

Article

Uncertainty-Guided Multi-Center Prototype Alignment for Cross-Domain Few-Shot Hyperspectral Image Classification

Qinzheng Wang, Menglei Li , Shiping Du * and Li Wang *

School of Computer Science and Software Engineering, University of Science and Technology Liaoning, Anshan 114051, China; wqz0351@stu.ustl.edu.cn (Q.W.); 312571@ustl.edu.cn (M.L.)

* Correspondence: dsp001@ustl.edu.cn (S.D.); wangli9966@ustl.edu.cn (L.W.)

Abstract

Accurate hyperspectral image analysis plays a critical role in environmental monitoring, precision agriculture, and urban mapping, yet acquiring large-scale annotated datasets for newly emerging scenes and sensors remains challenging. Cross-domain few-shot hyperspectral image classification addresses this bottleneck by transferring knowledge from a labeled source domain to a sparsely annotated target domain. Existing prototype-based approaches commonly use a single-mean prototype per class, which is often inadequate for target classes with spectral-spatial heterogeneity, intra-class dispersion, and unstable episode-level statistics. Consequently, the resulting alignment reference may fail to accurately characterize class structure, thereby exacerbating negative transfer under domain shift. To address this issue, we propose a plug-and-play prototype-stability module that combines Uncertainty-Guided Clustered Alignment (UGCA) with Center Regularization. UGCA identifies hard classes using a class-level dispersion proxy and dynamically constructs multi-center prototypes to better capture intra-class heterogeneity beyond the single-mean assumption. Meanwhile, Center Regularization adds a lightweight compactness constraint on query embeddings to reduce prototype drift under sparse supervision. Experiments on three cross-domain tasks demonstrate the effectiveness of the proposed method. Compared with the reproduced MLPA baseline over 10 independent runs, the proposed method improves the mean OA from $69.21 \pm 2.71\%$, $81.90 \pm 3.45\%$, and $76.92 \pm 1.10\%$ to $71.24 \pm 3.39\%$, $83.73 \pm 3.89\%$, and $78.24 \pm 1.55\%$ on the Indian Pines (IP), University of Pavia (UP), and Houston (HT) target domains, respectively. Furthermore, cross-framework insertion into Gia-CFSL further verifies that the module is host-agnostic across prototype-driven CD-FSL frameworks and improves prototype-based hyperspectral image analysis without changing the inference pipeline. These results indicate that improving prototype quality is a critical and complementary dimension for robust cross-domain few-shot hyperspectral classification under sparse supervision.



Academic Editor: Silvia Liberata Ullo

Received: 1 June 2026

Revised: 28 June 2026

Accepted: 10 July 2026

Published: 14 July 2026

Copyright: © 2026 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 \(CC BY\)](https://creativecommons.org/licenses/by/4.0/) license.

Keywords: hyperspectral image classification; cross-domain few-shot learning; prototype alignment; uncertainty proxy; prototype stabilization

1. Introduction

Hyperspectral imaging captures hundreds of contiguous narrow spectral bands for each pixel, forming a high-dimensional spectral-spatial data cube that preserves both the geometric structure of a scene and the spectral signatures of its constituent materials [1,2]. Owing to this rich joint representation, hyperspectral imagery has become a key data source

for Earth observation tasks such as precision agriculture, mineral exploration, environmental monitoring, and urban land-cover mapping. Pixel-level classification, which assigns a semantic label to each hyperspectral pixel based on its spectral signature and spatial context, underpins most of these applications. However, the high dimensionality of the spectra, the strong correlation between adjacent bands, the large intra-class spectral variability caused by illumination and sensor conditions, and the scarcity of expert annotations together make hyperspectral image classification a long-standing challenge.

Before the advent of deep learning, the field was dominated by a two-stage pipeline in which hand-crafted spectral or spectral–spatial features were extracted and then fed into shallow classifiers such as support vector machines (SVMs) [3]. Such pipelines are efficient and interpretable, and they remain useful baselines under extremely limited training data, but their representational capacity is constrained by manual feature engineering, which often lacks robustness across sensors and scenes. Consequently, they generalize poorly when intra-class spectral variability is large and inter-class spectral overlap is severe, motivating the search for hierarchical, data-driven feature learners.

Deep learning has reshaped the field by enabling end-to-end learning of hierarchical spectral–spatial representations directly from raw cubes [4]. Chen et al. [5] were among the first to introduce deep architectures by combining principal component analysis with stacked autoencoders, and Hu et al. [6] showed that even a compact one-dimensional convolutional neural network (CNN) operating on spectral signatures can outperform classical classifiers, although purely spectral models ignore the spatial context that is essential for hyperspectral discrimination. Later work therefore moved toward joint spectral–spatial encoders, including the spectral–spatial residual network (SSRN) of Zhong et al. [7] and the deeper contextual CNN of Lee and Kwon [8]. More recently, Transformer-style models have revisited the problem by capturing long-range dependencies, as in the spectral–spatial Transformer of Sun et al. [9] and the SpectralFormer of Hong et al. [10]. Despite their representational power, these models are heavily parameterized and require abundant, high-quality annotations. In practice, labeling hyperspectral pixels demands domain expertise and field validation, so the label budget is typically very small, leaving purely supervised models prone to overfitting on newly acquired scenes.

To ease this annotation bottleneck, cross-domain learning has been studied as a way to transfer knowledge from a label-rich source dataset to a label-scarce target dataset, broadly along two lines. The first follows a pre-train-and-finetune strategy: Jiang et al. [11], for example, pre-train a 3D separable residual network on an auxiliary source dataset and transfer the parameters to small target datasets, which reduces overfitting but remains sensitive to the spectral gap between sensors. The second adopts an explicit domain adaptation formulation that minimizes the divergence between feature distributions; representative examples include the discriminative transfer joint matching of Peng et al. [12], the adversarial adaptation network of Huang et al. [13], and the unseen-category extension of Pan et al. [14]. These methods narrow the source–target gap when the two label spaces are well aligned, but they do not address the case in which both the label space and the label budget of the target domain differ substantially from the source.

Beyond hyperspectral classification, Dong et al. [15] studied cross-scene generalization in universal moving object segmentation by learning temporal pixel distributions and refining spatial correlations through a Stochastic Bayesian Refinement network. This distribution-level modeling strategy is conceptually related to our motivation of replacing fragile point prototypes with more stable distribution-aware prototype representations under domain shift.

A complementary line tackles the small-label problem through few-shot learning (FSL), which learns transferable embeddings from a large meta-training set so that new categories

can be recognized from only a handful of examples; prototypical networks [16], relation networks [17], and meta-transfer learning [18] are representative formulations. Several works adapt this paradigm to hyperspectral imagery, such as the deep few-shot framework of Liu et al. [19] and the learnable relation module of Gao et al. [20]. More recent studies in the broader machine learning community have extended few-shot learning toward contemporary hybrid paradigms and cross-domain visual recognition [21,22]. These works suggest that few-shot generalization is increasingly viewed as robust representation transfer under distribution shift, which is closely aligned with the motivation of hyperspectral CD-FSL. These methods reduce the per-scene annotation burden, yet their meta-training and meta-testing are usually performed on the same dataset, so they do not handle the additional distribution gap that arises when source and target scenes are acquired by different sensors or over different regions.

Cross-domain few-shot learning (CD-FSL) integrates domain adaptation with few-shot learning and has therefore emerged as a promising paradigm for label-scarce hyperspectral image classification. Early studies focused on domain-adversarial optimization; for instance, Li et al. [23] proposed DCFSL, which jointly optimizes episodic meta-learning and conditional adversarial alignment. Subsequent works increasingly emphasized relational and structural information transfer, including the graph information aggregation of Gia-CFSL [24] and the cross-domain graph structures of Xu et al. [25]. Attention- and Transformer-based architectures later gained traction for modeling long-range spectral-spatial dependencies, as in the convolution-Transformer hybrid of Peng et al. [26], the dual-branch domain-gating framework of Wang et al. [27], the spectral-spatial domain attention of Zhang et al. [28], and the Spectral Coordinate Transformer (SCFormer) of Li et al. [29], which embeds spectral coordinates through rotary position embedding to improve cross-domain generalization. In parallel, another line explored compact and prototype-driven representations, such as the higher-order tensor factorization of Tang et al. [30] and Multi-Level Prototype Alignment (MLPA) [31], which performs adversarial prototype alignment at several semantic depths within a shared 3D residual encoder. Owing to its strong prototype-based adaptation capability, MLPA is adopted in this work as the primary prototype-driven CD-FSL baseline.

Despite this progress, prototype-driven CD-FSL frameworks such as MLPA [31] rely on a unified representation in which the support samples of each class are compressed into a single-mean prototype for deep adversarial alignment. Although prototype diversity and mixture-based prototypes have been explored in the broader few-shot learning literature, such as infinite mixture prototypes that represent each class with multiple latent clusters [32], their role in prototype-driven cross-domain few-shot hyperspectral classification remains insufficiently investigated. In particular, existing hyperspectral CD-FSL prototype-alignment methods still commonly use a single-mean prototype as the alignment reference, which rests on two strong assumptions. First, it assumes that target-class embeddings are compact and approximately uni-modal, so that a single mean can characterize the class structure. In practice, hyperspectral classes often exhibit pronounced intra-class dispersion and even multi-modal sub-structures under spectral variability, spatial heterogeneity, illumination changes, and sample sparsity [33], so a single-mean prototype may provide an inaccurate alignment reference for complex classes. Second, it assumes that the prototype remains stable across episodes, which rarely holds when only a few labeled target samples are available per class; the prototype then becomes highly sensitive to outliers and short-term encoder fluctuations, causing decision boundaries to drift during training. A third issue amplifies both: treating all classes equally during alignment prevents the adversarial pressure from being steered toward the classes that actually suffer from prototype mismatch. These observations point to three challenges: (i) insufficient

representation of complex target classes, since a single-mean prototype cannot capture multi-modal or highly dispersed class topologies; (ii) prototype instability under sparse target supervision, since the support-derived prototype is dominated by sampling noise and short-term encoder drift; and (iii) the lack of class-adaptive alignment, since uniform per-class treatment wastes alignment capacity on easy classes while leaving hard classes under-aligned. Taken together, they suggest that the quality and stability of the alignment reference itself, rather than ever more elaborate feature extractors or alignment losses, is a critical but largely overlooked direction for robust cross-domain few-shot hyperspectral image classification.

To address these challenges, we propose a prototype-stabilization enhancement, instantiated on MLPA [31], that plugs into the deep alignment stage of prototype-driven CD-FSL frameworks and combines two complementary modules. The Uncertainty-Guided Clustered Alignment (UGCA) module maintains dual-domain class-wise memory banks, derives a class-level uncertainty proxy from intra-class dispersion, and applies K-means to build multi-center prototypes only for hard classes selected by a quantile-plus-Top-M criterion, while easy classes retain single-center alignment. In this paper, the term uncertainty does not refer to Bayesian parameter uncertainty or predictive entropy; instead, it denotes a dispersion-based prototype-stability proxy computed from intra-class feature variance in the embedding space. The Center Regularization strategy introduces learnable class centers and pulls source- and target-domain query embeddings toward the corresponding centers through Center Loss, providing a global compactness signal that stabilizes prototype estimation under sparse supervision. Acting on complementary aspects—reference expressiveness and embedding stability—the two modules jointly mitigate the negative transfer caused by coarse or drifting prototypes. The proposed method serves as a plug-and-play enhancement for prototype-driven CD-FSL networks, as it operates only during training and leaves the metric-based inference pipeline unchanged. It should be noted that the framework is specifically designed for methods that rely on explicit prototype representations and prototype alignment, rather than as a generic module applicable to all cross-domain few-shot learning architectures.

Overall, the main contributions of this paper can be summarized as follows:

1. A prototype-stability enhancement framework is proposed for prototype-driven cross-domain few-shot hyperspectral image classification. The framework explicitly addresses both prototype expressiveness and prototype stability, two issues that are largely overlooked in existing prototype-alignment methods.
2. An Uncertainty-Guided Clustered Alignment (UGCA) module is developed to identify structurally complex classes and construct adaptive multi-center prototypes, thereby providing more representative alignment references than conventional single-mean prototypes.
3. A training-stage plug-and-play integration strategy is designed to incorporate UGCA and Center Regularization into prototype-driven CD-FSL frameworks without modifying the metric-based inference pipeline, thereby improving prototype expressiveness and stability while preserving the original testing protocol.

The remainder of this paper is organized as follows. Section 2 formulates the cross-domain few-shot hyperspectral image classification task, reviews the episodic metric-learning paradigm, and summarizes the MLPA baseline. Section 3 presents the proposed prototype-stability enhancement, including the overall architecture, the UGCA module, the Center Regularization strategy, the training procedure, and the overall objective. Section 4 presents experiments on the Chikusei-to-IP/UP/HT cross-domain tasks, covering comparisons with the MLPA baseline and representative methods, component and alignment-level ablation studies, hard-class triggering analysis, plug-and-play validation

on Gia-CFSL, parameter sensitivity analysis, and complexity analysis. Section 5 discusses the effectiveness of UGCA and Center Regularization, as well as the training setup and current limitations. Finally, Section 6 concludes the paper and provides an outlook on future research directions.

2. Problem Definition and Baseline Framework

In this section, we first formulate the cross-domain few-shot hyperspectral image classification task. We then introduce the episodic metric-learning paradigm. Finally, we summarize the core architecture of the baseline framework (MLPA) upon which our method is developed. Clarifying this baseline is necessary, as our proposed method is specifically designed to enhance prototype stability and representational capacity within this pipeline.

2.1. Problem Definition

We consider a cross-domain few-shot hyperspectral image classification setting involving a labeled source domain and a sparsely labeled target domain. Let the source-domain dataset be denoted by $D_s = \{(x_i^s, y_i^s)\}_{i=1}^{N_s}$, where x_i^s denotes a hyperspectral sample and y_i^s is its corresponding class label. Similarly, let the target-domain dataset be denoted by $D_t = \{(x_j^t, y_j^t)\}_{j=1}^{N_t}$.

In practical cross-domain scenarios, the source domain contains relatively abundant labeled samples to provide transferable prior knowledge. By contrast, target-domain labels are extremely scarce. Following the standard few-shot transfer protocol, the target-domain dataset is divided into a few-labeled subset D_f and an unlabeled testing subset D_u , such that $D_f \cup D_u = D_t$ and $D_f \cap D_u = \emptyset$. Here, D_f contains only K_s labeled samples per class for target-domain adaptation and classifier fitting, while D_u is used for final evaluation.

