

Article

Fed-RSAdapter: Federated Fine-Tuning of Remote Sensing Images via Multi-Scale Adapter Modules

Yuelel Wang ¹, Liangkui Lin ^{1,*}, Yirui Zhou ¹, Shaolin Wang ¹, Zhirui Wang ² and Xiyu Qi ²

¹ Shanghai Institute of Satellite Engineering, Shanghai 201109, China; wangyuelel20@mailsucas.ac.cn (Y.W.); zyr050798@gmail.com (Y.Z.); shaolinwa@126.com (S.W.)

² Aerospace Information Research Institute, Chinese Academy of Sciences, Beijing 100094, China; zhirui1990@126.com (Z.W.); qixiyu20@mailsucas.ac.cn (X.Q.)

* Correspondence: gxckk1980@sina.com

Highlights

What are the main findings?

- A multi-scale adapter module with cross-layer dense connections is proposed for federated fine-tuning of remote sensing images, achieving competitive interpretation performance with only 4.93% trainable parameters relative to the full backbone.
- A parameter similarity-aware aggregation strategy based on Gaussian kernel similarity is designed, which constructs a similarity matrix among client adapter parameters to guide personalized weighted aggregation, effectively balancing global generalization and client-specific adaptation under Non-IID data distributions.

What are the implications of the main findings?

- The proposed Fed-RSAdapter framework demonstrates communication-efficient federated fine-tuning for remote sensing models, reducing transmitted trainable parameters to less than 5% of conventional full-parameter federated learning while achieving an average performance gain of 3.75% OA in scene classification, 0.74% in mAP for object detection, and 0.25% OA for semantic segmentation over full-parameter federated learning baselines.
- The similarity-aware personalized aggregation mechanism provides a generalizable solution for handling data heterogeneity in distributed remote sensing platforms, with broad applicability to satellite, aerial, and UAV-based edge computing scenarios across classification, object detection, and semantic segmentation tasks.

Abstract

In edge computing scenarios for remote sensing image interpretation, two fundamental challenges constrain the effectiveness of federated fine-tuning: the limited computational capacity of edge devices restricts the multi-scale feature learning capability of lightweight deployed models, while the highly heterogeneous and imbalanced data distributions (Non-IID settings) across clients render conventional parameter aggregation strategies ineffective. To address these challenges jointly, this paper proposes Fed-RSAdapter, a federated fine-tuning framework for remote sensing imagery based on multi-scale adapter modules. On the edge side, a multi-scale adapter architecture with cross-layer dense connections and feature map concatenation alignment is introduced. By inserting lightweight adapter modules in parallel within a frozen backbone, the proposed design captures multi-scale spatial characteristics inherent in high-resolution remote sensing imagery while confining trainable parameters to less than 5% of the full model, thereby satisfying strict on-device computational and communication constraints. On the server side, a parameter similarity-aware aggregation strategy is designed to handle client heterogeneity. Client



Academic Editor: Zhenwei Shi

Received: 25 May 2026

Revised: 26 June 2026

Accepted: 9 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.

adapter parameters are normalized and mapped into a similarity matrix via Gaussian kernel distance, enabling the server to perform personalized weighted aggregation that balances global generalization with client-specific adaptation. Extensive experiments on scene classification, object detection, and semantic segmentation benchmarks demonstrate that Fed-RSAdapter achieves an average improvement of 3.75% in overall accuracy for scene classification, 0.74% in mAP for object detection, and 0.25% in overall accuracy for semantic segmentation over federated baselines, while reducing the volume of transmitted parameters to approximately 4.93% of that required by conventional full-parameter federated learning. These results demonstrate the communication-efficient fine-tuning capability of the proposed framework under federated remote sensing settings. We further clarify that actual deployment on embedded edge hardware is not directly evaluated in this work and is therefore treated as an important direction for future validation rather than an experimentally proven conclusion.

Keywords: federated learning; remote sensing image interpretation; parameter-efficient fine-tuning; multi-scale modeling; edge computing

1. Introduction

In recent years, the rapid development of satellite and airborne platforms has generated a great volume of high-resolution remote sensing imagery, causing growing demand for interpretation models that can operate directly on resource-constrained edge devices [1,2]. While large-scale pretrained vision models have demonstrated remarkable feature learning capability, deploying and fine-tuning them on edge terminals with limited memory and computation remains a fundamental challenge. Adapter-based parameter-efficient fine-tuning (PEFT) methods have emerged as a promising solution to this problem [3–6]. By freezing the parameters of a pretrained backbone and updating only a small number of inserted adapter layers, these methods preserve the original generalization capability of pretrained models while enabling rapid task-specific adaptation, significantly reducing both computational cost and memory consumption.

Within the broader context of PEFT, adapter-based methods have been extensively studied alongside other paradigms such as prompt tuning and parameter tuning [7]. Early studies demonstrated that task-specific parameters can be reduced to less than 5% of the original model size while maintaining competitive performance [8,9]. In the vision domain, AdaptFormer [4] integrates adapter modules into the MLP bypass of Swin Transformer [10] blocks, effectively capturing both local and global contextual information. More recently, Low-Rank Adaptation (LoRA) [11] and its extensions [12,13] have gained popularity by injecting trainable low-rank matrices into Transformer layers [14], further reducing training parameters. In remote sensing applications, adapter-based fine-tuning has proven particularly effective due to the strong domain specificity and scale diversity of remote sensing imagery. Representative works include ACTNet [15] for semantic segmentation, AiRs [3] for remote sensing foundation models, and RSAdapter [6] for multimodal visual question answering. These methods demonstrate that PEFT provides an effective means for adapting large vision models to remote sensing tasks under limited computational resources.

Despite this progress, two critical and intertwined challenges arise when adapter-based fine-tuning is deployed in distributed edge computing scenarios for remote sensing, where multiple satellite or aerial platforms independently collect and process heterogeneous imagery.

Challenge 1: Insufficient multi-scale feature learning under edge constraints. In remote sensing tasks, high-resolution imagery inherently contains rich multi-scale spatial patterns. The conventional adapter layers struggle to capture features when embedded within compact lightweight backbones [16,17] from fine-grained object textures to large-scale scene structures. Existing PEFT methods are mainly designed for large-scale foundation models with abundant parameters [18,19], and their direct application to lightweight architectures on edge devices leads to degraded feature learning due to insufficient model capacity. This limitation is particularly significant in federated settings, where each edge device independently fine-tunes a local model on limited local samples, further restricting the diversity of feature learning.

Challenge 2: Ineffective parameter aggregation under Non-IID distributions. Federated learning enables collaborative model training across distributed clients without sharing raw data [20], making it particularly suited for remote sensing platforms where data privacy and communication constraints are critical [21]. However, since only adapter layers are updated while the backbone remains frozen, conventional full-model transmission protocols such as FedAvg [22] waste the vast majority of communication bandwidth on frozen parameters that carry no update signal. In addition, the highly Non-IID data distributions across remote sensing platforms render uniform parameter averaging across clients suboptimal. Clients with divergent data distributions should contribute unequal aggregation weights rather than being treated indiscriminately.

Existing federated methods have partially addressed these issues in isolation. FedProx [23] introduces proximal regularization to constrain local updates, while SCAFFOLD [24] employs control variates to reduce client drift. Personalized approaches such as pFedMe [25] and FedRep [26] aim to balance global and client-specific adaptation. In the remote sensing domain, FedPM [27], DCM [28], and PAD [29] have demonstrated the feasibility of federated learning for remote sensing tasks. However, these methods focus primarily on full-model fine-tuning or conventional aggregation paradigms, and do not explicitly address the joint challenge of multi-scale feature modeling under edge constraints and similarity-aware aggregation for adapter-only updates. Communication-reduction techniques such as gradient sparsification [30] and quantized encoding [31] reduce transmission cost but do not resolve the aggregation inefficiency caused by frozen backbone parameters. While prior works have addressed multi-scale feature modeling and personalized federated aggregation separately, their joint design within a parameter-efficient fine-tuning framework for edge-deployed remote sensing models remains largely unexplored.

