, 2024.

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

[23] L. Zhang, L. Zhang, and B. Du, "Deep learning for remote sensing data: A technical tutorial on the state of the art," *IEEE Geosci. Remote Sens. Mag.*, vol. 4, no. 2, pp. 22–40, Jun. 2016.

[24] J. Quiñonero-Candela, M. Sugiyama, and A. Schwaighofer, *Dataset Shift in Machine Learning*. Cambridge, MA, USA: MIT Press, 2022.

[25] J. Gou, B. Yu, S. J. Maybank, and D. Tao, "Knowledge distillation: A survey," *Int. J. Comput. Vis.*, vol. 129, no. 6, pp. 1789–1819, Mar. 2021.

[26] G. Hinton, O. Vinyals, and J. Dean, "Distilling the knowledge in a neural network," 2015, *arXiv:1503.02531*.

[27] A. Romero, N. Ballas, S. E. Kahou, A. Chassang, C. Gatta, and Y. Bengio, "FitNets: Hints for thin deep nets," in *Proc. Int. Conf. Learn. Represent. (ICLR)*, Dec. 2015, pp. 1–11.

[28] S. Zagoruyko and N. Komodakis, "Paying more attention to attention: Improving the performance of convolutional neural networks via attention transfer," in *Proc. Int. Conf. Learn. Represent. (ICLR)*, Jan. 2016, pp. 1–16.

[29] H. Park and B. Ham, "Relation network for person re-identification," in *Proc. AAAI Conf. Artif. Intell.*, vol. 34, no. 7, 2020, pp. 11839–11847.

[30] C.-C. Hsu, C.-C. Ni, C.-M. Lee, and L.-W. Kang, "CSAKD: Knowledge distillation with cross self-attention for hyperspectral and multispectral image fusion," 2024, *arXiv:2406.19666*.

[31] H. Qin, T. Xu, P. Liu, J. Xu, and J. Li, "DMSSN: Distilled mixed spectral–spatial network for hyperspectral salient object detection," *IEEE Trans. Geosci. Remote Sens.*, vol. 62, 2024, Art. no. 5512618.

[32] C. Sun, X. Wang, Z. Liu, Y. Wan, L. Zhang, and Y. Zhong, "SiamOHOT: A lightweight dual Siamese network for onboard hyperspectral object tracking via joint spatial–spectral knowledge distillation," *IEEE Trans. Geosci. Remote Sens.*, vol. 61, 2023, Art. no. 5521112.

[33] W. Xie, Z. Zhang, L. Jiao, and J. Wang, "Decoupled knowledge distillation via spatial feature blurring for hyperspectral image classification," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 17, pp. 8938–8955, 2024.

[34] H. Wu, Z. Xue, S. Zhou, and H. Su, "Beyond spectral shift mitigation: Knowledge swap net for cross-domain few-shot hyperspectral image classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 62, 2024, Art. no. 5528218.

[35] C. Yu et al., "Distillation-constrained prototype representation network for hyperspectral image incremental classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 62, 2024, Art. no. 5507414.

[36] L. Zhang, J. Song, A. Gao, J. Chen, C. Bao, and K. Ma, "Be your own teacher: Improve the performance of convolutional neural networks via self distillation," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 3713–3722.

[37] R. Shang et al., "Hyperspectral image classification based on pyramid coordinate attention and weighted self-distillation," *IEEE Trans. Geosci. Remote Sens.*, vol. 60, 2022, Art. no. 5544316.

[38] J. Wang, S. Guo, Z. Hua, R. Huang, J. Hu, and M. Gong, "CL-CaGAN: Capsule differential adversarial continual learning for cross-domain hyperspectral anomaly detection," *IEEE Trans. Geosci. Remote Sens.*, vol. 62, 2024, Art. no. 5517315.

[39] B. Qin, S. Feng, C. Zhao, W. Li, R. Tao, and W. Xiang, "Cross-domain few-shot learning based on feature disentanglement for hyperspectral image classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 62, 2024, Art. no. 5514215.

[40] J. Yue, L. Fang, H. Rahmani, and P. Ghamisi, "Self-supervised learning with adaptive distillation for hyperspectral image classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 60, 2022, Art. no. 5501813.

[41] 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, Feb. 2018.