The objective is to exploit the sufficient supervision from D_s together with the extremely limited labeled samples from D_f , so that the learned model can generalize effectively to D_u . This task is challenging as it requires the model to simultaneously cope with severe label scarcity and substantial domain shift induced by different sensors, observation regions, or environmental factors.

2.2. Episodic Few-Shot Classification

To learn transferable representations under scarce supervision, the baseline framework adopts an episodic few-shot learning paradigm, which repeatedly constructs small classification tasks to simulate the few-shot evaluation scenario.

For a given episode, C classes are randomly sampled. For each class, K_s labeled samples form the support set S , and N additional samples form the query set Q . Taking the source domain as an example, we have $S_s = \{(x_i, y_i)\}_{i=1}^{C \times K_s}$ and $Q_s = \{(x_j, y_j)\}_{j=1}^{C \times N}$. Let $F_\phi(\cdot)$ denote the feature extraction network parameterized by ϕ . The prototype of class k is obtained by averaging the support embeddings of that class:

$$\mathbf{z}_{\text{mean}}(k) = \frac{1}{K_s} \sum_{(x_i, y_i) \in S_s^{(k)}} F_\phi(x_i) \quad (1)$$

where $S_s^{(k)}$ denotes the set of support samples belonging to class k . For a query sample $x_j \in Q_s$, its predicted probability of belonging to class k is computed using the softmax over Euclidean distances $d(\cdot, \cdot)$ to the prototypes:

$$P_\phi(y_j = k \mid x_j) = \frac{\exp(-d(F_\phi(x_j), \mathbf{z}_{\text{mean}}(k)))}{\sum_{k'=1}^C \exp(-d(F_\phi(x_j), \mathbf{z}_{\text{mean}}(k')))} \quad (2)$$

The source-domain few-shot classification loss is then formulated as the negative log-likelihood of the true classes:

$$L_{fsl}^s = \mathbb{E}_{\mathcal{S}_s, Q_s} \left[- \sum_{(x_j, y_j) \in Q_s} \log P_\phi(y_j = k \mid x_j) \right] \quad (3)$$

A structurally identical loss, denoted as L_{fsl}^t , can be derived for the target-domain episodes. This metric-based decision rule is highly robust for target domains where training a fully parameterized classifier from scratch is prone to overfitting.

2.3. Baseline Framework

The proposed method builds upon the Multi-Level Prototype Alignment (MLPA) framework [31], which integrates spectral dimension alignment, shared feature extraction, and multi-level prototype alignment into a unified architecture.

Mapping Layer and 3D Residual Feature Extractor. To handle the discrepancy in spectral dimensionality between source and target datasets, MLPA utilizes an independent 1×1 convolutional mapping layer for each domain to project hyperspectral cubes into a unified spectral space. Subsequently, both domains share a 3D residual feature extractor (comprising residual blocks and max-pooling layers) to jointly capture local spatial context and spectral continuity.

Multi-Level Prototype Alignment. To mitigate domain shift, MLPA aligns the source and target domains not only at the final embedding layer but also at intermediate feature levels. Instead of directly aligning all sample features, MLPA calculates the class mean prototypes for both domains at each level, and feeds them into a domain discriminator. At the n -th alignment level, the prototype adversarial objective is formulated as

$$\min_{F_n} \max_{D_n} L_{adv}^{(n)} = \mathbb{E}_{c \sim \mathcal{C}_s} \log [D_n(\mu_c^s)] + \mathbb{E}_{c \sim \mathcal{C}_t} \log [1 - D_n(\mu_c^t)] \quad (4)$$

where $\mathcal{C}_s$ and $\mathcal{C}_t$ denote the source- and target-domain class sets, μ_c^s and μ_c^t denote the source- and target-domain class prototypes for class c extracted at the n -th level, and $D_n(\cdot)$ is the domain discriminator. By summing the alignment losses across three hierarchical levels ($L_{adv} = \sum_{n=1}^3 L_{adv}^{(n)}$), the model reduces domain discrepancy progressively from shallow to deep layers.

Alternating Training and Inference. During training, MLPA adopts a cross-domain alternating fine-tuning strategy. In each iteration, it alternately optimizes the source-domain task (minimizing $L_{fsl}^s + \lambda L_{adv}$) and the target-domain task (minimizing $L_{fsl}^t + \lambda L_{adv}$), effectively preserving discriminative source knowledge while adapting to the target distribution. During inference, the unlabeled target-domain samples are mapped into the embedding space and classified based on their Euclidean proximity to the prototypes constructed from the few labeled target samples [31].

Limitations of the Baseline. While MLPA improves cross-domain generalization through multi-level alignment, it relies fundamentally on single-mean prototypes. When target-domain classes exhibit strong intra-class dispersion or complex multi-modal structures, such a simplistic prototype fails to faithfully characterize the true category distribution. Moreover, under extremely sparse target-domain supervision, these mean prototypes are highly vulnerable to local noise and outliers. These observations directly motivate our proposed Uncertainty-Guided Clustered Alignment and Center Regularization strategies, as detailed in the next section.

3. Proposed Method

Built upon the prototype-driven cross-domain few-shot learning framework of MLPA, our method aims to enhance prototype reliability under severe domain shift and extremely limited target-domain supervision. Instead of redesigning the entire feature extractor or replacing the baseline training protocol, we focus on improving the quality of the prototypes themselves. To this end, two complementary modules are introduced: (1) an Uncertainty-Guided Clustered Alignment (UGCA) module, which enhances the representation of difficult classes by replacing unreliable single-mean prototypes with clustered multi-center alignment units at the deep alignment stage; (2) a Center Regularization strategy, which explicitly constrains intra-class compactness in the embedding space to mitigate prototype drift under scarce supervision. These two modules operate on different aspects of the prototype-learning process and jointly improve the robustness of cross-domain few-shot classification.

3.1. Overall Architecture

The overall framework follows the multi-level prototype-based cross-domain pipeline of MLPA. Specifically, source-domain and target-domain hyperspectral cubes are first projected into a unified spectral space through domain-specific mapping layers and then passed into a shared 3D residual feature extractor. While the baseline performs Multi-Level Prototype Alignment across all depths, high-level semantic distributions tend to be more complex and more susceptible to negative transfer than low-level textural features [24,31]. Therefore, we preserve the baseline protocol in the shallow layers but augment the deep embedding level with our prototype-stabilization modules.

The first module, UGCA, is applied to the deep prototype construction and alignment stage. Its purpose is to detect difficult classes whose intra-class feature distributions are unstable or structurally complex, and then replace the conventional single-center representation with clustered multi-center prototypes. The second module, Center Regularization, is introduced simultaneously at the deep embedding stage. By imposing an explicit compactness constraint, it encourages samples from the same class to aggregate tightly around learnable class centers.

The overall training architecture is illustrated in Figure 1. These two modules play complementary roles. UGCA addresses the expressiveness issue, recognizing that a single mean is often insufficient to characterize a class with a complex internal topology. Center Regularization addresses the stability issue, ensuring that prototype estimation from very few labeled samples remains robust against outliers and local noise.

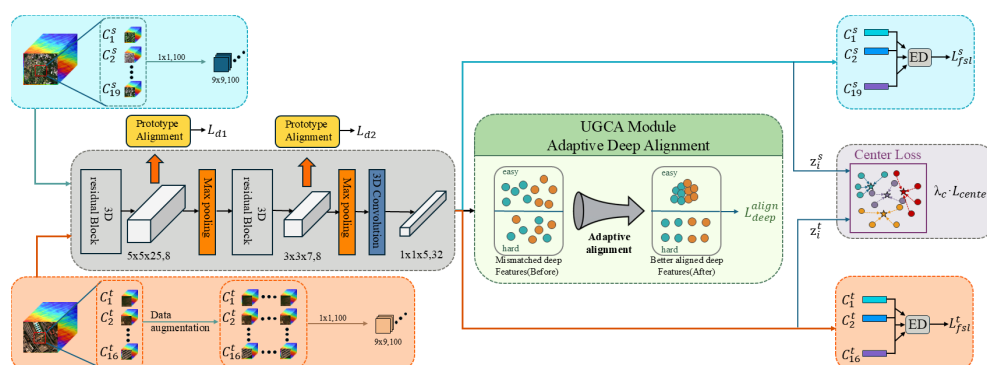


Figure 1. Overall training architecture. Source-domain and target-domain inputs are processed by a shared feature extractor. Prototype alignment is applied to shallow features, and deep features are refined by the UGCA module for adaptive alignment. The network is optimized under few-shot classification objectives combined with the center loss.

3.2. Uncertainty-Guided Clustered Alignment (UGCA)

The design of UGCA is motivated by a critical limitation of existing prototype-driven methods: the assumption that each class can be sufficiently represented by a single-mean prototype. In cross-domain hyperspectral scenarios, factors such as spatial heterogeneity and the spectral variability of identical materials under varying conditions cause samples from the same class to exhibit substantial dispersion, often forming multi-modal sub-clusters. Under such conditions, a single-mean prototype becomes an overly coarse approximation, weakening domain alignment and aggravating negative transfer. UGCA addresses this issue through an uncertainty-guided mechanism that identifies difficult classes and applies fine-grained multi-center alignment to them at the deep stage. The overall workflow of the UGCA module is illustrated in Figure 2.

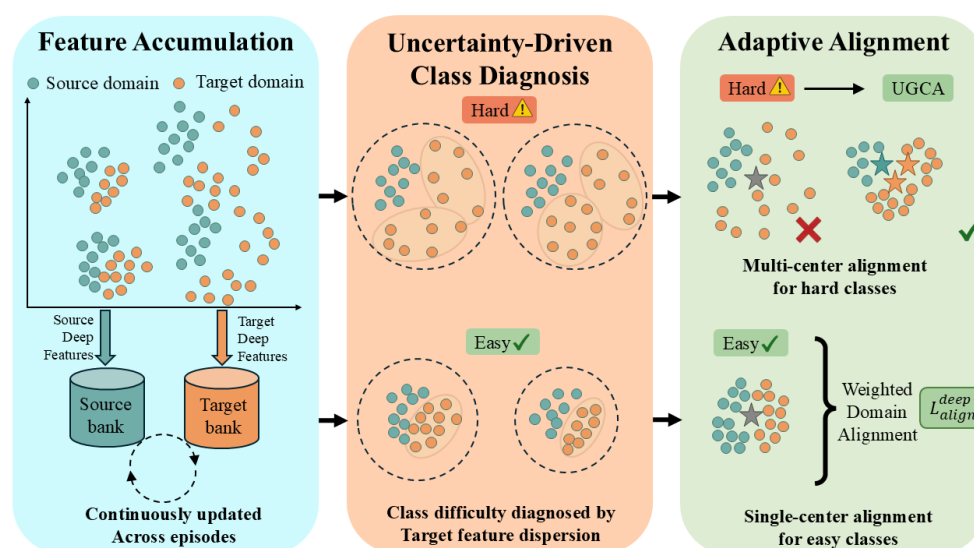


Figure 2. Illustration of the UGCA mechanism. Dual-domain memory banks accumulate class features across episodes; a class-level uncertainty proxy identifies hard classes; K-means clustering constructs multi-center prototypes for fine-grained deep alignment. Stars of different colors denote the feature centers of the corresponding classes.

3.2.1. Dual Memory Bank Construction

A major challenge in few-shot learning is the instability of class statistics when estimated solely from the limited samples of a single episode. To achieve reliable cross-domain distribution modeling, we maintain a pair of parallel class-wise memory banks to dynamically accumulate feature embeddings for both the source and target domains.

For a domain $d \in \{s, t\}$ (source or target) and each class c , the memory bank is denoted as

$$B_d(c) = \{\mathbf{z}_j\}_{j=1}^{|B_d(c)|}, \quad d \in \{s, t\} \quad (5)$$

where $\mathbf{z}_j$ represents the deep embedding of a sample belonging to class c in domain d . In what follows, $B_s(c)$ and $B_t(c)$ denote the source- and target-domain memory banks specifically; the generic notation $B_d(c)$ is used when a statement applies to both domains. During training, newly observed embeddings from episodic tasks are continuously enqueued into the corresponding banks using a first-in-first-out (FIFO) mechanism. This provides a rich and temporally smoothed feature pool, alleviating the dependence on the sparse support samples of the current episode.

In the current implementation, the maximum cache size is set to $M = 256$ for each class in each domain.

3.2.2. Valid Update Protocol for the Target-Domain Memory Bank

To avoid any ambiguity regarding data leakage, the target-domain memory bank is updated only with samples from the labeled target-training subset D_f and its augmentations. Neither the target testing subset D_u nor any labels associated with D_u are used in memory-bank updates, hard-class selection, clustered prototype construction, or parameter optimization. Let Q_i^{train} denote the target-domain episodic query samples drawn from D_f , and let D_f^{aug} denote the augmented target-domain training samples. The target-domain memory bank for class c is updated as

$$B_i^{(e+1)}(c) = \text{FIFO}\left(B_i^{(e)}(c) \cup \{F_\phi(x) \mid (x, y = c) \in Q_i^{train} \cup D_f^{aug}\}\right) \quad (6)$$

The above update rule establishes the target-domain memory bank as a temporal feature buffer that accumulates embeddings of labeled target-training samples and their augmentations throughout training. This design is strictly confined to D_f , ensuring that no information from D_u participates in memory-bank maintenance, hard-class selection, clustered prototype construction, or parameter optimization.

This formulation is intentional. Rather than attempting to approximate the full target-domain population distribution, the memory bank reflects the organization of each class in the model's current embedding space. When a class exhibits persistently large intra-class dispersion across episodes, it cannot be stably characterized by a single prototype—regardless of whether this dispersion originates from intrinsic class complexity, augmentation-induced variability, or the temporal evolution of the encoder. UGCA is therefore designed to intervene directly at the representation level, which is where the cross-domain alignment objective ultimately operates. Accordingly, the uncertainty proxy should be interpreted as a prototype-stability indicator in the current embedding space rather than as an estimator of the true target-domain data distribution.

3.2.3. Class-Level Uncertainty-Proxy Estimation

Relying on the target-domain memory bank $B_i(c)$, we compute a class-level uncertainty proxy, namely a dispersion-based prototype-stability indicator, from the intra-class feature variance of each target class. Let the target-domain class mean be $\mu_c^t = \frac{1}{|B_i(c)|} \sum_{\mathbf{z} \in B_i(c)} \mathbf{z}$. At training iteration τ , the target-domain class-wise uncertainty proxy is defined by its feature variance:

$$u_c^{t,(\tau)} = \frac{1}{|B_i(c)|} \sum_{\mathbf{z} \in B_i(c)} \|\mathbf{z} - \mu_c^t\|_2^2 \quad (7)$$

A larger $u_c^{t,(\tau)}$ indicates that the class is more scattered or structurally complex in the target-domain embedding space, implying that it cannot be adequately represented by a single-mean prototype.