Motivated by these observations, this paper proposes **Fed-RSAdapter**, a federated fine-tuning framework for remote sensing imagery that jointly addresses both challenges through a unified design. On the edge side, a multi-scale adapter architecture with cross-layer dense connections and feature map concatenation alignment is proposed to enhance multi-scale feature learning within a lightweight frozen backbone, confining trainable parameters to less than 5% of the full model. On the server side, adapter-only gradient transmission reduces communication cost, eliminating the structural mismatch caused by frozen backbone transmission. Simultaneously, a parameter similarity-aware aggregation strategy is designed, in which client adapter parameters are normalized and mapped into a Gaussian kernel similarity matrix to guide personalized weighted aggregation, enabling each edge model to balance global generalization with client-specific specialization under heterogeneous data distributions.

The main contributions of this paper are summarized as follows:

1. We propose a multi-scale adapter module with cross-layer dense connections and feature map concatenation alignment for lightweight remote sensing backbones. The proposed design effectively captures multi-scale spatial characteristics of high-resolution

- remote sensing imagery while confining trainable parameters to less than 5% of the backbone, providing a communication-efficient basis for federated fine-tuning.
2. We design a communication-efficient and heterogeneity-aware server-side aggregation framework comprising two components. By transmitting only adapter gradients, communication cost is reduced to resolve the structural mismatch caused by frozen backbone transmission. Then, a parameter similarity-aware aggregation strategy constructs a Gaussian kernel similarity matrix to enable personalized weighted aggregation that explicitly accounts for Non-IID data distributions.
 3. Extensive experiments on five benchmark datasets across three remote sensing interpretation tasks, scene classification (UCM, AID, NWPU), object detection (DIOR), and semantic segmentation (Potsdam), demonstrate that Fed-RSAdapter consistently outperforms both general federated learning baselines and adapter-based fine-tuning methods, achieving an average OA improvement of 3.75% in classification, 0.74% in mAP for detection, and 0.25% in OA for segmentation, while reducing transmitted parameter volume to approximately 4.93% of full-model federated learning.

The remainder of this paper is organized as follows. Section 2 presents the proposed methodology, including the overall federated fine-tuning framework, the multi-scale adapter module with cross-layer connections, the parameter similarity-aware aggregation strategy, and the loss function design for balancing heterogeneous clients. Section 3 reports experimental results on remote sensing image classification, object detection, and semantic segmentation tasks. Section 4 provides comprehensive analysis and discussion of the experimental results. Finally, Section 5 concludes the paper.

2. Materials and Methods

2.1. Overall Framework of Fed-RSAdapter

This paper proposes Fed-RSAdapter, a federated fine-tuning method based on multi-scale adapters for remote sensing image interpretation. The overall framework is illustrated in Figure 1. The proposed method performs adapter-based fine-tuning on edge devices and similarity-aware parameter aggregation on the central server, enabling efficient model adaptation under limited computational resources.

On the edge side, a CNN-Transformer-based architecture is adopted as the backbone network. Each edge device performs local training and parameter updates using its own data samples and a shared pretrained model. To reduce computational cost while enhancing task-specific adaptability, all backbone CNN-Transformer parameters are frozen during fine-tuning. Although these frozen parameters still participate in forward propagation, they are excluded from gradient backpropagation, thereby preserving the stability of the backbone's fundamental feature extraction capability.

Only the parameters of the newly introduced adapter modules are involved in gradient updates. Since the number of adapter parameters is relatively small, the overall computational cost of fine-tuning is significantly reduced, making the proposed approach well suited for resource-constrained edge environments. In addition, each adapter module is connected to the backbone network via residual connections, enabling effective interaction between adapter features and backbone representations. This design not only facilitates information fusion across layers but also ensures that the expressive power of the model is not compromised.

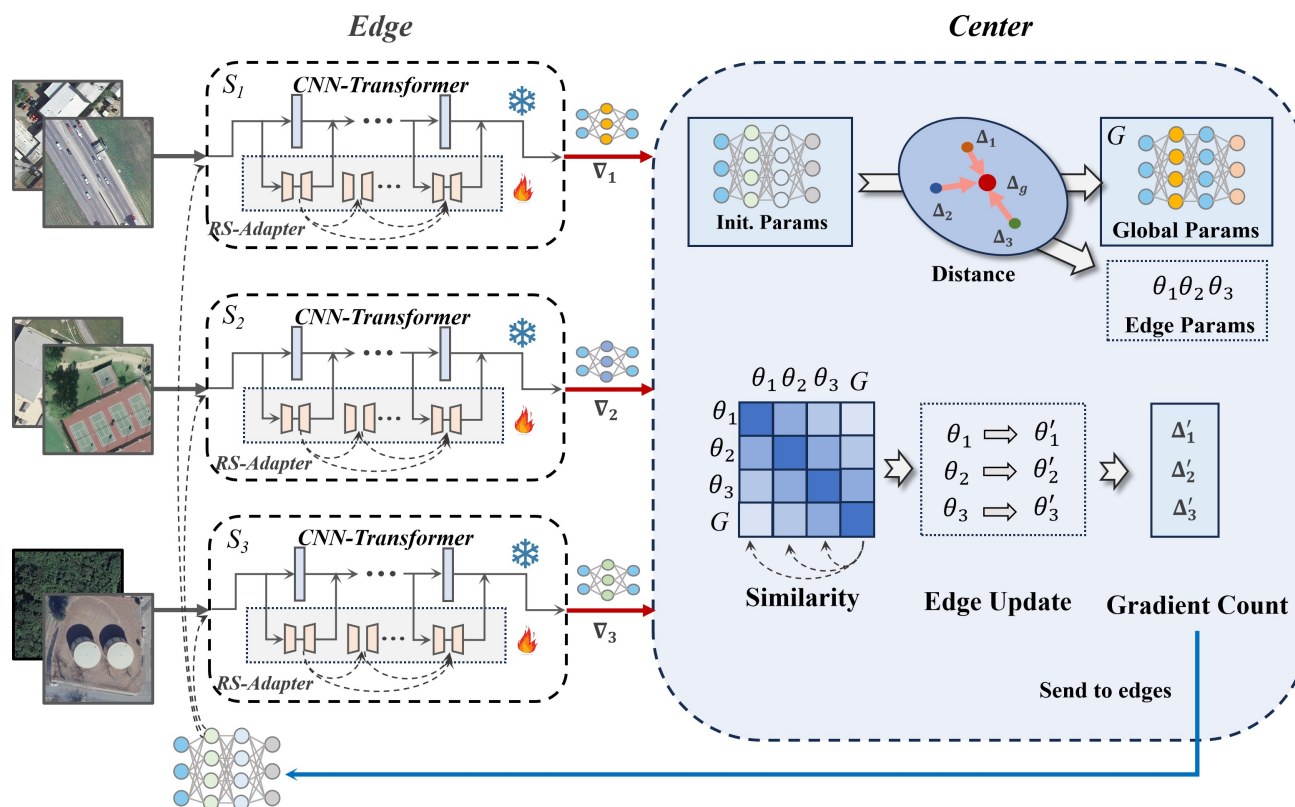


Figure 1. Overall framework of the federated learning parameter fine-tuning method for remote sensing (Fed-RSAdapter).