[42] L. Sun, G. Zhao, Y. Zheng, and Z. Wu, "Spectral–spatial feature tokenization transformer for hyperspectral image classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 60, 2022, Art. no. 5522214.
[43] H. Liu, W. Li, X.-G. Xia, M. Zhang, and R. Tao, "Multiarea target attention for hyperspectral image classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 61, 2023, Art. no. 5524916.
[44] R. Xu, X.-M. Dong, W. Li, J. Peng, W. Sun, and Y. Xu, "DBCT-Net: Double branch convolution-transformer network for hyperspectral image classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 62, 2024, Art. no. 5509915.
[45] W. Zhou, S.-I. Kamata, H. Wang, M. S. Wong, and H. Hou, "Mamba-in-mamba: Centralized mamba-cross-scan in tokenized mamba model for hyperspectral image classification," *Neurocomputing*, vol. 613, Jan. 2025, Art. no. 128751.

Hao Wu received the B.E. degree in geodesy and geomatics engineering and the M.E. degree in photogrammetry and remote sensing from the School of Earth Sciences and Engineering, Hohai University, Nanjing, China, in 2018 and 2022, respectively, where he is currently pursuing the Ph.D. degree in surveying and mapping.

His research concerns remote sensing image processing, hyperspectral image analysis, and transfer learning.

Zhaohui Xue (Member, IEEE) received the B.S. degree in geomatics engineering from Shandong Agricultural University, Tai'an, China, in 2009, the M.E. degree in remote sensing from China University of Mining and Technology, Beijing, China, in 2012, and the Ph.D. degree in cartography and geographic information system from Nanjing University, Nanjing, China, in 2015.

He is currently a Full Professor (Ph.D. Supervisor) with the College of Geography and Remote Sensing, Hohai University, Nanjing. He has authored more than 70 scientific articles including more than 40 Science Citation Index (SCI) articles. His research interests include hyperspectral image classification, time-series image analysis, pattern recognition, and machine learning.

Dr. Xue was a recipient of the National Scholarship for Doctoral Graduate Students granted by the Ministry of Education of the People's Republic of China in 2014. He was awarded the Best Reviewer for the IEEE Geoscience and Remote Sensing Society. He was an Editorial Board Member of *National Remote Sensing Bulletin* from 2020 to 2024. He has been a reviewer for more than ten famous remote sensing journals including IEEE TRANSACTIONS ON GEOSCIENCE AND REMOTE SENSING, *Remote Sensing of Environment*, and *ISPRS Journal of Photogrammetry and Remote Sensing*.

Shaoguang Zhou received the B.S. degree in management engineering in industrial enterprises from Zhenjiang Shipbuilding Institute, Zhenjiang, China, in 1988, the M.S. degree in optical and precision instruments from Shanghai Institute of Mechanical Engineering, Shanghai, China, in 1991, and the Ph.D. degree in optics from Xi'an Institute of Optics and Fine Mechanics, Chinese Academy of Sciences, Beijing, China, in 1997.

From 1998 to 1999, he was a Post-Doctoral Fellow with the School of Power Engineering, Nanjing University of Science and Technology, Nanjing, China. Since 2000, he has been an Associate Professor with the School of Earth Sciences and Engineering, Hohai University, Nanjing. From 2014 to 2015, he was a Visiting Scholar with the School of Science, Engineering and Information Technology, Federation University Australia, Gippsland, Churchill, VIC, Australia. He has authored or co-authored approximately 50 journal articles. His research interests include remote sensing image segmentation and classification, ground feature detection and recognition, and image matching and computer vision.

Hongjun Su (Senior Member, IEEE) received the Ph.D. degree in cartography and geography information system from the Key Laboratory of Virtual Geographic Environment (Ministry of Education), Nanjing Normal University, Nanjing, China, in 2011.

He is currently a Full Professor with the College of Geography and Remote Sensing, Hohai University, Nanjing. His main research interests include hyperspectral remote sensing dimensionality reduction, classification, and spectral unmixing.

Dr. Su received the 2016 Best Reviewer Award from the IEEE Geoscience and Remote Sensing Society. He is an Associate Editor of IEEE JOURNAL OF SELECTED TOPICS IN APPLIED EARTH OBSERVATIONS AND REMOTE SENSING.