To ensure temporal stability across iterations, an exponential moving average (EMA) is applied:

$$\bar{u}_c^{t,(\tau)} = \alpha \bar{u}_c^{t,(\tau-1)} + (1 - \alpha) u_c^{t,(\tau)} \quad (8)$$

where $\bar{u}_c^{t,(\tau)}$ is the EMA-smoothed uncertainty proxy of target class c at iteration τ , and $\alpha \in (0, 1)$ is the momentum coefficient. For brevity, in subsequent equations we abbreviate $\bar{u}_c^{t,(\tau)}$ as $\bar{u}_c$ when the target-domain and current-iteration meanings are clear from context.

Here, the proxy is not intended as a Bayesian uncertainty estimate. Rather, it serves as a representation-level indicator of class complexity derived from the labeled target-training cache. A larger value suggests that the class is more dispersed, less stable, or more likely to contain potential substructures in the current embedding space. It should be noted that the

proxy is not designed to distinguish between intrinsic class complexity and label-induced dispersion. Instead, it measures the overall degree of representation instability that may undermine single-prototype alignment.

3.2.4. Hard-Class Selection (Quantile Threshold + Top-M)

In this work, a class is considered hard if its feature embeddings exhibit persistently high intra-class dispersion across training episodes, indicating that a single-mean prototype cannot stably represent its internal structure. Such dispersion typically arises in spectrally similar vegetation types or heterogeneous urban materials whose sub-populations overlap in the embedding space. Conversely, a class is considered easy if its embeddings remain compact and well-clustered, for which a single prototype already provides a reliable alignment anchor.

Not all classes need multi-center modeling. Applying clustering to every class would increase the computational burden and may introduce unnecessary perturbations for structurally simple classes. Therefore, we first define a candidate set that contains only the classes whose source-domain and target-domain memory banks have accumulated sufficient samples:

$$\mathcal{C}_{cand} = \{c \mid |B_s(c)| \geq N_{min}^s, |B_t(c)| \geq N_{min}^t\} \quad (9)$$

We then compute a quantile threshold on the EMA-smoothed class-level uncertainty proxies within this candidate set:

$$\tau_q = \text{Quantile}_q(\{\bar{u}_c \mid c \in \mathcal{C}_{cand}\}) \quad (10)$$

Finally, the hard-class set is determined by selecting the Top-M classes among those whose uncertainty proxies exceed the quantile threshold:

$$\mathcal{C}_{hard} = \text{TopM}(\{\bar{u}_c \mid c \in \mathcal{C}_{cand}, \bar{u}_c > \tau_q\}) \quad (11)$$

This two-stage selection serves two purposes. First, it prevents premature hard-class activation when the memory banks are still too sparse. Second, it limits multi-center modeling to the classes that are most likely to benefit from it. In the current implementation, $N_{min}^s = 24$, $N_{min}^t = 24$, $q = 0.75$, and Top-M = 4.

3.2.5. Clustered Prototype Construction and Periodic Refresh

After the warm-up stage, the model periodically refreshes the clustered prototypes of the hard classes using K-means on the corresponding memory-bank features. Specifically, clustering is activated only when the training episode index reaches the warm-up threshold E_w , and the refresh is performed every T_{up} episodes. For each hard class $c \in \mathcal{C}_{hard}$, K-means is applied separately to the source-domain and target-domain memory-bank features to produce K cluster centers:

$$\{\mu_{c,1}^s, \dots, \mu_{c,K}^s\}, \quad \{\mu_{c,1}^t, \dots, \mu_{c,K}^t\} \quad (12)$$

If a class does not belong to $\mathcal{C}_{hard}$, or if the memory-bank contents do not satisfy the minimum sample requirement, the model continues to use the conventional single-mean prototype for that class. In the current implementation, $E_w = 200$, $T_{up} = 10$, and $K = 2$. In the current implementation, K is fixed to 2. This choice reflects a compromise between representational flexibility and cluster reliability under sparse supervision. The objective of UGCA is to relax the single-prototype assumption rather than to recover the exact class topology.

3.2.6. Clustered Alignment Units and Deep Adversarial Objective

For a hard class c , each of its K clustered centers is treated as an independent alignment unit and assigned a uniform weight of $1/K$. This design ensures that splitting a class into K sub-centers does not inflate its total adversarial contribution relative to non-hard classes, while still allowing each sub-region of the feature space to receive dedicated alignment pressure. Let $\mathcal{U}_c$ denote the deep alignment units and let ω denote the corresponding domain-alignment weight. Then

$$(\mathcal{U}_c, \omega) = \begin{cases} (\{\mu_{c,1}, \dots, \mu_{c,K}\}, 1/K), & c \in \mathcal{C}_{hard}, \\ (\{\mu_c\}, 1), & c \notin \mathcal{C}_{hard} \end{cases} \quad (13)$$

These alignment units are then fed into the deep domain discriminator. The resulting deep adversarial objective can be written as

$$\mathcal{L}_{adv}^{deep} = \sum_u \omega_u \text{BCE}(D(h_u), d_u) \quad (14)$$

where h_u denotes a deep alignment unit, d_u denotes its domain label, $D(\cdot)$ is the domain discriminator, and BCE denotes the binary cross-entropy loss. In the MLPA main instantiation, UGCA replaces only the deep single-center prototype-alignment branch, while the shallow and middle branches retain the original mean-prototype alignment of the host framework.

The uniform $1/K$ weighting is well-suited to the current few-shot setting. Each class accumulates at most $M = 256$ cached features, and splitting them into $K = 2$ clusters typically yields groups of comparable size. Even when the two clusters are imbalanced, weighting by cluster size would allow the larger cluster to dominate the adversarial signal, effectively reverting to near-single-center behavior. The uniform design ensures that both sub-regions of the feature space receive balanced alignment pressure, which is the primary goal of UGCA.

3.3. Center Regularization

Unlike UGCA, which constitutes the primary methodological contribution of this work, Center Regularization serves as a supporting component that improves embedding compactness and prototype stability, as illustrated in Figure 3.

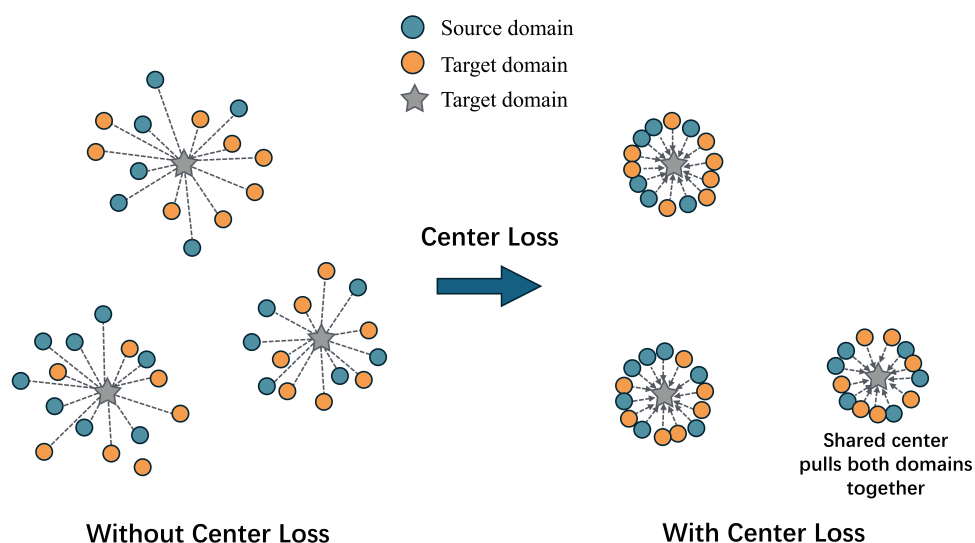


Figure 3. Center Regularization and joint objective. Learnable class centers provide stable geometric anchors, and the Center Loss constrains query embeddings to aggregate around their corresponding centers.

3.3.1. Center Loss Formulation

For each class c , we introduce a persistent, learnable class center $\mathbf{c}_c \in \mathbb{R}^d$, where d is the embedding dimension. Unlike support-derived prototypes that fluctuate in every episode, these learnable centers provide a stable global geometric reference.

In the current implementation, Center Loss is computed on the query embeddings of the source-domain branch and the target-domain branch separately, and the resulting losses are added to the overall training objective. For a batch of query embeddings $\mathbf{z}_i$ with labels y_i , the Center Regularization loss is defined as

$$\mathcal{L}_{center} = \frac{1}{2} \sum_i \|\mathbf{z}_i - \mathbf{c}_{y_i}\|_2^2 \quad (15)$$

This penalty explicitly reduces intra-class variance and encourages sample embeddings to cluster around their respective centers. A tighter class distribution yields a more robust mean prototype (and better-defined sub-clusters), making prototype estimation more resilient to the sampling noise inherent in few-shot settings.

3.3.2. Joint Objective and Update Strategy

The Center Regularization is additively integrated into the baseline objective:

$$\mathcal{L}_{total} = \mathcal{L}_{base} + \lambda_c \mathcal{L}_{center} \quad (16)$$

where $\mathcal{L}_{base}$ encompasses the few-shot classification loss and the UGCA deep adversarial loss with uniformly weighted clustered units, and λ_c is the balancing coefficient.

To guarantee numerical stability, the class centers $\mathbf{c}_c$ are decoupled from the main network optimizer and updated using a separate optimization step with gradient re-scaling. This ensures that the compactness constraint acts as a reliable auxiliary pull without interfering with the feature extractor's primary spectral-spatial learning. For the main network parameters, the contribution of Center Loss is scaled by λ_c , whereas the learnable centers are updated by an auxiliary SGD optimizer after gradient re-scaling.

3.4. Training Strategy and Inference

The proposed training strategy integrates the aforementioned modules into the alternating episodic framework, as illustrated in Figure 4. In each iteration, episodic tasks are sampled to extract shallow, middle, and deep embeddings. The source and target memory banks are updated under the valid-update protocol for D_f , class-level uncertainty proxies are smoothed via EMA, and hard classes are determined by minimum-cache screening together with quantile-threshold and Top-M selection. After warm-up, hard-class clustered prototypes are refreshed periodically and converted into uniformly weighted deep alignment units, while non-hard classes retain single-mean prototypes. Concurrently, $\mathcal{L}_{center}$ is computed to tighten intra-class distributions. The overall process is outlined in Algorithm 1.

During inference, we retain the exact testing protocol of the baseline, as shown in Figure 5. Target-domain testing samples are projected into the learned embedding space and classified based on their Euclidean proximity to the prototypes of the few-labeled target support set. Because our method acts purely as a training-stage enhancement to sculpt a better, more discriminative feature space, it preserves the simplicity and efficiency of the original metric-based inference without altering the forward-pass computation during testing.

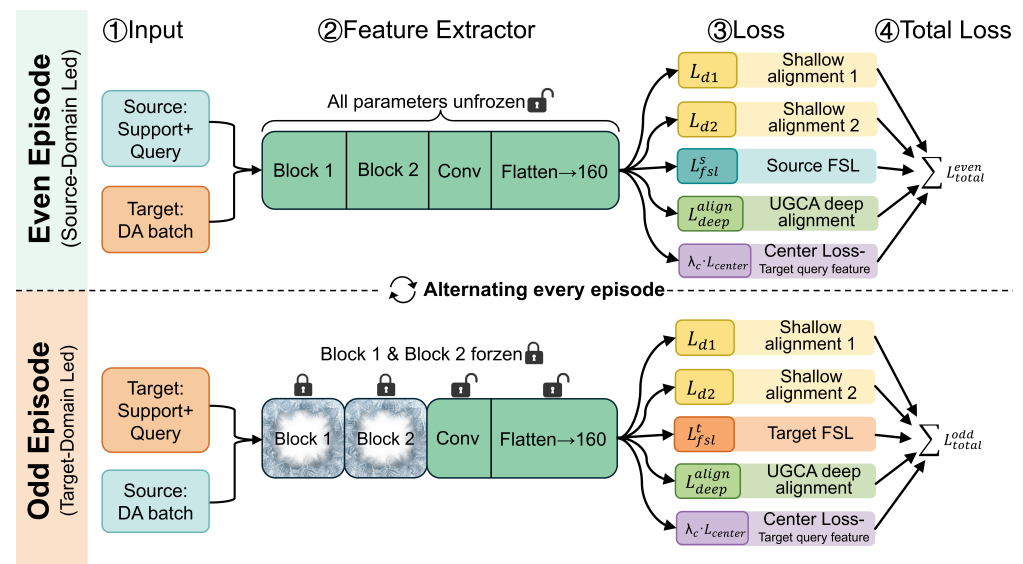


Figure 4. Alternating episodic training. Source-domain and target-domain episodes are alternated; UGCA and Center Regularization are jointly embedded in the training loop.

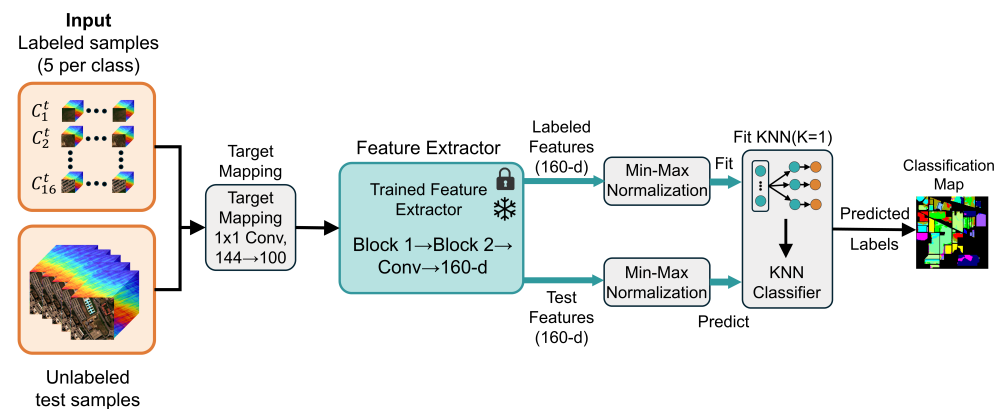


Figure 5. Inference/testing pipeline. At test time, unlabeled target samples are classified by their Euclidean distances to the prototypes constructed from the few-labeled support set. No UGCA or Center Regularization overhead is incurred.