After local training on edge devices, the updated parameters of the adapter modules are transmitted to the central server. The server measures the similarity between adapter parameters from different clients by computing the distances among their parameter representations. Based on this similarity information, the central model selectively aggregates fine-tuned parameters from participating edge devices, assigning appropriate weights during the fusion process.

This distance-based parameter aggregation strategy effectively exploits the characteristics of client-side data distributions while mitigating the negative impact of irrelevant or highly divergent updates. As a result, the overall performance of federated fine-tuning is significantly improved. The aggregated adapter parameters are then redistributed to edge devices to initialize the next round of local fine-tuning, leading to continuous and stable optimization of the global model across federated training rounds.

2.2. Multi-Scale Adapter Fine-Tuning Module

The parameter fine-tuning module deployed on edge intelligent terminals consists of multiple adapter modules. Unlike conventional full-parameter fine-tuning methods that require updating the entire backbone network, the proposed RSAdapter performs task-specific adaptation through a lightweight bottleneck architecture, significantly reducing the number of trainable parameters while preserving the prior knowledge of the pretrained model.

As illustrated in Figure 2, each RSAdapter is inserted into the backbone network as an independent lightweight adaptation block. Given an input feature map $x \in \mathbb{R}^{H \times W \times C}$, the feature is first normalized through a Layer Normalization (LN) operation to stabilize feature distributions. Subsequently, a 1×1 convolution layer is employed to project the high-dimensional feature learning into a compact latent space:

During the fine-tuning process, the pretrained backbone parameters W remain frozen to ensure that the prior knowledge encoded in the original model is preserved. Fine-tuning is performed exclusively within the adapter modules. The update process can be formulated as

$$x_d = W_d x, \quad (1)$$

where $W_d \in \mathbb{R}^{C/r \times C}$ denotes the channel compression matrix and r represents the reduction ratio.

After dimensionality reduction, a GELU activation function is introduced to enhance nonlinear representation capability. To better capture local spatial dependencies in remote sensing imagery, a depthwise 3×3 convolution is further applied in the compressed feature space:

$$x_s = \text{DWConv}(\text{GELU}(x_d)), \quad (2)$$

where the depthwise convolution introduces spatial inductive bias while maintaining low computational complexity.

The compressed spatial features are then projected back to the original feature dimension through an additional 1×1 convolution layer:

$$x_u = W_u x_s, \quad (3)$$

where $W_u \in \mathbb{R}^{C \times C/r}$ denotes the feature reconstruction matrix that projects the bottleneck feature from C/r channels back to the original C channels.

Finally, a residual connection is adopted to preserve the original semantic representation and improve gradient propagation efficiency:

$$h = x + x_u, \quad (4)$$

where h denotes the final output feature.

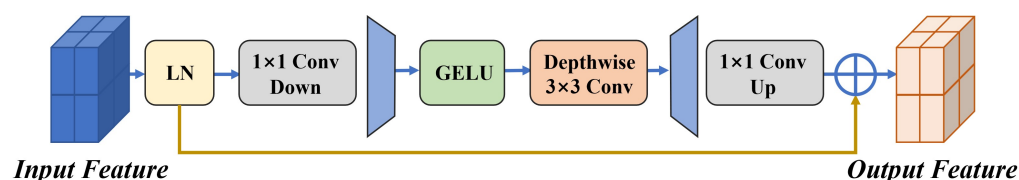


Figure 2. Details of the RSAAdapter for the fine-tuning method.

During the entire fine-tuning process, the backbone parameters W remain frozen, and only the parameters within the RSAAdapter are updated. This strategy avoids direct optimization of high-dimensional backbone parameters and significantly reduces memory consumption, computational cost, and communication costs in distributed edge environments.

The number of trainable parameters in a standard full fine-tuning layer is approximately C^2 , while the proposed RSAAdapter only introduces:

$$P_{\text{adapter}} = 2 \frac{C^2}{r} + 9 \frac{C}{r}, \quad (5)$$

trainable parameters, where the first term corresponds to the two 1×1 convolution layers and the second term corresponds to the depthwise convolution operation.

Therefore, the parameter reduction ratio can be formulated as

$$\rho = \frac{P_{\text{adapter}}}{C^2} = \frac{2}{r} + \frac{9}{rC}, \quad R = 1 - \rho, \quad (6)$$

where ρ denotes the trainable parameter ratio of the adapter relative to a full $C \times C$ projection and R denotes the corresponding proportion of reduced trainable parameters. The second term $\frac{9}{rC}$ originates from the depthwise 3×3 convolution in the bottleneck space.

In practical implementations, the channel dimension C typically takes values such as 96, 192, 384, or 768 in hierarchical vision backbones, while the reduction ratio r is selected according to task requirements. This lightweight design enables efficient fine-tuning on resource-constrained edge devices while enhancing spatial feature adaptation for remote sensing imagery.

To further enhance the modeling of multi-scale characteristics in remote sensing imagery, a carefully designed multi-scale adapter connection scheme is introduced for edge intelligent terminals, as illustrated in Figure 3. The overall architecture adopts the CNN-Transformer hybrid attention module of RingMo-lite [2], enabling simultaneous capture of global semantic information and local structural details. RingMo-lite is selected as the backbone because it is a lightweight remote-sensing-oriented CNN-Transformer hybrid model pretrained on large-scale remote sensing imagery. Its combination of local convolutional inductive bias and global Transformer modeling is consistent with the multi-scale characteristics of aerial and satellite images, and its compact parameter scale enables a fair evaluation of parameter-efficient federated fine-tuning across classification, detection, and segmentation tasks. Through densely connected multi-scale links, the proposed design enables effective feature interaction across different layers and resolutions.

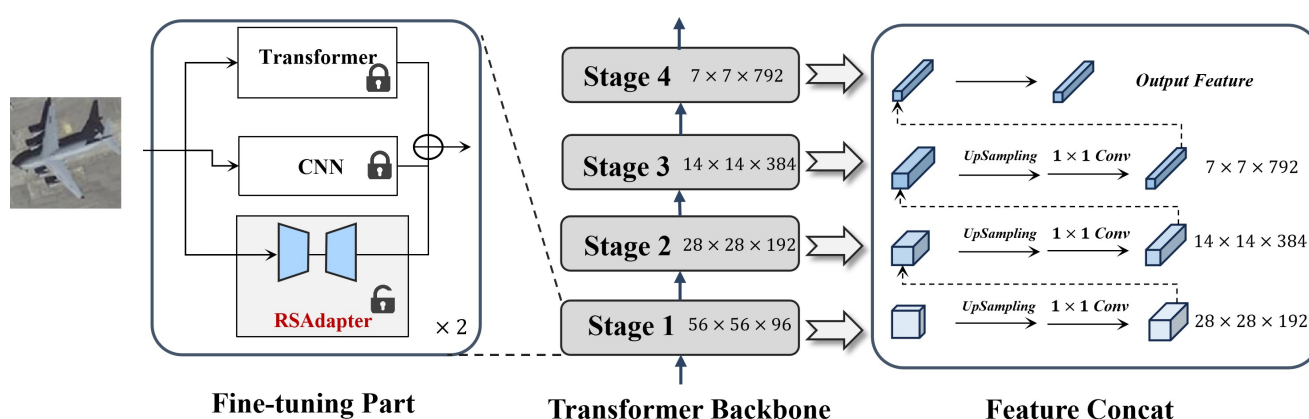


Figure 3. Structure of the multi-scale adapter method.