Algorithm 1 Training Procedure of the Proposed Method

Require: Source-domain data D_s ; target-domain few-labeled data D_f

Require: Mapping layers, shared feature extractor, domain discriminator, learnable class centers

- 1: **for** each training iteration (alternating source/target tasks) **do**
- 2: Sample an episodic task and extract shallow, middle, and deep embeddings.
- 3: Update the source-domain memory bank with source training samples and update the target-domain memory bank only with D_f -derived target training samples and their augmentations.
- 4: Compute the class-level uncertainty proxy from the target-domain memory bank and smooth it by EMA.
- 5: Construct the candidate class set using the minimum cache thresholds N_{min}^s and N_{min}^t
- 6: Apply the quantile threshold q and Top-M screening to determine the hard-class set C_{hard}
- 7: If the warm-up condition is satisfied and the current iteration reaches the update interval, run K-means on the hard classes to refresh clustered prototypes.
- 8: Construct deep alignment units: K clustered centers, each with uniform weight $1/K$, for hard classes; a single-mean prototype with weight 1 for all other classes.
- 9: Compute the baseline FSL loss, the host-framework shallow/middle alignment loss, the UGCA deep adversarial loss, and the Center Regularization loss.
- 10: Backpropagate the total loss, apply gradient clipping, update the main network and domain discriminator, re-scale the center gradients, and update the learnable centers with the auxiliary SGD optimizer.
- 11: **end for**

3.5. Overall Objective Function

Combining the few-shot classification loss, the Multi-Level Prototype Alignment loss, and the Center Regularization loss, the overall training objective in each episodic iteration is formulated as

$$\mathcal{L}_{total} = \mathcal{L}_{fsl} + \lambda \left(\mathcal{L}_{adv}^{(1)} + \mathcal{L}_{adv}^{(2)} + \mathcal{L}_{adv}^{deep} \right) + \lambda_c \mathcal{L}_{center}. \quad (17)$$

where $\mathcal{L}_{fsl}$ is the few-shot classification loss in the current episodic iteration, corresponding to the source- or target-domain classification loss under the alternating training protocol. $\mathcal{L}_{adv}^{(1)}$ and $\mathcal{L}_{adv}^{(2)}$ denote the shallow- and middle-level prototype alignment losses inherited from MLPA, respectively. $\mathcal{L}_{adv}^{deep}$ is the UGCA-based deep alignment loss defined in Equation (14), and $\mathcal{L}_{center}$ is the Center Regularization loss defined in Equation (15). The coefficients λ and λ_c control the strengths of prototype alignment and Center Regularization, respectively.

4. Experiments and Analysis

4.1. Datasets and Transfer Tasks

We follow the public experimental setting of the original MLPA paper [31], using Chikusei as the source domain and Indian Pines (IP), the University of Pavia (UP), and Houston (HT) as the target domains. The three main transfer tasks are denoted as Chikusei-to-IP, Chikusei-to-UP, and Chikusei-to-HT, respectively.

The Chikusei dataset [34] (Figure 6) was acquired on 29 July 2014 in Chikusei City, Japan, using a Hyperspec-VNIR-C visible/near-infrared hyperspectral camera. It contains 19 classes, has a spatial size of 2517×2335 pixels, a spatial resolution of 2.5 m/pixel, 128 spectral bands ranging from 363 to 1018 nm, and 77,592 labeled samples.

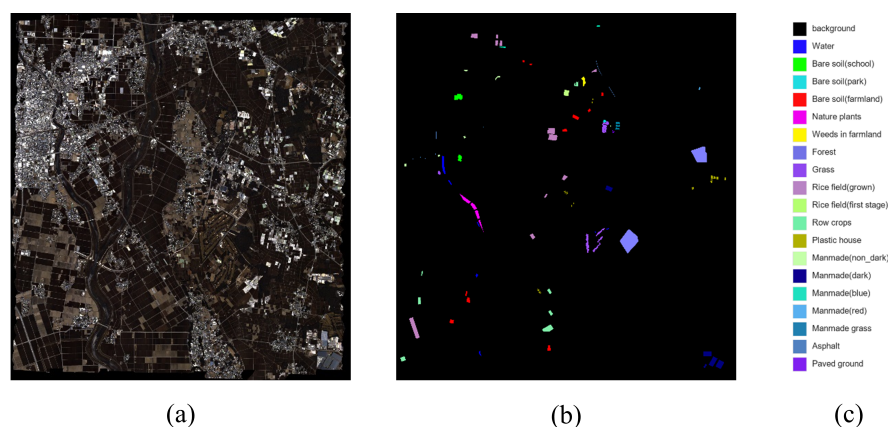


Figure 6. Chikusei dataset. (a) Pseudo-color image, (b) ground-truth map, and (c) class legend of CH.

The Indian Pines (IP) dataset (Figure 7) was acquired by the Airborne Visible/Infrared Imaging Spectrometer (AVIRIS) in 1992 over northwestern Indiana. It contains 16 vegetation classes, has a spatial size of 145×145 pixels, a spatial resolution of 20 m/pixel, 200 spectral bands ranging from 400 to 2500 nm, and 10,249 labeled samples.

The Houston (HT) dataset (Figure 8) was acquired by the ITRES CASI-1500 sensor and was provided by the 2013 IEEE GRSS Data Fusion Contest. It contains 15 land-cover classes, has a spatial size of 349×1905 pixels, 144 spectral bands ranging from 364 to 1046 nm, and 15,029 labeled samples. The detailed statistics of all datasets are summarized in Table 1.

The University of Pavia (UP) dataset (Figure 9) was acquired by the Reflective Optics System Imaging Spectrometer (ROSIS). It contains nine land-cover classes, has a spatial size of 610×340 pixels, a spatial resolution of 1.3 m/pixel, 103 spectral bands ranging from 430 to 860 nm, and 42,776 labeled samples.

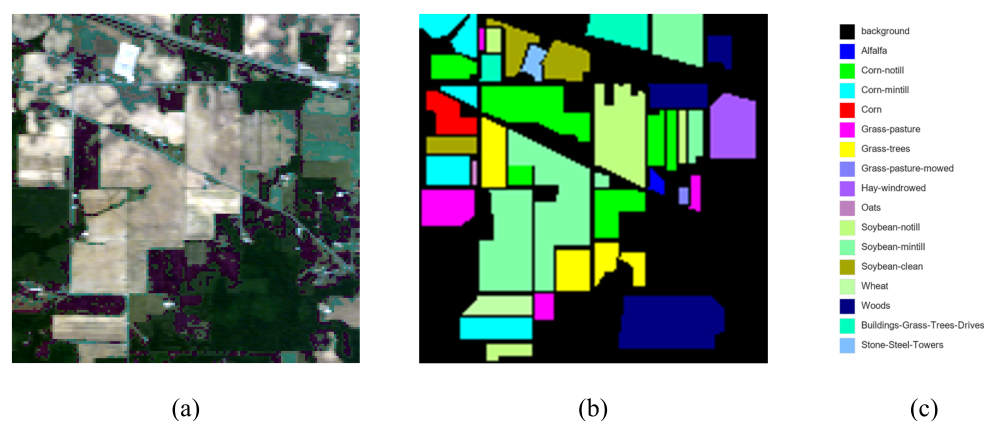


Figure 7. Indian Pines dataset. (a) Pseudo-color image, (b) ground-truth map, and (c) class legend of IP.

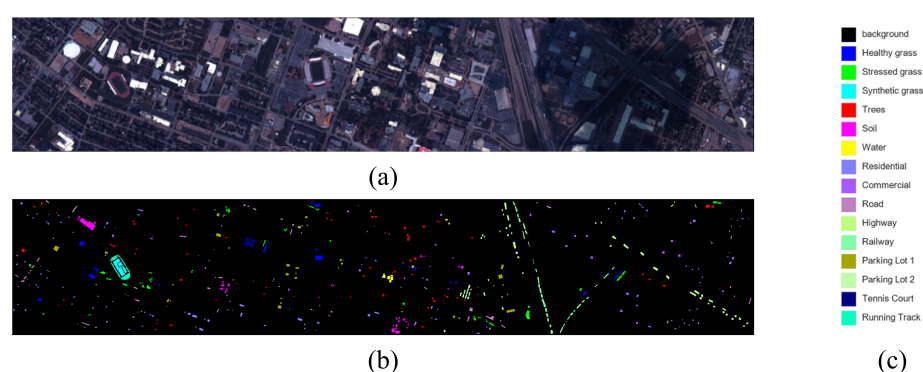


Figure 8. HT dataset. (a) Pseudo-color image, (b) ground-truth map, and (c) class legend of HT.

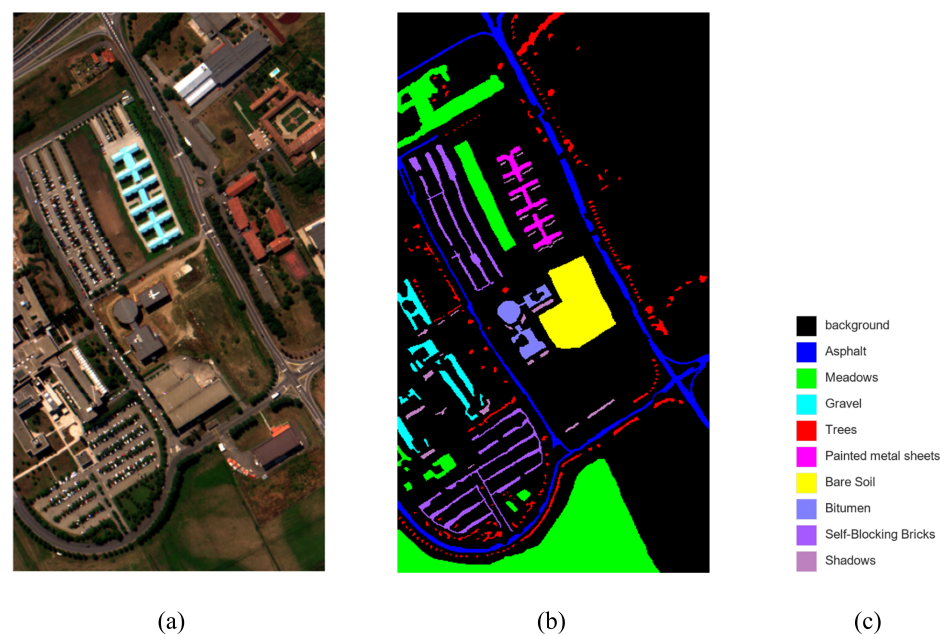


Figure 9. UP dataset. (a) Pseudo-color image, (b) ground-truth map, and (c) class legend of UP.

Table 1. Dataset statistics.

Dataset	Spatial Size	Bands	Spectral Range (nm)	Sensor	Classes	Samples
Chikusei	2517 × 2335	128	363–1018	Hyperspec-VNIR-C	19	77,592
IP	145 × 145	200	400–2500	AVIRIS	16	10,249
UP	610 × 340	103	430–860	ROSIS	9	42,776
HT	349 × 1905	144	364–1046	ITRES CASI-1500	15	15,029

4.2. Experimental Settings and Evaluation Protocol

All experiments were conducted on a Windows platform using Python 3.8 and PyTorch 1.12.1 with CUDA 11.3. Training and evaluation were conducted on two NVIDIA GeForce RTX 3090 GPUs with 24 GB memory each and an Intel Core i9-10980XE CPU. Following the experimental setup of the original MLPA paper [31], the meta-training phase constructs one-shot episodic tasks, while the testing phase fits a KNN classifier with five labeled samples per class in the target domain and evaluates on the remaining target pixels. This protocol is consistent with that of MLPA, DCFSL [23], and Gia-CFSL [24], ensuring direct comparability across all reported results.

In terms of implementation details, the main instantiation uses an input patch size of 9×9 and a feature dimension of 160. The augmentation applied to labeled target-domain training samples is a radiation-based random spectral noise perturbation. Specifically, when constructing the augmented target-domain training set, light random radiation noise is applied to the original spectral-spatial patches through the function `radiation_noise`, so that sample diversity is enriched while core class semantics and local spatial structure are preserved as much as possible.

This augmentation strategy is consistent with the idea of using random Gaussian noise in the original MLPA paper, but here it is further used to populate the target-domain memory bank and thus support class-level dispersion estimation and hard-class selection. The backbone and domain discriminator are both optimized by Adam. For the MLPA main instantiation, the initial learning rate is 0.001 and the total number of training episodes is 10,000. Center Loss is updated with an auxiliary SGD optimizer, where $\lambda_c = 0.005$ and the center learning rate `lr_centers` = 0.5. The UGCA-related hyperparameters include memory bank size $M = 256$, EMA momentum $\beta = 0.9$, cluster number $K = 2$, Top-M = 4, quantile threshold $q = 0.75$, minimum cache sizes `UGCA_MIN_SRC` = 24 and `UGCA_MIN_TAR` = 24, warm-up episodes $E_w = 200$, and update interval $T_{up} = 10$. During training, `CosineAnnealingLR` ($\eta_{\min} = 10^{-6}$) and gradient clipping with `max_norm` = 10 are used. Gradient re-scaling is additionally applied to the Center Loss parameters. All convolutional layers are initialized using the Xavier scheme. To reduce the impact of random sampling, we use the same fixed random seeds as MLPA and Gia-CFSL, and each experiment is repeated 10 times to report the mean and standard deviation of overall accuracy (OA), average accuracy (AA), and the Kappa coefficient.

Table 2 summarizes the key hyperparameters and implementation settings. The Gia-CFSL adaptation code follows the same UGCA logic and uses independent hyperparameter settings ($\lambda_c = 0.003$ and `lr_centers` = 0.03).

Table 2. Key hyperparameters and implementation settings.

Parameter	MLPA (Main)	Gia-CFSL (Plug-and-Play)
Input patch	9×9	Follow the host setting
Training episodes	10,000	Follow the host setting
Backbone optimizer	Adam	Adam
Backbone learning rate	0.001	Set by the configuration file
Center Loss weight	0.005	0.003
Center optimizer	SGD	SGD
Center learning rate	0.5	0.03
Center gradient handling	Re-scaling	Re-scaling
Memory bank size M	256	256
EMA momentum	0.9	0.9
Cluster number K	2	2
Top-M	4	4
Quantile threshold q	0.75	0.75
Minimum cached samples	24/24	24/24
Warm-up episodes	200	200
Update interval	10	10
Learning-rate schedule	<code>CosineAnnealingLR</code>	<code>CosineAnnealingLR</code>
Gradient clipping	<code>max_norm</code> = 10	<code>max_norm</code> = 10

4.3. Comparison with Baselines and Representative Methods

4.3.1. Main-Protocol Comparison with the MLPA Baseline

To verify the effectiveness of the proposed module, we run the original MLPA baseline and the full model (MLPA + UGCA + Center Regularization) in the same experimental environment. Table 3 summarizes the main-protocol comparison results on the three target domains. On the IP dataset, the proposed method improves OA, AA, and Kappa from 69.21%, 79.88%, and 65.32% to 71.24%, 81.90%, and 67.67%, respectively, achieving improvements of 2.03%, 2.02%, and 2.35% over the MLPA baseline. On the UP dataset, the proposed method improves OA, AA, and Kappa from 81.90%, 82.53%, and 76.57% to 83.73%, 85.30%, and 78.97%, respectively, with corresponding gains of 1.83%, 2.77%, and 2.40%. On the HT dataset, the proposed method improves OA, AA, and Kappa from 76.92%, 79.74%, and 75.07% to 78.24%, 81.08%, and 76.50%, respectively, with corresponding gains of 1.32%, 1.34%, and 1.43%. The reported OA, AA, and Kappa gains are consistent in sign across all three target domains. A detailed component ablation is given in Section 4.4.

Table 3. Main-protocol comparison between the MLPA baseline and Ours (%).

Dataset	OA (%)			AA (%)			Kappa (%)		
	MLPA	Ours	Gain	MLPA	Ours	Gain	MLPA	Ours	Gain
IP	69.21	71.24	+2.03	79.88	81.90	+2.02	65.32	67.67	+2.35
UP	81.90	83.73	+1.83	82.53	85.30	+2.77	76.57	78.97	+2.40
HT	76.92	78.24	+1.32	79.74	81.08	+1.34	75.07	76.50	+1.43

4.3.2. Comparison with Representative Existing Methods

Since most published results are also based on protocols compatible with MLPA, we report them separately as horizontal comparison tables. Tables 4–6 present the five-shot comparison results on the IP, UP, and HT datasets, respectively. Among them, MLPA, Gia-CFSL, and Ours are all reproduced by us in the same experimental environment, while the remaining methods are taken from the results reported in the original MLPA paper [31]. The compared baseline methods include SVM [3], 3DCNN [5], SSRN [7], DFSL+NN [19], and RN-FSC [17,20].

Table 4. Classification results (%) on the IP dataset (five shots). The numbers in bold indicate the highest performance metrics of all methods and the underlined result is the sub-optimal result.

Class	SVM	3DCNN	SSRN	DFSL+NN	RN-FSC	DCFSL	Gia-CFSL	SC Former-R	MLPA	Ours
1	72.20	95.12	18.38	96.75	<u>96.34</u>	95.85	86.34	87.80	93.90	96.10
2	34.27	37.70	64.79	38.65	46.13	41.88	38.10	46.13	<u>55.93</u>	54.78
3	39.18	19.77	27.65	42.79	40.61	47.96	42.21	45.68	<u>59.85</u>	65.83
4	50.34	32.51	26.97	68.10	58.62	79.53	69.61	64.01	<u>80.56</u>	86.98
5	69.75	88.45	<u>80.76</u>	71.20	64.96	71.76	71.09	73.87	75.61	79.04
6	66.36	73.65	86.87	76.18	69.45	84.72	76.68	<u>87.54</u>	85.28	90.52
7	89.13	81.82	32.24	100.00	100.00	99.13	99.57	96.96	<u>99.57</u>	100.00
8	68.73	53.35	100.00	74.84	77.70	83.70	<u>90.19</u>	86.15	89.28	89.15
9	86.67	100.00	57.69	100.00	100.00	99.33	98.67	98.67	<u>99.33</u>	100.00
10	37.49	41.35	59.69	47.98	25.49	60.94	57.67	58.42	<u>64.89</u>	67.55
11	33.96	<u>66.71</u>	70.87	57.95	65.51	59.47	60.41	64.48	60.89	63.07
12	31.43	37.40	45.00	38.21	27.13	46.68	39.22	34.46	<u>52.77</u>	54.25
13	86.50	85.71	88.29	97.50	99.75	97.75	96.80	98.20	97.30	<u>99.05</u>
14	62.93	62.57	97.18	83.44	76.35	84.71	78.44	85.35	85.02	<u>86.09</u>
15	28.08	56.42	36.64	62.29	70.34	68.45	64.07	66.85	78.95	<u>78.61</u>
16	90.91	90.36	60.98	100.00	100.00	98.64	97.27	99.89	98.98	99.32
OA	45.85 ± 2.44	54.76 ± 0.03	61.36 ± 0.49	59.65 ± 0.63	58.17 ± 0.02	64.29 ± 2.76	61.24 ± 3.34	65.01 ± 2.44	<u>69.21</u> ± 2.71	71.24 ± 3.39
AA	59.24 ± 1.36	63.93 ± 0.02	59.75 ± 0.20	72.24 ± 0.42	69.90 ± 0.40	76.28 ± 1.24	72.90 ± 1.69	74.65 ± 3.23	<u>79.88</u> ± 1.90	81.90 ± 1.64
Kappa	39.68 ± 2.48	48.72 ± 0.03	56.91 ± 0.48	54.55 ± 0.52	52.52 ± 0.14	59.87 ± 2.82	56.36 ± 3.56	60.20 ± 2.79	<u>65.32</u> ± 2.76	67.67 ± 3.58

Table 5. Classification results (%) on the UP dataset (five shots). The numbers in bold indicate the highest performance metrics of all methods and the underlined result is the sub-optimal result.

Class	SVM	3DCNN	SSRN	DFSL+ NN	RN- FSC	DCFSL	Gia- CFSL	SC Former -R	MLPA	Ours
1	88.98	59.82	91.84	69.19	68.55	78.09	77.13	76.92	80.33	80.74
2	83.91	63.05	95.13	84.63	<u>93.44</u>	81.63	85.47	90.92	85.50	86.46
3	39.98	68.91	55.23	57.47	<u>49.81</u>	64.72	60.32	<u>69.26</u>	65.01	72.47
4	60.22	77.31	78.02	89.99	92.15	94.65	93.13	86.54	<u>93.24</u>	92.53
5	95.44	90.77	98.34	100	99.43	99.38	96.93	98.90	99.07	98.99
6	37.12	63.40	53.56	71.23	<u>57.99</u>	74.94	71.57	57.50	73.74	<u>73.80</u>
7	40.62	87.64	60.07	70.62	70.04	78.44	77.01	<u>86.97</u>	78.09	86.55
8	68.17	57.27	85.34	58.13	63.48	68.25	71.31	74.21	68.62	<u>77.52</u>
9	99.13	95.57	98.08	96.92	<u>99.19</u>	98.62	96.21	99.08	99.25	98.61
OA	64.12 ± 4.55	65.74 ± 1.77	76.26 ± 5.78	77.75 ± 1.16	80.19 ± 2.18	80.08 ± 3.92	80.97 ± 3.51	<u>82.31</u> ± 2.06	81.90 ± 3.45	83.73 ± 3.89
AA	68.18 ± 2.27	73.72 ± 1.01	79.51 ± 3.21	77.57 ± 0.31	77.12 ± 0.84	82.08 ± 1.91	81.01 ± 1.73	<u>82.25</u> ± 2.60	<u>82.53</u> ± 1.72	85.30 ± 1.88
Kappa	55.59 ± 4.64	57.37 ± 1.97	70.56 ± 6.69	71.11 ± 1.22	73.73 ± 2.79	74.46 ± 4.45	75.37 ± 4.04	76.55 ± 2.62	<u>76.57</u> ± 4.09	78.97 ± 4.76

Table 6. Classification results (%) on the HT dataset (five shots). The numbers in bold indicate the highest performance metrics of all methods and the underlined result is the sub-optimal result.

Class	SVM	3DCNN	SSRN	DCFSL	Gia- CFSL	MLPA	Ours
1	81.03	<u>89.24</u>	81.94	90.20	88.64	88.14	88.92
2	74.54	77.15	80.72	74.45	79.63	<u>87.20</u>	87.21
3	<u>97.60</u>	95.06	99.68	94.65	94.15	95.95	95.51
4	86.28	88.15	<u>89.81</u>	91.38	89.39	88.83	88.56
5	89.06	94.68	95.78	<u>96.90</u>	91.96	93.01	97.58
6	82.52	72.12	83.57	<u>84.34</u>	72.16	83.88	86.19
7	55.96	54.53	66.79	59.29	58.78	<u>67.32</u>	70.01
8	24.85	42.38	22.36	42.59	39.85	<u>44.97</u>	47.75
9	58.49	59.98	61.50	59.95	58.40	<u>66.66</u>	67.26
10	47.67	53.29	49.14	66.79	64.65	69.35	68.02
11	51.04	61.13	77.97	56.02	60.90	61.64	<u>64.49</u>
12	39.67	44.77	40.70	75.42	72.65	67.61	69.45
13	19.61	31.56	75.09	86.73	83.86	<u>90.84</u>	92.22
14	93.48	86.36	99.80	88.60	89.08	92.25	<u>95.72</u>
15	96.05	96.56	99.77	94.65	86.08	<u>98.46</u>	97.27
OA	63.75 ± 2.26	68.58 ± 2.58	70.46 ± 1.24	74.22 ± 2.92	73.24 ± 2.13	<u>76.92</u> ± 1.10	78.24 ± 1.55
AA	66.52 ± 1.91	69.80 ± 2.61	74.97 ± 0.99	77.47 ± 2.62	75.34 ± 2.08	<u>79.74</u> ± 1.25	81.08 ± 1.06
Kappa	60.89 ± 2.39	65.92 ± 2.76	68.08 ± 1.33	72.16 ± 3.16	71.10 ± 2.29	<u>75.07</u> ± 1.19	76.50 ± 1.67

On all three target-domain datasets, the proposed method achieves the best performance. Compared with the reproduced MLPA baseline, it improves OA/AA/Kappa by 2.03%/2.02%/2.35% on IP, 1.83%/2.77%/2.40% on UP, and 1.32%/1.34%/1.43% on HT, respectively. In particular, compared with DCFSL on HT, the proposed method further improves OA, AA, and Kappa by 4.02%, 3.61%, and 4.34%, respectively.

The classification maps in Figures 10–12 provide a qualitative view of spatial prediction quality. To give a quantitative reference, we highlight representative per-class accuracy changes from the detailed tables above. On the IP dataset, easily confused vegetation classes show clear gains: Class 3 improves from 59.85% to 65.83% (+5.98%), Class 4 from 80.56% to 86.98% (+6.42%), and Class 6 from 85.28% to 90.52% (+5.24%); these gains visually correspond to the suppression of patch-like errors among spectrally similar classes. On the UP dataset, several heterogeneous urban classes show clear gains: Class 3 improves from 65.01% to 72.47% (+7.46%), Class 7 from 78.09% to 86.55% (+8.46%), and Class 8 from 68.62% to 77.52% (+8.90%). On the HT dataset, Class 5 improves from 93.01% to 97.58% (+4.57%) and Class 13 from 90.84% to 92.22% (+1.38%), corresponding to fewer misclassifications near roads and building boundaries. These per-class improvements confirm that the proposed

multi-center alignment together with Center Regularization improves not only overall accuracy but also spatial consistency for structurally complex categories.

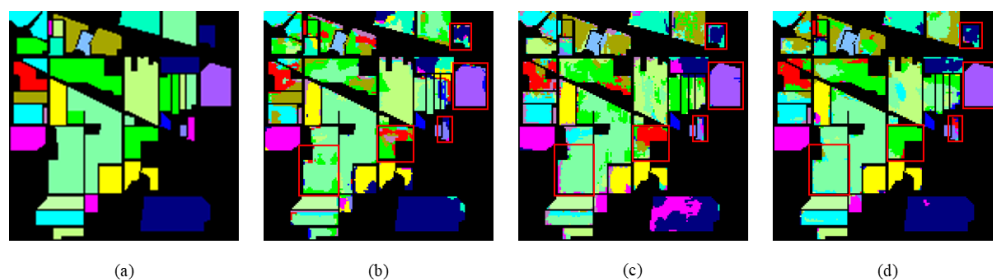


Figure 10. Visualization results of various methods on the IP dataset. (a) Ground truth, (b) Gia-CFSL, (c) MLPA, and (d) Ours.

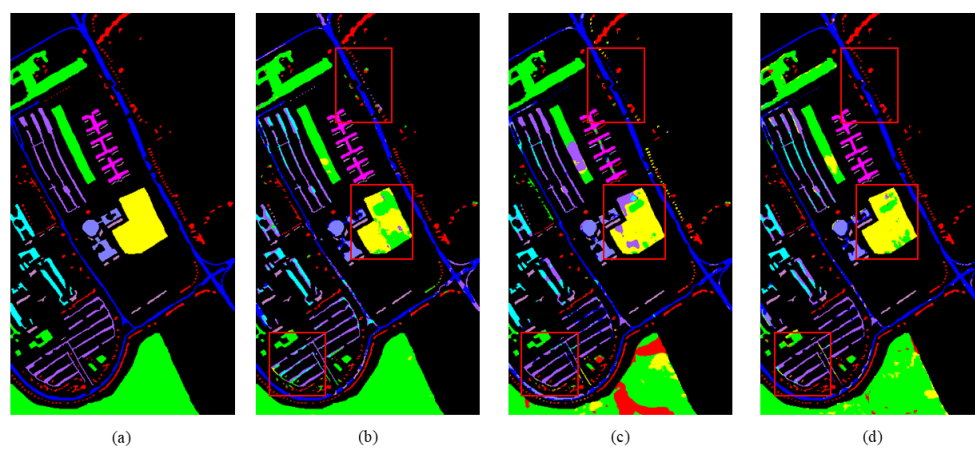


Figure 11. Visualization results of various methods on the UP dataset. (a) Ground truth, (b) Gia-CFSL, (c) MLPA, and (d) Ours.

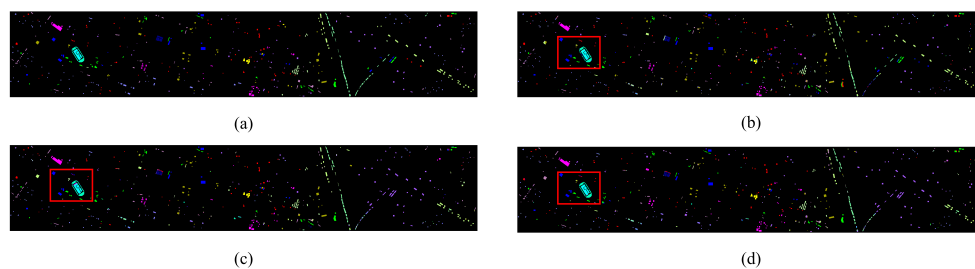


Figure 12. Visualization results of various methods on the HT dataset. (a) Ground truth, (b) Gia-CFSL, (c) MLPA, and (d) Ours.