Since feature maps from different layers vary in spatial resolution and channel dimensions, compatibility during fusion is ensured through an upsampling and channel alignment strategy. Specifically, the feature map F_i from a lower-resolution layer is first upsampled to a higher resolution and then passed through a 1×1 convolution for channel alignment. The transformed feature map F'_i is computed as

$$F'_i = \text{Conv}_{1 \times 1}(\text{Upsampling}(F_i)), \quad (7)$$

where the upsampling operation increases the spatial resolution, and the 1×1 convolution adjusts the channel dimension to facilitate subsequent feature concatenation. Feature fusion within each stage is then performed as

$$F_i = F'_{i-1} \oplus F_{T,i-1} \oplus F_{C,i-1}, \quad (8)$$

where F'_{i-1} denotes the adapter output from the $(i-1)$ -th layer after upsampling and channel alignment, and $F_{T,i-1}$ and $F_{C,i-1}$ represent the outputs of the frozen Transformer and CNN branches at the same layer, respectively. This design enables efficient fusion of

features from different branches and multiple scales, significantly enhancing the representational capacity and robustness of the model.

In terms of parameter efficiency, freezing the CNN-Transformer backbone ensures that only the parameters of the RSAdapter modules participate in training. At each stage, the number of trainable parameters in the adapter modules is given by $2\alpha nd^2$, where n denotes the number of adapter modules. For example, in the first stage with $d = 96$, compression ratio $\alpha = 0.25$, and $n = 2$, the number of trainable parameters is $2 \times 0.25 \times 2 \times 96^2 = 9216$. The parameter counts for subsequent stages can be computed in a similar manner.

Overall, the proposed design introduces approximately 1.08 M additional trainable parameters, accounting for only 3.82% of the original Swin Transformer Tiny backbone (28.3 M parameters). After incorporating dense multi-scale connections, the additional upsampling and 1×1 convolution layers introduce a small number of extra parameters, increasing the total number of trainable parameters to 1.467 M, which corresponds to approximately 4.93% of the backbone network. This multi-scale adapter connection strategy not only ensures effective fusion of features across different scales but also substantially reduces the number of parameters and computational cost required for training, making it particularly suitable for efficient and low-cost fine-tuning on edge devices with limited data.

While the RSAdapter module achieves efficient parameter reduction and local spatial adaptation, a single adapter inserted at each backbone stage operates independently and lacks explicit interaction across different spatial scales. To address this limitation and fully exploit the multi-scale characteristics of high-resolution remote sensing imagery, a cross-layer dense connection scheme is further introduced in Section 2.3, enabling feature fusion across adapter stages at different resolutions.

2.3. Parameter Similarity Sharing Module

In conventional federated learning, each client is required to upload the entire set of model parameters to the central server for global aggregation, resulting in a unified global model parameter denoted as θ_{global} . In the proposed method, however, the backbone network parameters are fixed during the pretraining stage and remain frozen throughout the fine-tuning process, while only the adapter modules are updated at each client. Since the backbone parameters are identical across all clients and the central server and do not change during training, they do not need to be repeatedly transmitted.

To further improve communication efficiency, each client only computes the gradients of the adapter parameters, defined as the difference between the current adapter parameters and the initial global adapter parameters, denoted as $\Delta\theta^{(i)}$. Only these gradient updates are transmitted to the central server. The Euclidean distance [32] between local gradients and the global gradient is utilized to determine the update direction of the global parameters, which is then used to compute the global gradient. This strategy not only avoids redundant data transmission but also significantly reduces communication cost, thereby improving the efficiency of parameter transmission in federated fine-tuning.

Due to the substantial heterogeneity in data distributions across different edge devices, directly broadcasting a unified global model to all clients may degrade local performance and hinder adaptation to local data characteristics. To address this issue, a parameter similarity sharing mechanism is introduced. By computing the similarity between the parameters of edge devices and the central server, parameters that are more consistent with a given client are selectively aggregated and transformed into corresponding gradients to assist local model updates.

First, to eliminate scale discrepancies among parameters from different edge devices, we only use the trainable adapter parameters for similarity estimation. The local adapter parameter vector of client i at communication round t is denoted as $\theta_{a,i}^t$, while the frozen

backbone parameters θ_b are shared by all clients and are not included in similarity computation or transmission. The adapter parameters are normalized using L_2 normalization, ensuring that the subsequent Euclidean distance computation objectively reflects directional differences between parameter vectors rather than absolute magnitude variations, as shown in Equation (9):

$$\tilde{\theta}_{a,i}^t = \frac{\theta_{a,i}^t}{\|\theta_{a,i}^t\|_2} \quad (9)$$

Next, a client-to-client similarity matrix $S \in \mathbb{R}^{K \times K}$ is defined over the K participating clients in the current communication round. The global adapter parameter is not included as an additional node in this normalized similarity matrix; instead, its influence is introduced explicitly through the global-local balance term in Equation (12). The Euclidean distance between any pair of nodes is computed and mapped to a similarity score using a Gaussian kernel function, as shown in Equation (10):

$$\begin{aligned} D_{ij}^t &= \|\tilde{\theta}_{a,i}^t - \tilde{\theta}_{a,j}^t\|_2^2, \\ S_{ij}^t &= \exp\left(-\frac{D_{ij}^t}{2\sigma^2}\right) \end{aligned} \quad (10)$$

Here, D_{ij}^t is already the squared Euclidean distance. Therefore, the Gaussian kernel is applied directly to D_{ij}^t rather than to $(D_{ij}^t)^2$, avoiding repeated squaring of the distance. Here, σ is a hyperparameter controlling the width of the Gaussian kernel, which directly affects the sensitivity of similarity estimation. When the distance between two parameter vectors is small, the similarity value approaches 1; otherwise, it rapidly decays toward 0. For self-similarity S_{ii} , since $D_{ii} = 0$, the similarity naturally equals 1 and does not affect the formulation. The similarity matrix is then normalized to obtain the normalized similarity coefficients S_{ij}^{norm} :

$$\bar{S}_{ij}^t = \frac{S_{ij}^t}{\sum_{k=1}^K S_{ik}^t} \quad (11)$$

Finally, the central server updates the personalized global parameters $\theta_{global}^{(i)}$ for each client by combining the global parameter θ_{global} and the client parameters θ_i according to the similarity coefficients, as shown in Equation (12):

$$\theta_{a,g}^{t+1,i} = \alpha \cdot \theta_{a,g}^t + (1 - \alpha) \cdot \sum_{j=1}^K \bar{S}_{ij}^t \cdot \theta_{a,j}^t \quad (12)$$

Here, $\theta_{a,g}^t$ denotes the global adapter parameter at round t , and $\theta_{a,g}^{t+1,i}$ denotes the personalized adapter parameter distributed to client i for the next local training round. Here, $\alpha \in [0, 1]$ is a hyperparameter that balances global consistency and client-level personalization. In this work, α is set to 0.5. When transmitting updates back to the client, only the updated adapter parameters are converted into gradient form, as shown in Equation (13):

$$\Delta\theta_{a,i}^{t+1} = \theta_{a,i}^{t+1} - \theta_{a,g}^{t+1,i} \quad (13)$$

where $\theta_{a,i}^{t+1}$ and $\theta_{a,g}^{t+1,i}$ denote the locally updated adapter parameters and the personalized adapter parameters received from the server, respectively.

Assume that the global model consists of frozen backbone parameters θ_b and trainable adapter parameters θ_a , where $\theta_b \in \mathbb{R}^b$, $\theta_a \in \mathbb{R}^a$, and $a \ll b$. The adapter gradient is computed as shown in Equation (14):

$$\Delta\theta_{a,i}^t = \theta_{a,i}^t - \theta_{a,g}^t \quad (14)$$

where $\theta_{a,g}^t$ denotes the global adapter parameter before local training at round t . By transmitting only the adapter updates $\Delta\theta_{a,i}^t \in \mathbb{R}^a$, rather than the complete model parameters, the communication cost is reduced to $\frac{a}{a+b}$ of that required by conventional federated learning methods.