4.4. Ablation Experiments

4.4.1. Component Ablation

The component ablation addresses two questions. First, can UGCA alone improve deep alignment for hard classes? Second, can Center Regularization improve intra-class stability without modifying the testing path of the host framework? Tables 7–9 report the ablation results of four variants on the three target-domain datasets.

Several trends can be observed from Tables 7–9. First, on all three target domains, the full model (UGCA + Center Regularization) achieves the best performance, indicating that the two modules provide complementary gains when used jointly. On the IP dataset, Center Regularization alone contributes an OA gain of 0.66%, whereas UGCA alone contributes only 0.18%. When combined, the OA gain reaches 2.03%, which is clearly larger than the sum of the single-module gains (0.84%), demonstrating a strong synergy. Second,

on the HT dataset, UGCA only improves OA by 0.74%, whereas Center Regularization only improves OA by 0.07%, indicating that multi-center alignment is more critical for complex categories in this urban scene. Third, on the UP dataset, neither individual module yields a noticeable gain, and both even exhibit slight fluctuations, but their combination still improves OA by 1.83%. This further suggests that the gains of UGCA and Center Regularization are not simply additive; instead, they are complementary in enhancing complex-category representation and stabilizing query representations.

Table 7. Component ablation results on the IP dataset. The numbers in bold indicate the highest performance metrics.

Method	OA (%)	AA (%)	Kappa (%)
MLPA (original baseline)	69.21 ± 2.71	79.88 ± 1.90	65.32 ± 2.76
MLPA + UGCA only	69.39 ± 3.32	80.32 ± 1.58	65.56 ± 3.53
MLPA + Center Regularization only	69.87 ± 2.32	80.66 ± 0.77	66.12 ± 2.37
MLPA + UGCA + Center Regularization (Ours)	71.24 ± 3.39	81.90 ± 1.64	67.67 ± 3.58

Table 8. Component ablation results on the UP dataset. The numbers in bold indicate the highest performance metrics.

Method	OA (%)	AA (%)	Kappa (%)
MLPA (original baseline)	81.90 ± 3.45	82.53 ± 1.72	76.57 ± 4.09
MLPA + UGCA only	81.90 ± 4.50	83.00 ± 1.86	76.62 ± 5.14
MLPA + Center Regularization only	81.87 ± 4.39	82.37 ± 2.35	76.55 ± 5.03
MLPA + UGCA + Center Regularization (Ours)	83.73 ± 3.89	85.30 ± 1.88	78.97 ± 4.76

Table 9. Component ablation results on the HT dataset. The numbers in bold indicate the highest performance metrics.

Method	OA (%)	AA (%)	Kappa (%)
MLPA (original baseline)	76.92 ± 1.10	79.74 ± 1.25	75.07 ± 1.19
MLPA + UGCA only	77.66 ± 1.21	80.36 ± 0.94	75.87 ± 1.31
MLPA + Center Regularization only	76.99 ± 1.35	79.80 ± 1.09	75.16 ± 1.46
MLPA + UGCA + Center Regularization (Ours)	78.24 ± 1.55	81.08 ± 1.06	76.50 ± 1.67

4.4.2. Center Loss Learning-Rate Ablation

To examine the sensitivity of Center Regularization to the learning rate of its auxiliary center optimizer, we vary `lr_centers` on the IP target domain while keeping all other training settings unchanged. Table 10 reports the resulting OA, AA, and Kappa values over 10 independent runs.

Table 10. Center learning-rate ablation for Center Loss on the IP dataset. The numbers in bold indicate the highest performance metrics.

<code>lr_centers</code>	OA (%)	AA (%)	Kappa (%)
0.05	69.70 ± 2.66	80.76 ± 1.04	65.87 ± 2.85
0.1	70.59 ± 2.92	81.18 ± 1.55	66.86 ± 3.18
0.3	71.12 ± 4.38	81.52 ± 2.06	67.47 ± 4.77
0.5	71.24 ± 3.39	81.90 ± 1.64	67.67 ± 3.58
1.0	69.89 ± 2.75	80.90 ± 1.45	66.12 ± 2.85

The results show that a moderate center learning rate provides the best overall performance. In particular, `lr_centers` = 0.5 achieves the highest OA (71.24%), AA (81.90%), and Kappa (67.67%). Lower learning rates lead to more conservative updates of the learnable class centers, whereas increasing the learning rate to 1.0 reduces all three metrics,

suggesting that overly aggressive center updates may destabilize the compactness constraint. These results support the use of 0.5 as the default center learning rate in the main experiments.

4.4.3. Analysis of the UGCA Alignment Level

To investigate at which level of the feature extractor UGCA should be inserted, we conduct an alignment-level ablation study on the HT dataset. Three configurations are compared: deep only (the default setting), deep plus shallow, and shallow plus middle plus deep. In all configurations, any level not replaced by UGCA retains the original single-center prototype alignment of MLPA.

As shown in Table 11, inserting UGCA only at the deep level yields the best performance (OA = 78.24%), whereas inserting UGCA at more levels leads to performance degradation. A plausible explanation is that, in the feature extraction network, shallow features carry mostly low-level generic information such as edges and textures, and their class structures are relatively simple, so single-center prototypes are usually sufficient. Middle-level features begin to form semantic category information but are still not fully mature, and forcing multi-center clustered alignment at these stages provides little structural compensation; instead, the high variance of shallow and middle cached features may introduce noisy clusters and interfere with the originally effective single-center alignment. By contrast, deep features have the strongest semantic discriminability, their class structures are the most mature, and they are the most likely to require multi-center characterization of complex multi-modal distributions. Concentrating UGCA at the deep level is therefore the most reasonable design choice, and this result also validates our decision to position UGCA as a deep-alignment enhancement rather than a full-level replacement.

Table 11. UGCA alignment-level ablation on the HT dataset. The numbers in bold indicate the highest performance metrics.

UGCA Insertion Position	OA (%)	AA (%)	Kappa (%)
Deep only (default)	78.24 ± 1.55	81.08 ± 1.06	76.50 ± 1.67
Deep + shallow	77.18 ± 1.26	80.24 ± 0.83	75.36 ± 1.36
Shallow + middle + deep	77.65 ± 1.28	80.62 ± 1.13	75.87 ± 1.38

4.4.4. Analysis of Hard-Class Selection

To verify that the uncertainty-guided hard-class selection mechanism identifies semantically meaningful classes, we record the UGCA triggering statistics across the three target domains over 10 independent runs. Figure 13 summarizes the results.

As shown in Figure 13, the triggering frequency varies substantially across classes within each dataset, confirming that the uncertainty-guided selection mechanism is adaptive rather than random. On the IP dataset, Class 8 (Hay-windrowed) is triggered most frequently (54.0%), followed by Class 16 (Stone-Steel-Towers, 36.5%) and Class 10 (Soybean-notill, 35.4%). The frequently triggered classes on IP are dominated by spectrally similar crop types (e.g., the Soybean and Corn families) and mixed land-cover categories whose intra-class variability is well documented. On the UP dataset, Class 2 (Meadows, 55.0%) and Class 7 (Bitumen, 41.5%) rank highest; Meadows is the largest spatially contiguous class with considerable internal spectral variability, while Bitumen is a heterogeneous urban material. On the HT dataset, Class 8 (Commercial, 47.6%) and Class 3 (Synthetic grass, 40.1%) are the most frequently triggered, both of which exhibit complex spatial structures that are difficult to characterize with a single prototype.

Comparing the triggering statistics with the per-class accuracy gains in Tables 4–6 reveals a consistent pattern for several high-trigger classes. On the UP dataset, Class 7

(Bitumen) has the second-highest trigger frequency (41.5%) and achieves a large per-class gain (+8.46%), while Class 3 (Gravel, 40.5%) gains +7.46%. At the same time, Class 8 (Bricks) achieves the largest UP per-class gain (+8.90%) despite a moderate trigger frequency of 18.6%, suggesting that Center Regularization also benefits classes that are not selected as hard as frequently. On the HT dataset, Class 8 (Commercial, 47.6%) gains +2.78% and Class 14 (Tennis Court, 39.3%) gains +3.47%. On the IP dataset, high-trigger classes such as Class 10 (35.4%) and Class 11 (30.7%) also show gains of +2.66 and +2.18%, respectively. Class 4 (Corn) on IP also gains +6.42% with a trigger frequency of only 21.6%. These improvements are attributable to Center Regularization, which tightens intra-class distributions globally and benefits all classes regardless of whether they are selected as hard. This observation is consistent with the synergy between UGCA and Center Regularization reported in the component ablation (Section 4.4.1).

To complement the triggering statistics with a qualitative view of the learned feature space, Figure 14 presents t-SNE visualizations of the deep target-domain embeddings for representative classes that are frequently selected as hard. Compared with the MLPA baseline, the proposed method generally produces more clearly organized class-wise distributions, particularly on the UP and HT datasets. These results provide feature-level evidence that uncertainty-guided multi-center alignment and Center Regularization reshape the embedding structure of complex target classes under sparse supervision.

Figure 14 visualizes the feature distributions of representative hard classes identified by UGCA. Compared with the MLPA baseline, the proposed method produces more discriminative representations for these hard classes by organizing their complex distributions into more clearly separated class-wise regions while preserving meaningful local substructures. This observation demonstrates the effectiveness of uncertainty-guided hard-class selection and fine-grained multi-center alignment.

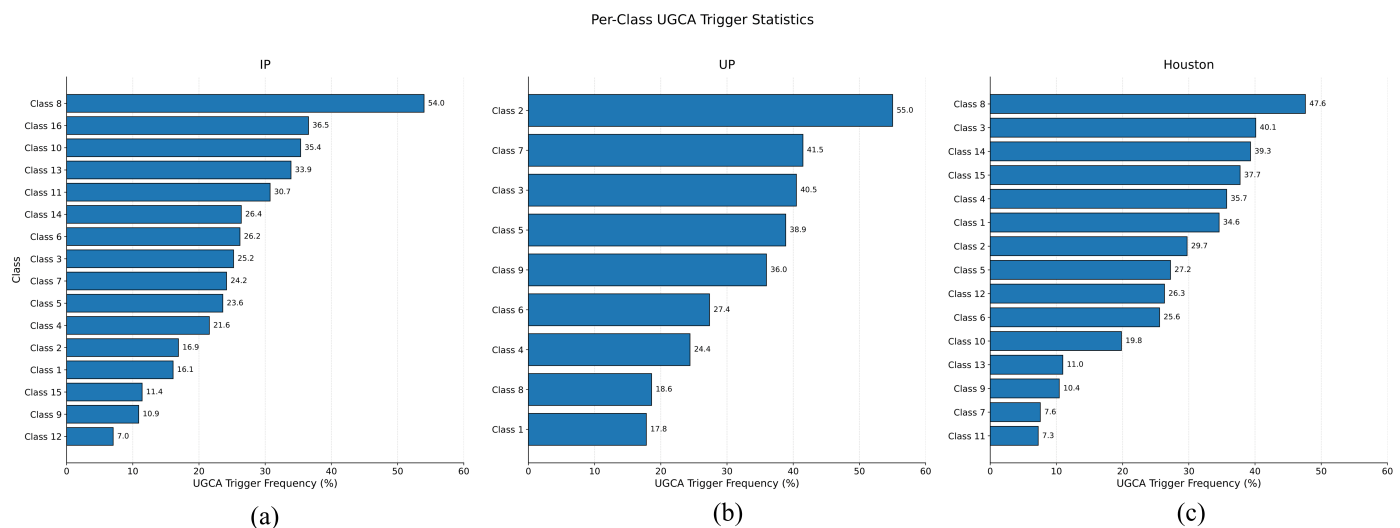


Figure 13. Per-class UGCA triggering frequency (%) on the three target domains over 10 independent runs. (a) IP, (b) UP, and (c) HT. A higher value indicates that the class is more frequently selected into C_{hard} during training.

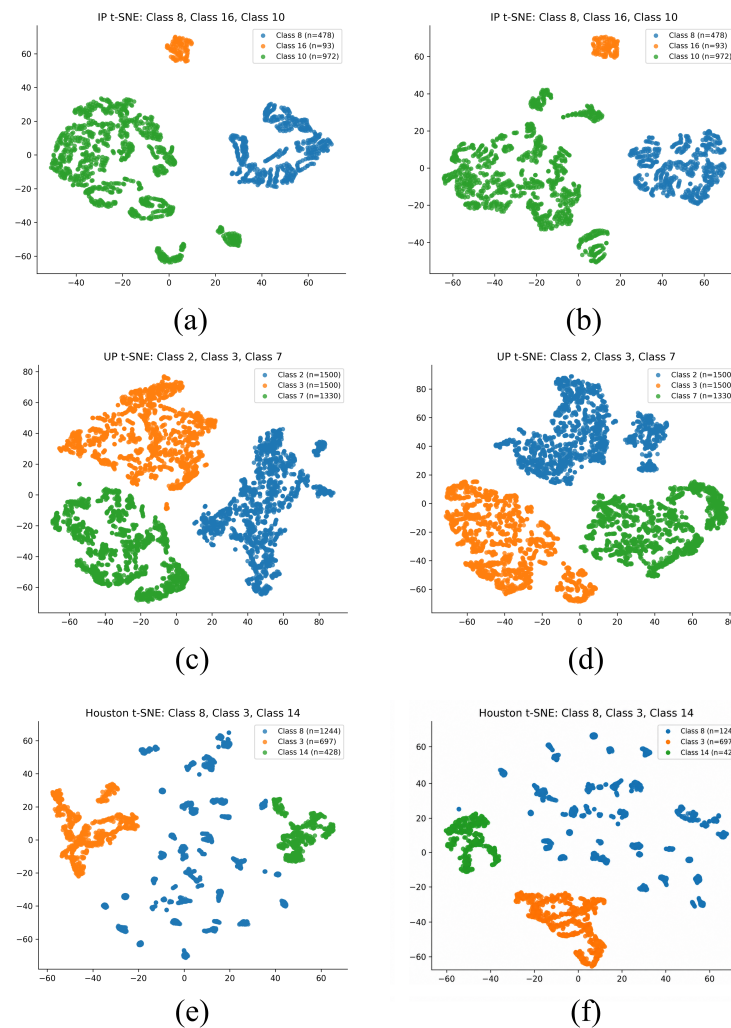


Figure 14. Two-dimensional feature visualization on three datasets. (a) MLPA features on IP. (b) UGCA features on IP. (c) MLPA features on UP. (d) UGCA features on UP. (e) MLPA features on HT. (f) UGCA features on HT.