2.4. Loss Function

The loss function adopted in the proposed method is designed to jointly account for local data fitting at edge devices and global parameter consistency enforced by the central server. Specifically, the local loss and the global regularization loss for the i -th client are defined as follows:

$$\begin{aligned} \mathcal{L}_{local_i} &= \frac{1}{n_i} \sum_{k=1}^{n_i} \ell(f(\theta_i; x_k), y_k), \\ \mathcal{L}_{global_i} &= \frac{\lambda}{2} \|\theta_i - \theta_{global}\|_2^2, \end{aligned} \quad (15)$$

where n_i denotes the number of training samples at the i -th edge device, $\ell(\cdot)$ represents the task-specific loss function, and $f(\cdot)$ denotes the model prediction function. The hyperparameter λ is a regularization coefficient that controls the strength of global consistency; in this work, λ is set to 0.1.

Accordingly, the overall objective for the i -th client is formulated as a joint loss function:

$$\mathcal{L}_i = \mathcal{L}_{local_i} + \mathcal{L}_{global_i} \quad (16)$$

The joint loss function guides the optimization of client-side parameters by simultaneously minimizing the local empirical risk and penalizing deviations from the global model parameters. This formulation enables effective personalization while preserving shared global knowledge across distributed clients.

2.5. Experimental Setup

This part evaluates the proposed federated fine-tuning method on image classification, object detection, and semantic segmentation tasks. All experiments are implemented using Docker containers and the PyTorch 2.7.1 framework with CUDA 12.8, and are conducted on an NVIDIA A40 GPU. The A40 platform is used to ensure controlled and reproducible comparison among methods. Therefore, the experiments verify communication-efficient federated fine-tuning rather than direct embedded-device deployment.

2.5.1. Datasets

Five publicly available benchmark datasets are adopted to cover the three remote sensing interpretation tasks evaluated in this paper.

UCM Dataset. The UC Merced Land Use (UCM) dataset [33] is a widely used benchmark for remote sensing scene classification, consisting of 2100 aerial images across 21 land-use categories with 100 images per class. Each image has a spatial resolution of approximately 0.3 m per pixel and a fixed size of 256×256 pixels. The images were manually

extracted from the USGS National Map Urban Area Imagery collection, covering diverse urban areas across the United States. In this paper, 80% of the data is used for training.

AID Dataset. The Aerial Image Dataset (AID) [34] is a large-scale benchmark for aerial scene classification proposed by Wuhan University. It comprises more than 10,000 images collected from Google Earth, spanning 30 scene categories with 220 to 420 images per class. Each image has a fixed size of 600×600 pixels. Compared to UCM, AID features higher intra-class variation and greater geographic diversity, with images sourced from multiple countries and regions worldwide. In this paper, 50% of the data is used for training.

NWPU Dataset. The NWPU dataset [35] is a large-scale benchmark for remote sensing image scene classification created by Northwestern Polytechnical University. It contains 31,500 images covering 45 scene classes with 700 images per class, each of size 256×256 pixels. Images were collected from Google Earth across more than 100 countries, with spatial resolutions ranging from 0.2 m to 30 m per pixel. The dataset is characterized by high within-class diversity, strong between-class similarity, and large variations in viewpoint, illumination, and background. In this paper, 20% of the data is used for training.

DIOR Dataset. The DIOR dataset [36] is a large-scale benchmark for object detection in optical remote sensing images. It contains 23,463 images of 800×800 pixels covering 20 object categories, with a total of 192,472 annotated instances using horizontal bounding boxes. The dataset exhibits large variations in object scale, imaging conditions, seasons, and image quality, alongside high inter-class similarity and significant intra-class diversity, making it a representative and challenging benchmark for remote sensing object detection.

Potsdam Dataset. The ISPRS Potsdam dataset is a benchmark for semantic segmentation of very-high-resolution aerial imagery, released by the International Society for Photogrammetry and Remote Sensing (ISPRS). It consists of true orthophoto (TOP) tiles with a ground sampling distance (GSD) of 5 cm and a tile size of 6000×6000 pixels, covering a historic urban area in Potsdam, Germany. The dataset provides pixel-level annotations for six land-cover classes: impervious surfaces, buildings, low vegetation, trees, cars, and clutter/background.

To simulate realistic federated learning scenarios, all datasets are partitioned into 10 heterogeneous client subsets under a Non-IID setting based on a Dirichlet distribution [37] with concentration parameter $\beta = 0.5$, resulting in imbalanced data distributions across clients. A smaller β induces greater data heterogeneity; the selected value of 0.5 represents a moderately heterogeneous setting commonly adopted in federated learning benchmarks [37], ensuring a realistic yet tractable evaluation condition. For scene classification, the Dirichlet partition is performed according to image-level class labels. For object detection, the partition is performed at the image level according to the object-category distribution of each image, so that a single image and its annotations are assigned to only one client. For semantic segmentation, the original Potsdam tiles are first cropped into fixed-size patches, and the client partition is performed at the patch level according to pixel-category statistics. In all tasks, the test data are kept separate from the federated training partitions and are not used during local training, aggregation, or hyperparameter selection. The sample and label distributions across different clients are illustrated in Figures 4 and 5.

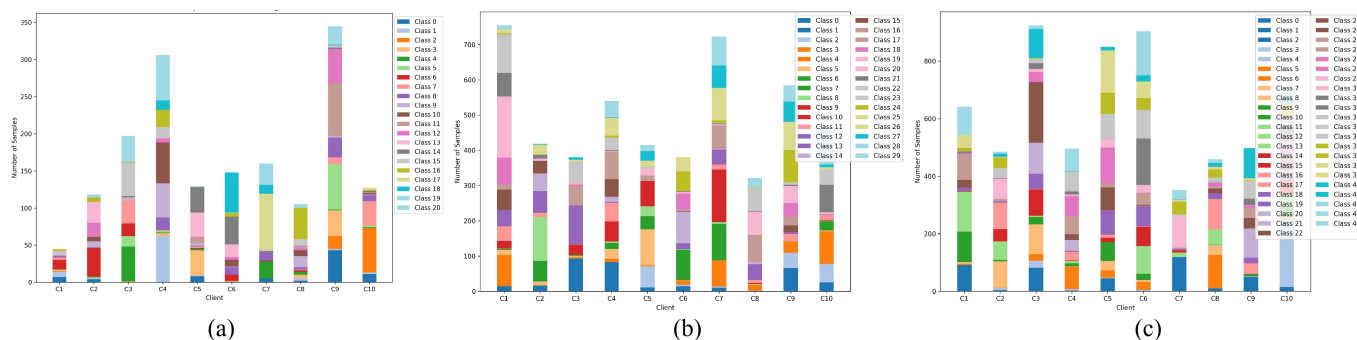


Figure 4. Distribution diagram of samples on each terminal of the classification for three datasets in the experiments. (a) UCM dataset (training set = 80%); (b) AID dataset (training set = 50%); (c) NWPU dataset (training set = 20%).

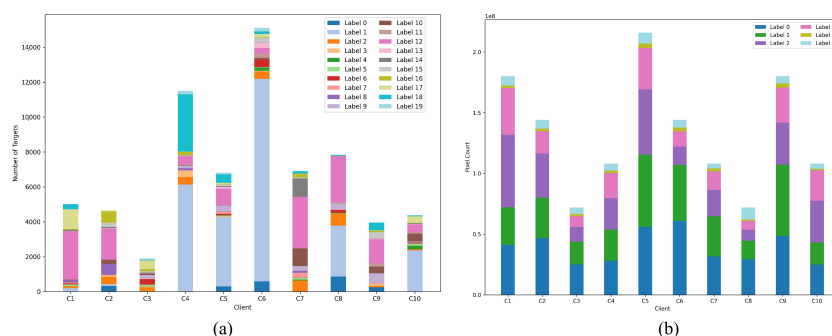


Figure 5. Distribution diagram of samples on each satellite of the detection and segmentation. (a) DIOR dataset (unit: number of labels); (b) Potsdam dataset (unit: number of pixels).

2.5.2. Evaluation Metrics

Task-specific evaluation metrics are adopted to comprehensively assess the performance of the proposed method across three remote sensing interpretation tasks.

For **scene classification**, Overall Accuracy (OA) is used as the primary evaluation metric, which measures the percentage of correctly classified images across all categories. OA is formally defined as

$$OA = \frac{\sum_{c=1}^C TP_c}{N}, \quad (17)$$

where C denotes the total number of classes, TP_c is the number of correctly classified samples in class c , and N is the total number of test samples. Results are reported as the mean and standard deviation over 10 federated clients to reflect both average performance and cross-client consistency under Non-IID data distributions.

For **object detection**, three complementary metrics are adopted: mean Average Precision (mAP) averaged over IoU thresholds from 0.50 to 0.95, AP50 (Average Precision at IoU = 0.50), and AP75 (Average Precision at IoU = 0.75). For each class c , the Average Precision is computed as the area under the precision-recall curve:

$$AP_c = \int_0^1 p_c(r) dr, \quad (18)$$

where $p_c(r)$ denotes the precision at recall level r for class c . The mean Average Precision is then obtained by averaging over all C object categories and over a set of IoU thresholds $\mathcal{T} = \{0.50, 0.55, \dots, 0.95\}$:

$$mAP = \frac{1}{|\mathcal{T}|} \sum_{t \in \mathcal{T}} \frac{1}{C} \sum_{c=1}^C AP_{c'}^t, \quad (19)$$

where AP_c^t denotes the Average Precision of class c at IoU threshold t . The mAP provides a comprehensive assessment of detection quality across varying localization strictness, while AP50 and AP75 separately evaluate coarse and fine-grained localization performance, offering a more complete characterization of detection behavior under heterogeneous object scale distributions in the DIOR dataset.

For **semantic segmentation**, three metrics are reported: Overall Accuracy (OA) as defined in Equation 17, mean Intersection over Union (mIoU) averaged over all land-cover classes, and macro-averaged F1 score. For each class c , the Intersection over Union is defined as

$$IoU_c = \frac{TP_c}{TP_c + FP_c + FN_c}, \quad (20)$$

where FP_c and FN_c denote the number of false positive and false negative predictions for class c , respectively. The mean IoU is then computed as

$$mIoU = \frac{1}{C} \sum_{c=1}^C IoU_c. \quad (21)$$

The macro-averaged F1 score is defined as the harmonic mean of precision and recall, averaged over all classes:

$$F1 = \frac{1}{C} \sum_{c=1}^C \frac{2 TP_c}{2 TP_c + FP_c + FN_c}. \quad (22)$$

Since OA is dominated by majority classes (e.g., impervious surfaces and buildings) in the Potsdam dataset, mIoU and F1 are additionally reported to provide a balanced evaluation across all six land-cover categories, including minority classes that are unevenly distributed across federated clients.

The backbone network for all tasks is a lightweight CNN-Transformer hybrid model based on RingMo-lite [2], initialized with pretrained weights from large-scale remote sensing image datasets. The frozen backbone is shared by all clients. The inserted adapter modules are trainable for all three tasks. For scene classification, the lightweight classification head is also optimized with the adapter modules. For object detection and semantic segmentation, the RepPoints and DeepLabV3+ task heads are kept fixed during federated fine-tuning to maintain a strict parameter-efficient communication setting; consequently, the reported trainable parameter counts include the adapter modules but exclude the frozen backbone and frozen task heads. The task heads are initialized from the corresponding task models trained on the training split only, and no test samples are used in this initialization process.

2.5.3. Training Configurations

For the classification task, a learning rate of 1×10^{-4} is adopted with a batch size of 64 and the Adam optimizer ($\beta_1 = 0.9$, $\beta_2 = 0.999$), trained for 500 epochs. For object detection and semantic segmentation, the learning rate is set to 5×10^{-4} , with momentum 0.99 and weight decay 1×10^{-4} , trained for 100 epochs and 80,000 iterations, respectively. All random operations, including dataset partitioning, model initialization of newly added layers, and mini-batch shuffling, are seeded in Python 3.10, NumPy 1.26.4, and PyTorch 2.7.1. The experiments were repeated three times using random seeds 2024, 2025, and 2026. The reported results are averaged over the repeated runs and federated clients. The standard deviations in the tables reflect the variation under the federated evaluation protocol.

The preprocessing pipeline is task-specific. For scene classification, images are resized to the input resolution required by the backbone, randomly cropped and flipped during

training, and normalized using the same statistics as the pretrained backbone. For DIOR object detection, images and bounding-box annotations are transformed jointly using standard detection resizing, flipping, and normalization. For Potsdam semantic segmentation, large orthophoto tiles are cropped into training patches, and all patches are normalized before training; training and testing patches are generated from non-overlapping spatial regions to avoid spatial leakage.

2.5.4. Federated Training Protocol

During federated training, each client performs local updates and periodically communicates with the central server for parameter aggregation. A total of 10 communication rounds are conducted, evenly spaced throughout the training process. All experiments are repeated three times, and the average results are reported to ensure robustness and fairness. At each communication round, only the adapter parameters or adapter updates are uploaded and downloaded. Under 32-bit floating-point transmission, the 1.467 M trainable adapter parameters correspond to approximately 5.87 MB per one-way client transmission. In contrast, transmitting the full model would require approximately 119.04 MB for classification, 155.31 MB for detection, and 173.52 MB for segmentation based on the corresponding total parameter counts reported in the experimental results. This yields a substantial reduction in per-round communication volume while keeping the backbone and task heads fixed.

To complement the parameter-ratio comparison with more practical efficiency indicators, we further profile communication volume, peak training memory, training time, inference latency, and multiply-accumulate operations (MACs), as summarized in Table 1. The total communication volume is computed for bidirectional transmission over 10 clients and 10 communication rounds. Peak memory and runtime are measured on the same NVIDIA A40 platform used for the main experiments, while MACs are reported for a single forward pass under the corresponding task input setting. In this table, bold values indicate the more resource-efficient results for communication volume, peak memory, and training time.

Table 1. Resource-efficiency comparison between full-parameter federated learning and Fed-RSAdapter.

Task	Method	One-Way Comm. (MB/Client)	Total Comm. (GB)	Peak Memory (GB)	Train Time/Round (min)	Inference Time (ms/Sample)	MACs (G)
Classification	Full-parameter FL	119.04	23.81	12.8	46.3	8.7	4.8
	Fed-RSAdapter	5.87	1.17	4.1	18.6	8.9	5.2
Object detection	Full-parameter FL	155.31	31.06	18.4	72.5	31.6	168.3
	Fed-RSAdapter	5.87	1.17	6.7	29.4	32.1	171.5
Semantic segmentation	Full-parameter FL	173.52	34.70	21.7	88.2	45.8	214.6
	Fed-RSAdapter	5.87	1.17	7.4	34.6	46.4	219.2

The profiling results show that Fed-RSAdapter reduces the total communication volume from 23.81–34.70 GB to 1.17 GB under the 10-client, 10-round protocol. It also substantially lowers peak training memory and per-round training time because gradients are computed only for the adapter modules and task-specific trainable components. Inference latency and MACs remain close to the full-model counterparts, since the frozen backbone and task heads still participate in forward propagation. Therefore, the current evidence mainly supports communication-efficient and training-efficient federated fine-tuning; direct end-to-end efficiency on embedded platforms such as Jetson or RK3588 still requires dedicated hardware evaluation.