4.5. Plug-and-Play Validation: Cross-Framework Experiments on Gia-CFSL

To show that the proposed enhancement does not rely on the specific multi-level structure of MLPA, we further conduct cross-framework plug-and-play validation on Gia-CFSL [24]. Specifically, we run the original Gia-CFSL framework on the three target-domain tasks, namely Chikusei-to-HT, Chikusei-to-IP, and Chikusei-to-UP, and compare it with the version enhanced by our module under the same settings. The results are summarized in Table 12.

Table 12. Plug-and-play validation on Gia-CFSL (Chikusei-to-HT/IP/UP). The numbers in bold indicate the highest performance metrics.

Dataset	Gia-CFSL			Gia-CFSL + Ours			OA Gain
	OA	AA	Kappa	OA	AA	Kappa	
HT	73.24 ± 2.13	75.34 ± 2.08	71.10 ± 2.29	74.48 ± 2.11	77.13 ± 1.55	72.44 ± 2.27	+1.24
IP	61.24 ± 3.34	72.90 ± 1.69	56.36 ± 3.56	63.79 ± 3.39	75.79 ± 2.13	59.31 ± 3.65	+2.55
UP	80.97 ± 3.51	81.01 ± 1.73	75.37 ± 4.04	81.87 ± 2.86	81.03 ± 1.76	76.34 ± 3.20	+0.90

As shown in Table 12, plugging the proposed module into Gia-CFSL improves OA on all three target domains, with the magnitude of the improvement depending on the dataset. On the HT dataset, OA, AA, and Kappa are improved from 73.24%, 75.34%,

and 71.10% to 74.48%, 77.13%, and 72.44%, respectively, corresponding to gains of 1.24%, 1.79%, and 1.34%. On the IP dataset, OA, AA, and Kappa are improved from 61.24%, 72.90%, and 56.36% to 63.79%, 75.79%, and 59.31%, respectively, corresponding to gains of 2.55%, 2.89%, and 2.95%. On the UP dataset, OA and Kappa are improved from 80.97% and 75.37% to 81.87% and 76.34%, with corresponding gains of 0.90% and 0.97%, while AA remains essentially unchanged (81.01% vs. 81.03%). The gain on IP is particularly notable, since IP contains 16 fine-grained vegetation classes whose intra-class dispersion is pronounced, and multi-center alignment therefore provides a larger benefit than on the more spatially homogeneous UP urban classes.

It is important to note that the objective of the plug-and-play study is not to demonstrate universality across all cross-domain few-shot learning architectures. Instead, the purpose is to verify whether the proposed prototype-stability enhancement can be transferred across different prototype-driven frameworks. Since both UGCA and Center Regularization operate directly on prototype representations and prototype-alignment units, the current design is inherently tied to prototype-based learning paradigms. These results indicate that the module's effectiveness does not depend on the specific multi-level architecture of MLPA and can transfer to other prototype-driven cross-domain few-shot frameworks as a host-agnostic enhancement.

4.6. Parameter Sensitivity Analysis

We extend the parameter sensitivity analysis from the HT target domain to all three target domains, namely IP, UP, and HT. Each experiment varies one hyperparameter at a time while keeping the remaining settings fixed. The selected shared settings are $K = 2$, $q = 0.75$, $M = 256$, $E_w = 200$, and $T_{up} = 10$, unless otherwise specified.

4.6.1. Cluster Number K

As shown in Table 13, $K = 2$ consistently provides the best overall performance across the three target domains. Compared with $K = 1$, the multi-center setting better captures hard-class substructures, while increasing K to 3 or 4 does not bring further gains and may reduce clustering reliability under sparse supervision. Therefore, $K = 2$ is retained as the selected setting.

Table 13. Sensitivity analysis of the cluster number K on three target datasets. The numbers in bold indicate the highest performance metrics.

K	IP			UP			HT		
	OA (%)	AA (%)	Kappa (%)	OA (%)	AA (%)	Kappa (%)	OA (%)	AA (%)	Kappa (%)
1	70.40 ± 3.76	81.49 ± 1.66	66.74 ± 4.04	83.46 ± 5.32	85.05 ± 2.32	78.67 ± 6.21	77.35 ± 1.39	80.28 ± 1.02	75.55 ± 1.49
2 (selected)	71.24 ± 3.39	81.90 ± 1.64	67.67 ± 3.58	83.73 ± 3.89	85.30 ± 1.88	78.97 ± 4.76	78.24 ± 1.55	81.08 ± 1.06	76.50 ± 1.67
3	71.21 ± 3.60	81.82 ± 1.75	67.60 ± 3.91	83.36 ± 4.54	85.13 ± 1.54	78.54 ± 5.25	77.14 ± 1.27	79.97 ± 1.07	75.32 ± 1.36
4	70.65 ± 3.42	81.42 ± 1.23	66.98 ± 3.68	83.26 ± 3.63	84.57 ± 1.56	78.36 ± 4.33	77.64 ± 1.34	80.44 ± 1.45	75.86 ± 1.43

4.6.2. Quantile Threshold q

As shown in Table 14, $q = 0.75$ achieves the strongest overall performance across the three target domains. On IP, $q = 0.50$, $q = 0.60$, and $q = 0.75$ yield the same mean OA, AA, and Kappa values, while $q = 0.75$ produces smaller deviations for OA and Kappa. On UP and HT, $q = 0.75$ obtains the best overall metrics among the tested thresholds. These results indicate that $q = 0.75$ provides an appropriate balance between hard-class coverage and selective activation.

Table 14. Sensitivity analysis of the quantile threshold q on three target datasets. The numbers in bold indicate the highest performance metrics.

q	IP			UP			HT		
	OA (%)	AA (%)	Kappa (%)	OA (%)	AA (%)	Kappa (%)	OA (%)	AA (%)	Kappa (%)
0.50	71.24 ± 3.69	81.90 ± 1.64	67.67 ± 3.98	83.02 ± 3.70	85.14 ± 2.09	78.95 ± 4.40	77.17 ± 0.99	80.02 ± 0.94	75.35 ± 1.08
0.60	71.24 ± 3.69	81.90 ± 1.64	67.67 ± 3.98	83.02 ± 3.70	85.14 ± 2.09	78.95 ± 4.40	77.17 ± 0.99	80.02 ± 0.94	75.35 ± 1.08
0.75 (selected)	71.24 ± 3.39	81.90 ± 1.64	67.67 ± 3.58	83.73 ± 3.89	85.30 ± 1.88	78.97 ± 4.76	78.24 ± 1.55	81.08 ± 1.06	76.50 ± 1.67
0.90	70.09 ± 4.08	81.08 ± 1.91	66.35 ± 4.44	83.21 ± 4.92	85.10 ± 1.49	78.40 ± 5.64	77.46 ± 1.05	80.08 ± 1.01	75.65 ± 1.14

4.6.3. Memory Bank Size M

As shown in Table 15, $M = 256$ achieves the best overall performance across the three target domains. Smaller memory banks may provide insufficient temporal coverage for stable class-level dispersion estimation, whereas further increasing the cache size to $M = 512$ does not bring additional gains and may reduce the timeliness of cached features. Therefore, $M = 256$ is adopted as a balanced setting in terms of representation stability, feature freshness, and memory cost.

Table 15. Sensitivity analysis of the memory bank size M on three target datasets. The numbers in bold indicate the highest performance metrics.

M	IP			UP			HT		
	OA (%)	AA (%)	Kappa (%)	OA (%)	AA (%)	Kappa (%)	OA (%)	AA (%)	Kappa (%)
64	69.37 ± 4.06	80.53 ± 1.77	65.57 ± 4.27	83.50 ± 4.79	85.25 ± 1.79	78.86 ± 5.53	77.14 ± 1.32	80.03 ± 1.15	75.32 ± 1.42
128	70.52 ± 2.72	81.26 ± 1.37	66.85 ± 2.84	82.79 ± 5.19	84.11 ± 2.87	77.77 ± 6.15	77.14 ± 1.32	80.03 ± 1.15	75.32 ± 1.42
256 (selected)	71.24 ± 3.39	81.90 ± 1.64	67.67 ± 3.58	83.73 ± 3.89	85.30 ± 1.88	78.97 ± 4.76	78.24 ± 1.55	81.08 ± 1.06	76.50 ± 1.67
512	70.75 ± 4.16	81.32 ± 2.68	67.04 ± 4.55	83.35 ± 2.49	85.14 ± 2.20	78.76 ± 3.03	77.17 ± 0.99	80.02 ± 0.94	75.35 ± 1.08

4.6.4. Warm-Up Episodes and Update Interval

As shown in Table 16, the selected temporal setting $E_w = 200$ and $T_{up} = 10$ achieves the best overall performance across all three target domains. Activating clustering too early may use insufficiently trained embeddings, while an overly long warm-up delays the effect of UGCA and shortens the effective multi-center alignment stage. Similarly, overly frequent or sparse prototype refresh can weaken stability. These results confirm that sufficient warm-up and periodic prototype refresh are both important for stable multi-center alignment.

Table 16. Sensitivity analysis of warm-up episodes and update interval on three target datasets. The numbers in bold indicate the highest performance metrics.

E_w T_{up}	IP			UP			HT		
	OA (%)	AA (%)	Kappa (%)	OA (%)	AA (%)	Kappa (%)	OA (%)	AA (%)	Kappa (%)
0 10	70.56 ± 3.28	81.52 ± 1.63	66.88 ± 3.54	83.58 ± 4.91	85.28 ± 2.34	78.80 ± 5.77	77.95 ± 1.19	80.70 ± 0.86	76.18 ± 1.29
100 10	70.96 ± 3.89	81.84 ± 1.67	67.32 ± 4.17	82.87 ± 4.79	84.79 ± 2.29	77.95 ± 5.47	77.20 ± 1.11	79.91 ± 1.17	75.38 ± 1.20
200 10 (selected)	71.24 ± 3.39	81.90 ± 1.64	67.67 ± 3.58	83.73 ± 3.89	85.30 ± 1.88	78.97 ± 4.76	78.24 ± 1.55	81.08 ± 1.06	76.50 ± 1.67
400 10	69.28 ± 2.37	80.84 ± 1.49	65.46 ± 2.58	82.57 ± 4.87	85.23 ± 2.39	77.61 ± 5.57	77.44 ± 1.04	80.33 ± 0.73	75.65 ± 1.11
200 5	70.56 ± 2.97	81.37 ± 1.28	66.86 ± 3.22	83.38 ± 4.10	84.75 ± 2.04	78.46 ± 4.77	77.35 ± 1.41	80.42 ± 1.12	75.55 ± 1.52
200 20	70.19 ± 2.97	81.26 ± 1.31	66.53 ± 3.17	83.63 ± 4.52	85.25 ± 1.52	78.90 ± 5.28	77.99 ± 1.90	80.89 ± 1.73	76.24 ± 2.05

4.6.5. Augmentation-Scale Sensitivity Analysis

Since the target-domain memory bank is constructed from the few labeled target samples and their augmented variants, we further evaluate whether the hard-class estimation is dominated by the augmentation magnitude. Specifically, we vary the radiation-noise parameter β over 0.02, 0.04, and 0.10 on the IP and HT datasets, while keeping the labeled samples, random seeds, network configuration, optimization settings, and all UGCA parameters unchanged. The corresponding OA results and class-wise hard-class selection-frequency correlations are summarized in Table 17.

Table 17. Augmentation-scale sensitivity analysis on IP and HT.

Dataset	$\beta = 0.02$ OA (%)	$\beta = 0.04$ OA (%)	$\beta = 0.10$ OA (%)	Spearman
IP	70.89 $\pm$ 3.49	71.24 $\pm$ 3.39	70.70 $\pm$ 3.22	0.850–0.906
HT	77.44 $\pm$ 1.10	78.24 $\pm$ 1.55	78.00 $\pm$ 1.11	0.839–0.907

The maximum OA variation is only 0.54 percentage points on IP and 0.80 percentage points on HT across a five-fold change in augmentation strength. In addition, the pairwise Spearman correlations of class-wise hard-class selection frequencies remain between 0.839 and 0.907. These results indicate that the hard-class ranking and model performance are not dominated by a particular augmentation magnitude, but mainly reflect persistent relative class difficulty in the learned embedding space.

4.7. Complexity Analysis

To assess the computational efficiency of the proposed method, Table 18 compares its training time, testing time, and parameter count against RN-FSC, DCFSL, Gia-CFSL, and MLPA on the three target domains, where the figures for RN-FSC, DCFSL, Gia-CFSL, and MLPA are taken from the original MLPA paper [31]. It can be observed that the proposed method achieves training times comparable to or lower than those of the prototype-based baselines across all three datasets, while its inference cost remains close to that of MLPA. In terms of model size, the proposed method contains roughly 4.27 M trainable parameters. The gap with respect to the 12.65–12.66 M reported for MLPA mainly reflects implementation-level counting differences rather than backbone capacity: we adopt a trainable-parameters-only convention in which the dual-domain UGCA memory banks, EMA states, cached cluster centers, and the fixed random projection matrices in RandomLayer are excluded as runtime states, and the domain discriminator is instantiated once and reused across the three alignment levels rather than counted as three separate copies. Under this convention, the only parameter overhead our module introduces over the underlying MLPA backbone is the learnable class-center matrix of Center Loss of size $C \times d$, corresponding to a negligible increase of approximately 0.06%. Since UGCA and Center Regularization act only during training and do not alter the inference forward pass, the small testing-time differences in Table 18 are attributable to system-level measurement variance rather than to additional model computation.

Table 18. Complexity comparison. “-” indicates that the corresponding result was not reported.