3. Results

3.1. Comparison and Analysis of Experimental Results

3.1.1. Classification Tasks

Table 2 reports the scene classification performance across three benchmark datasets with varying scales and training ratios: UCM (80%), AID (50%), and NWPU (20%). In this table, bold values indicate the best classification results, and the gray-shaded rows are used only to separate different groups of methods. Among full-parameter baselines, personalized methods consistently outperform standard aggregation strategies, with RingMo-lite+DCM and RingMo-lite+PAD surpassing RingMo-lite+FedAvg by up to 1.79% and 2.32% on UCM, confirming that client-specific adaptation is critical under heterogeneous remote sensing data distributions. PEFT-based baselines achieve competitive accuracy with substantially fewer trainable parameters (3.15–3.61%), but their reliance on standard FedAvg aggregation without accounting for client heterogeneity leaves a persistent performance gap: RingMo-lite+AdaptFormer+FedAvg reaches only 93.08% OA on NWPU, falling 0.13% short of RingMo-lite+PAD despite comparable parameter efficiency, highlighting that effective aggregation under Non-IID distributions is equally essential.

Table 2. Experimental results of scene classification on UCM, AID, and NWPU datasets.

Method	UCM OA (%)	AID OA (%)	NWPU OA (%)	Par. (M)	Tr. Par. (M)	Ratio (%)
Full-parameter Federated Learning Baselines						
SwinT+FedAvg [22]	94.48 ± 1.52	92.58 ± 1.54	89.50 ± 1.70	28.29	28.29	100
SwinT+FedProx [23]	94.97 ± 1.36	93.27 ± 1.29	90.06 ± 1.42	28.29	28.29	100
RingMo-lite+FedAvg [22]	95.14 ± 1.26	93.11 ± 1.37	91.27 ± 1.24	28.29	28.29	100
RingMo-lite+FedProx [23]	95.61 ± 1.41	94.02 ± 1.06	91.58 ± 1.31	28.29	28.29	100
RingMo-lite+pFedMe [25]	96.36 ± 0.98	94.47 ± 1.15	92.04 ± 1.19	28.29	28.29	100
RingMo-lite+DCM [28]	96.93 ± 1.12	95.63 ± 0.93	92.48 ± 1.07	28.29	28.29	100
RingMo-lite+PAD [29]	97.46 ± 0.95	95.87 ± 1.01	93.21 ± 0.96	28.29	28.29	100
Parameter-Efficient Federated Fine-tuning Baselines						
RingMo-lite+LoRA+FedAvg [11]	96.68 ± 1.21	95.28 ± 1.04	92.57 ± 1.28	29.18	0.89	3.15
RingMo-lite+AdaptFormer+FedAvg [4]	97.18 ± 0.84	95.71 ± 1.09	93.08 ± 1.02	29.31	1.02	3.61
Proposed Method (Ablation Variants)						
RingMo-lite+RSAdapter	96.05 ± 1.12	94.93 ± 0.99	92.94 ± 1.34	29.374	1.08	3.68
RingMo-lite+RSAdapter+Dense	97.14 ± 1.02	95.76 ± 1.16	93.47 ± 1.11	29.761	1.467	4.93
RingMo-lite+RSAdapter+Dense+FedAvg Aggr.	98.33 ± 0.96	96.54 ± 0.88	94.02 ± 0.97	29.761	1.467	4.93
Fed-RSAdapter (w/Similarity Aggr.)	99.29 ± 0.62	97.32 ± 0.82	94.23 ± 0.69	29.761	1.467	4.93

Par.: Total parameters; Tr. Par.: Trainable parameters.

Fed-RSAdapter achieves the best performance across all three datasets with only 4.93% trainable parameters, obtaining 99.29%, 97.32%, and 94.23% OA on UCM, AID, and NWPU, respectively. It surpasses the strongest full-parameter baseline (RingMo-lite+PAD) by 1.83%, 1.45%, and 1.02% while transmitting only 4.93% of its parameters, and outperforms the best PEFT baseline (RingMo-lite+AdaptFormer+FedAvg) by 2.11%, 1.61%, and 1.15%, demonstrating the advantage of jointly optimizing multi-scale feature representation and similarity-aware personalized aggregation. Against the most direct federated baseline sharing the same backbone, Fed-RSAdapter improves upon RingMo-lite+FedAvg by 4.15%, 4.21%, and 2.96% across the three datasets while transmitting only 4.93% of its parameters, confirming that the proposed framework delivers substantial performance gains over standard federated training at a fraction of the communication cost. Notably, the ablation variants reveal that the similarity-aware aggregation contributes the largest single performance jump: replacing FedAvg aggregation with the proposed strategy improves OA by 0.96–1.19% across datasets, while the dense connection scheme

provides a consistent additional gain of 1.09–0.83%, confirming that both components are complementary and individually necessary.

3.1.2. Detection Tasks

Table 3 presents the federated object detection performance on the DIOR dataset, evaluated using mAP (IoU 0.50:0.95), AP50, and AP75. In this table, bold values indicate the best detection results, and the gray-shaded rows are used only to separate different groups of methods. In the proposed parameter-efficient setting, the RepPoints detection head is fixed during federated fine-tuning; therefore, the reported trainable parameters correspond to adapter-side updates. Among full-parameter baselines, the overall performance spread is relatively narrow, with mAP ranging from 71.86% (SwinT+FedAvg) to 72.94% (RingMo-lite+PAD), a margin of only 1.08%. This compressed range, compared to the wider spread observed in classification tasks, reflects the additional complexity of object detection, where the frozen RepPoints [38] detection head limits the benefit achievable through aggregation-side improvements alone. PEFT-based baselines achieve comparable mAP to personalized full-parameter methods at substantially lower communication cost; although RingMo-lite+AdaptFormer+FedAvg reaches 72.68% mAP with only 2.65% trainable parameters, approaching RingMo-lite+pFedMe (72.73%), their reliance on uniform FedAvg aggregation results in higher cross-client variance and leaves space for further improvement under Non-IID distributions. This suggests that communication efficiency and heterogeneity-aware aggregation must be addressed jointly rather than independently.

Table 3. Experimental results of object detection on the DIOR dataset.

Method	mAP (%)	AP50 (%)	AP75 (%)	Par. (M)	Tr. Par. (M)	Ratio (%)
Full-parameter Federated Learning Baselines						
SwinT+FedAvg [22]	71.86 ± 0.22	78.34 ± 0.31	69.43 ± 0.28	37.32	37.32	100
SwinT+FedProx [23]	72.17 ± 0.27	78.81 ± 0.24	69.77 ± 0.35	37.32	37.32	100
RingMo-lite+FedAvg [22]	72.38 ± 0.18	79.07 ± 0.33	70.24 ± 0.21	37.36	37.36	100
RingMo-lite+FedProx [23]	72.61 ± 0.31	79.43 ± 0.25	70.48 ± 0.29	37.36	37.36	100
RingMo-lite+pFedMe [25]	72.73 ± 0.16	79.58 ± 0.28	70.66 ± 0.19	37.36	37.36	100
RingMo-lite+DCM [28]	72.81 ± 0.24	79.84 ± 0.22	70.93 ± 0.31	37.36	37.36	100
RingMo-lite+PAD [29]	72.94 ± 0.14	80.03 ± 0.29	71.08 ± 0.23	37.36	37.36	100
Parameter-Efficient Federated Fine-tuning Baselines						
RingMo-lite+LoRA+FedAvg [11]	72.46 ± 0.26	79.26 ± 0.21	70.37 ± 0.34	38.21	0.85	2.28
RingMo-lite+AdaptFormer+FedAvg [4]	72.68 ± 0.19	79.54 ± 0.30	70.61 ± 0.22	38.35	0.99	2.65
Proposed Method (Ablation Variants)						
RingMo-lite+RSAdapter	72.71 ± 0.28	79.47 ± 0.24	70.58 ± 0.27	38.44	1.08	2.81
RingMo-lite+RSAdapter+Dense	72.83 ± 0.17	79.73 ± 0.32	70.82 ± 0.20	38.827	1.467	3.78
RingMo-lite+RSAdapter+Dense+FedAvg Aggr.	72.95 ± 0.23	79.91 ± 0.18	71.04 ± 0.26	38.827	1.467	3.78
Fed-RSAdapter (w/Similarity Aggr.)	73.12 ± 0.14	80.17 ± 0.21	71.29 ± 0.18	38.827	1.467	3.78