Dataset	Metrics	Method				
		RN-FSC	DCFSL	Gia-CFSL	MLPA	Ours
IP	Training time	675.21 s	2840.01 s	9382.18 s	2202.40 s	1892.50 s
	Testing time	45.14 s	1.93 s	1.06 s	1.01 s	1.91 s
	#Params	0.19 M	4.26 M	0.86 M	12.66 M	4.27 M
UP	Training time	411.34 s	1534.98 s	4001.86 s	1849.13 s	1533.59 s
	Testing time	47.23 s	6.52 s	5.33 s	7.65 s	8.58 s
	#Params	0.19 M	4.25 M	0.86 M	12.65 M	4.26 M
HT	Training time	-	5851.67 s	5952.41 s	3347.71 s	1579.20 s
	Testing time	-	4.34 s	1.51 s	2.59 s	2.66 s
	#Params	-	4.25 M	0.86 M	12.65 M	4.27 M

5. Discussion

The key to the effectiveness of UGCA is that it does not assume that all classes require multi-center prototypes. Instead, it turns the question of whether multi-center modeling is worthwhile into a hard-class detection problem driven by the target-domain training

cache. The fixed-length memory bank reduces the noise of a single episode, and EMA smoothing further suppresses random fluctuations in the class-level uncertainty proxy. The selected hard classes are then assigned more fine-grained clustered prototypes, so their deep alignment no longer depends on only a single mean point. In the current implementation, hard classes are split into multiple equally weighted clustered units, as defined in Section 3.2.6. This design keeps the total adversarial contribution of a hard class comparable to that of a non-hard class while allowing the deep alignment stage to model more detailed local structure. In essence, the contribution of UGCA lies in upgrading the original single-center deep alignment that treats all classes equally into an adaptive mixed representation. Complex classes obtain more precise cross-domain alignment units, whereas simple classes retain the original efficient single-center representation. In this way, overall classification performance is improved without introducing excessive overhead.

Under the few-shot setting, instability in category representations does not arise solely from class complexity itself. To a large extent, it is caused by sampling noise, outliers, and short-term fluctuations induced by alternating adversarial training. In the current implementation, Center Loss is not intended to model the local structure of hard classes. Instead, it acts as a lightweight global compactness regularizer that directly constrains episodic query embeddings. As discussed in Section 3.3, the complementary roles of UGCA and Center Loss help explain why the joint model performs more consistently than either individual module. Alternative discriminative objectives such as Ring Loss [35] and ArcFace [36] enforce different geometric constraints on the feature space; we adopt Center Loss here because of its simplicity and direct compatibility with prototype-based classifiers.

This paper explicitly constrains both the training setup and the evaluation protocol. The target-domain memory bank is updated only with D_f and its augmented samples, as specified in Section 3.2.2, and neither D_u nor its labels are involved in training. The evaluation protocol follows the original MLPA paper, as detailed in Section 4.2. The proposed method nonetheless has several limitations. First, the target-domain memory bank is essentially a temporal buffer formed from the labeled training subset and its augmentations; it does not constitute a sufficient statistical description of the true target-domain population distribution. Second, the current implementation uses uniformly weighted clustered alignment and standard global Center Loss. More sophisticated cluster-size-aware weighting and cluster-aware compactness are not implemented and therefore are not claimed as current contributions of this paper. Third, the sensitivity analysis in Section 4.6 shows that the proposed method is reasonably robust around the default parameter settings, but its performance still degrades under extreme settings, such as $K = 4$ or $q = 0.50$.

6. Conclusions

This paper addresses prototype instability in cross-domain few-shot hyperspectral image classification and proposes a plug-and-play prototype-stability enhancement module by integrating UGCA and Center Regularization into prototype-driven CD-FSL frameworks. UGCA is guided by a class-level uncertainty proxy computed from a target-domain training cache; hard classes identified by a quantile-plus-Top- M criterion are modeled with K-means-generated multi-center alignment units that participate in deep adversarial alignment with uniform weights, while easy classes retain the original single-prototype alignment. Center Regularization is applied to episodic query embeddings as a lightweight global compactness term that mitigates representation drift under sparse supervision. Both components act only at training time, so the metric-based inference path of the host framework is preserved. On the three Chikusei-to-IP/UP/HT cross-domain tasks, the module yields OA gains of +2.03%, +1.83%, and +1.32% over the MLPA baseline, with comparable improvements on AA and Kappa (full numbers are reported in Tables 3–6); inserting the

same module into Gia-CFSL brings further OA gains of +2.55%, +0.90%, and +1.24% on IP, UP, and HT, indicating that the enhancement transfers across prototype-based CD-FSL hosts, and sensitivity analyses on K , q , M , E_w , and T_{up} further indicate that the method is robust within a reasonable range around the default configuration. Future work will focus on adaptive prototype construction, density-aware alignment mechanisms, and noise-aware hard-class diagnosis to better distinguish intrinsic class complexity from label noise or poor annotations. We also plan to extend the proposed framework to transformer-based CD-FSL architectures across broader shot settings.

Author Contributions: Conceptualization, Q.W. and M.L.; methodology, Q.W. and S.D.; software, Q.W.; validation, Q.W. and L.W.; formal analysis, Q.W.; investigation, Q.W. and M.L.; resources, S.D. and L.W.; data curation, Q.W.; writing—original draft preparation, Q.W.; writing—review and editing, M.L. and S.D.; visualization, Q.W. and M.L.; supervision, S.D.; project administration, S.D.; funding acquisition, L.W. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Data Availability Statement: The benchmark hyperspectral datasets used in this study are publicly available at http://www.ehu.eus/ccwintco/index.php/Hyperspectral_Remote_Sensing_Scenes (accessed on 1 July 2026).

Acknowledgments: The authors thank the editors and reviewers for their valuable comments and suggestions.

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

References

- Landgrebe, D. Hyperspectral image data analysis. *IEEE Signal Process. Mag.* **2002**, *19*, 17–28. [\[CrossRef\]](#) [\[CrossRef\]](#)
- Vivone, G. Multispectral and hyperspectral image fusion in remote sensing: A survey. *Inf. Fusion* **2023**, *89*, 405–417. [\[CrossRef\]](#) [\[CrossRef\]](#)
- Melgani, F.; Bruzzone, L. Classification of hyperspectral remote sensing images with support vector machines. *IEEE Trans. Geosci. Remote Sens.* **2004**, *42*, 1778–1790. [\[CrossRef\]](#) [\[CrossRef\]](#)
- Li, Y.; Zhang, H.; Xue, X.; Jiang, Y.; Shen, Q. Deep learning for remote sensing image classification: A survey. *Wiley Interdiscip. Rev. Data Min. Knowl. Discov.* **2018**, *8*, e1264. [\[CrossRef\]](#) [\[CrossRef\]](#)
- Chen, Y.; Lin, Z.; Zhao, X.; Wang, G.; Gu, Y. Deep learning-based classification of hyperspectral data. *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.* **2014**, *7*, 2094–2107. [\[CrossRef\]](#) [\[CrossRef\]](#)
- Hu, W.; Huang, Y.; Wei, L.; Zhang, F.; Li, H. Deep convolutional neural networks for hyperspectral image classification. *J. Sens.* **2015**, *2015*, 258619. [\[CrossRef\]](#) [\[CrossRef\]](#)
- 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.* **2018**, *56*, 847–858. [\[CrossRef\]](#) [\[CrossRef\]](#)
- Lee, H.; Kwon, H. Going deeper with contextual CNN for hyperspectral image classification. *IEEE Trans. Image Process.* **2017**, *26*, 4843–4855. [\[CrossRef\]](#) [\[CrossRef\]](#) [\[PubMed\]](#)
- Sun, L.; Zhao, G.; Zheng, Y.; Wu, Z. Spectral-spatial feature tokenization transformer for hyperspectral image classification. *IEEE Trans. Geosci. Remote Sens.* **2022**, *60*, 5522214. [\[CrossRef\]](#) [\[CrossRef\]](#)
- Hong, D.; Han, Z.; Yao, J.; Gao, L.; Zhang, B.; Plaza, A.; Chanussot, J. SpectralFormer: Rethinking hyperspectral image classification with transformers. *IEEE Trans. Geosci. Remote Sens.* **2022**, *60*, 5518615. [\[CrossRef\]](#) [\[CrossRef\]](#)
- Jiang, Y.; Li, Y.; Zhang, H. Hyperspectral image classification based on 3-D separable ResNet and transfer learning. *IEEE Geosci. Remote Sens. Lett.* **2019**, *16*, 1949–1953. [\[CrossRef\]](#) [\[CrossRef\]](#)
- Peng, J.; Sun, W.; Ma, L.; Du, Q. Discriminative transfer joint matching for domain adaptation in hyperspectral image classification. *IEEE Geosci. Remote Sens. Lett.* **2019**, *16*, 972–976. [\[CrossRef\]](#) [\[CrossRef\]](#)
- Huang, Y.; Peng, J.; Zhang, G.; Sun, W.; Chen, N.; Du, Q. Adversarial domain adaptation network with calibrated prototype and dynamic instance convolution for hyperspectral image classification. *IEEE Trans. Geosci. Remote Sens.* **2024**, *62*, 5514613. [\[CrossRef\]](#) [\[CrossRef\]](#)
- Pan, E.; Ma, Y.; Fan, F.; Mei, X.; Huang, J. Hyperspectral image classification across different datasets: A generalization to unseen categories. *Remote Sens.* **2021**, *13*, 1672. [\[CrossRef\]](#) [\[CrossRef\]](#)

15. Dong, G.; Zhao, C.; Pan, X.; Basu, A. Learning Temporal Distribution and Spatial Correlation Toward Universal Moving Object Segmentation. *IEEE Trans. Image Process.* **2024**, *33*, 2447–2461. [\[CrossRef\]](#) [\[CrossRef\]](#) [\[PubMed\]](#)
16. Snell, J.; Swersky, K.; Zemel, R. Prototypical networks for few-shot learning. In *Advances in Neural Information Processing Systems 30 (NeurIPS 2017)*; Guyon, I., Luxburg, U.V., Bengio, S., Wallach, H., Fergus, R., Vishwanathan, S., Garnett, R., Eds.; Curran Associates: Red Hook, NY, USA, 2017; pp. 4077–4087.
17. Sung, F.; Yang, Y.; Zhang, L.; Xiang, T.; Torr, P.; Hospedales, T. Learning to compare: Relation network for few-shot learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*; IEEE: Piscataway, NJ, USA, 2018; pp. 1199–1208. [\[CrossRef\]](#)
18. Sun, Q.; Liu, Y.; Chua, T.-S.; Schiele, B. Meta-transfer learning for few-shot learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*; IEEE: Piscataway, NJ, USA, 2019; pp. 403–412. [\[CrossRef\]](#)
19. 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.* **2019**, *57*, 2290–2304. [\[CrossRef\]](#) [\[CrossRef\]](#)
20. 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. [\[CrossRef\]](#) [\[CrossRef\]](#)
21. Tsoumplekas, G.; Li, V.; Sarigiannidis, P.; Argyriou, V. A Complete Survey on Contemporary Methods, Emerging Paradigms and Hybrid Approaches for Few-Shot Learning. *arXiv* **2024**, arXiv:2402.03017.
22. Zhao, Y.; Zhang, T.; Li, J.; Tian, Y. Dual Adaptive Representation Alignment for Cross-Domain Few-Shot Learning. *IEEE Trans. Pattern Anal. Mach. Intell.* **2023**, *45*, 11720–11732. [\[CrossRef\]](#) [\[CrossRef\]](#) [\[PubMed\]](#)
23. Li, Z.; Liu, M.; Chen, Y.; Xu, Y.; Li, W.; Du, Q. Deep Cross-Domain Few-Shot Learning for Hyperspectral Image Classification. *IEEE Trans. Geosci. Remote Sens.* **2022**, *60*, 5501618. [\[CrossRef\]](#) [\[CrossRef\]](#)
24. 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.* **2024**, *35*, 1912–1925. [\[CrossRef\]](#) [\[CrossRef\]](#) [\[PubMed\]](#)
25. Xu, Y.; Zhang, Y.; Yue, T.; Yu, C.; Li, H. Graph-based domain adaptation few-shot learning for hyperspectral image classification. *Remote Sens.* **2023**, *15*, 1125. [\[CrossRef\]](#) [\[CrossRef\]](#)
26. Peng, Y.; Liu, Y.; Tu, B.; Zhang, Y. Convolutional transformer-based few-shot learning for cross-domain hyperspectral image classification. *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.* **2023**, *16*, 1335–1349. [\[CrossRef\]](#) [\[CrossRef\]](#)
27. Wang, Z.; Zhao, S.; Zhao, G.; Song, X. Dual-branch domain adaptation few-shot learning for hyperspectral image classification. *IEEE Trans. Geosci. Remote Sens.* **2024**, *62*, 5506116. [\[CrossRef\]](#) [\[CrossRef\]](#)
28. Zhang, Z.; Gao, D.; Liu, D.; Shi, G. Spectral-spatial domain attention network for hyperspectral image few-shot classification. *Remote Sens.* **2024**, *16*, 592. [\[CrossRef\]](#) [\[CrossRef\]](#)
29. Li, J.; Zhang, Z.; Song, R.; Li, Y.; Du, Q. SCFormer: Spectral coordinate transformer for cross-domain few-shot hyperspectral image classification. *IEEE Trans. Image Process.* **2024**, *33*, 840–855. [\[CrossRef\]](#) [\[CrossRef\]](#) [\[PubMed\]](#)
30. Tang, H.; Yang, X.; Tang, D.; Dong, Y.; Zhang, L.; Xie, W. Tensor-based few-shot learning for cross-domain hyperspectral image classification. *Remote Sens.* **2024**, *16*, 4149. [\[CrossRef\]](#) [\[CrossRef\]](#)
31. Liu, H.; He, J.; Li, Y.; Bi, Y. Multilevel prototype alignment for cross-domain few-shot hyperspectral image classification. *IEEE Trans. Geosci. Remote Sens.* **2025**, *63*, 4400115. [\[CrossRef\]](#) [\[CrossRef\]](#)
32. Allen, K.R.; Shelhamer, E.; Shin, H.; Tenenbaum, J.B. Infinite Mixture Prototypes for Few-shot Learning. In *Proceedings of the 36th International Conference on Machine Learning*; PMLR: Long Beach, CA, USA, 2019; pp. 232–241.
33. Chen, W.; Zhang, Y.; Chu, J.; Wang, X. Domain-invariant few-shot contrastive learning for hyperspectral image classification. *Appl. Sci.* **2024**, *14*, 11053. [\[CrossRef\]](#) [\[CrossRef\]](#)
34. Yokoya, N.; Iwasaki, A. *Airborne Hyperspectral Data over Chikusei*; Tech. Rep. SAL-2016-05-27; Space Application Laboratory, The University of Tokyo: Tokyo, Japan, 2016.
35. Zheng, Y.; Pal, D.K.; Savvides, M. Ring loss: Convex feature normalization for face recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*; IEEE: Piscataway, NJ, USA, 2018; pp. 5089–5097. [\[CrossRef\]](#)
36. Deng, J.; Guo, J.; Xue, N.; Zafeiriou, S. ArcFace: Additive Angular Margin Loss for Deep Face Recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*; IEEE: Piscataway, NJ, USA, 2019; pp. 4685–4694. [\[CrossRef\]](#)

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.