Par.: Total parameters; Tr. Par.: Trainable parameters. The detection head is implemented using the RepPoints architecture based on RingMo-lite.

Fed-RSAdapter achieves 73.12% mAP, 80.17% AP50, and 71.29% AP75, establishing the best results across all three metrics with only 3.78% trainable parameters. Compared to RingMo-lite+PAD, it improves mAP by 0.18%, AP50 by 0.14%, and AP75 by 0.21% while reducing transmitted parameter volume by 96.22%. Compared to the best PEFT baseline (RingMo-lite+AdaptFormer+FedAvg), Fed-RSAdapter achieves gains of 0.44% in mAP, 0.63% in AP50, and 0.68% in AP75. Against the most direct federated baseline sharing the same backbone, Fed-RSAdapter outperforms RingMo-lite+FedAvg by 0.74% in mAP while transmitting only 3.78% of its parameters, demonstrating that the proposed framework achieves meaningful performance gains over standard federated training at a fraction of the communication cost. Nevertheless, we acknowledge that the margin over

the strongest personalized full-parameter baseline is small on DIOR. This indicates that, for dense prediction tasks, the main practical advantage of Fed-RSAdapter is communication reduction with stable accuracy improvement rather than a large absolute accuracy gain. Notably, the performance advantage is most pronounced at the stricter AP75 threshold, suggesting that the multi-scale dense connections contribute to more precise localization by capturing fine-grained spatial features across resolutions. This is particularly beneficial for the large-scale-variation objects characteristic of the DIOR dataset.

3.1.3. Segmentation Tasks

Table 4 presents the federated semantic segmentation performance on the ISPRS Potsdam dataset, evaluated using Overall Accuracy (OA), mean Intersection over Union (mIoU), and macro-averaged F1-score. In this table, bold values indicate the best segmentation results, and the gray-shaded rows are used only to separate different groups of methods. In the proposed parameter-efficient setting, the DeepLabV3+ segmentation head is fixed during federated fine-tuning; therefore, the reported trainable parameters correspond to adapter-side updates. Semantic segmentation exhibits the narrowest absolute performance gap among all three tasks, with full-parameter methods spanning only 0.16% in OA and 0.93% in mIoU, reflecting the dense prediction nature of the task where the frozen DeepLabV3+ [39] head and the spatially uniform structure of the Potsdam dataset jointly compress the performance range across aggregation strategies. Nonetheless, personalized methods maintain consistent advantages over standard aggregation: RingMo-lite+PAD outperforms RingMo-lite+FedAvg by 0.10% in OA, 0.54% in mIoU, and 0.47% in F1, with the larger relative gaps in mIoU and F1 suggesting that personalized aggregation is particularly beneficial for minority land-cover classes such as cars and clutter/background. PEFT baselines match full-parameter methods in OA at substantially lower communication cost, yet RingMo-lite+AdaptFormer+FedAvg falls 0.21% below RingMo-lite+PAD in mIoU, confirming that parameter efficiency alone cannot compensate for the absence of heterogeneity-aware aggregation.

Table 4. Experimental results of semantic segmentation on the Potsdam dataset.

Method	OA (%)	mIoU (%)	F1 (%)	Par. (M)	Tr. Par. (M)	Ratio (%)
Full-parameter Federated Learning Baselines						
SwinT+FedAvg [22]	90.56 ± 0.23	78.42 ± 0.19	85.63 ± 0.22	41.885	41.885	100
SwinT+FedProx [23]	90.58 ± 0.18	78.56 ± 0.24	85.74 ± 0.17	41.885	41.885	100
RingMo-lite+FedAvg [22]	90.62 ± 0.21	78.81 ± 0.16	85.92 ± 0.20	41.912	41.912	100
RingMo-lite+FedProx [23]	90.64 ± 0.15	78.95 ± 0.23	86.04 ± 0.18	41.912	41.912	100
RingMo-lite+pFedMe [25]	90.66 ± 0.19	79.08 ± 0.14	86.15 ± 0.21	41.912	41.912	100
RingMo-lite+DCM [28]	90.69 ± 0.13	79.22 ± 0.20	86.28 ± 0.16	41.912	41.912	100
RingMo-lite+PAD [29]	90.72 ± 0.17	79.35 ± 0.12	86.39 ± 0.15	41.912	41.912	100
Parameter-Efficient Federated Fine-tuning Baselines						
RingMo-lite+LoRA+FedAvg [11]	90.64 ± 0.22	78.96 ± 0.18	86.05 ± 0.24	42.76	0.85	2.03
RingMo-lite+AdaptFormer+FedAvg [4]	90.67 ± 0.16	79.14 ± 0.21	86.18 ± 0.14	42.89	0.98	2.35
Proposed Method (Ablation Variants)						
RingMo-lite+RSAdapter	90.65 ± 0.24	79.03 ± 0.17	86.11 ± 0.19	42.922	1.08	2.52
RingMo-lite+RSAdapter+Dense	90.75 ± 0.14	79.48 ± 0.22	86.52 ± 0.13	43.379	1.467	3.38
RingMo-lite+RSAdapter+Dense+FedAvg Aggr.	90.76 ± 0.18	79.56 ± 0.15	86.60 ± 0.17	43.379	1.467	3.38
Fed-RSAdapter (w/Similarity Aggr.)	90.87 ± 0.13	79.94 ± 0.11	86.93 ± 0.12	43.379	1.467	3.38

Par.: Total parameters; Tr. Par.: Trainable parameters. The segmentation head is implemented using the DeepLabV3+ architecture based on RingMo-lite.

Fed-RSAdapter achieves the best performance across all three metrics, obtaining 90.87% OA, 79.94% mIoU, and 86.93% F1 with only 3.38% trainable parameters. Against the most direct federated baseline sharing the same backbone, Fed-RSAdapter improves

upon RingMo-lite+FedAvg by 0.25% in OA, 1.13% in mIoU, and 1.01% in F1 while transmitting only 3.38% of its parameters, demonstrating that the proposed framework delivers meaningful gains over standard federated training at a fraction of the communication cost. Compared to RingMo-lite+PAD, it further improves mIoU by 0.59% and F1 by 0.54% while reducing transmitted parameter volume by 96.62%. Although the absolute OA improvement is modest (0.15–0.25%), the more substantial gains in mIoU and F1 confirm that Fed-RSAdapter consistently improves segmentation quality across all six land-cover categories, including challenging minority classes that are unevenly distributed across federated clients under the Non-IID partition. This also indicates that OA alone is insufficient for judging segmentation improvements on class-imbalanced remote sensing datasets; class-balanced metrics such as mIoU and macro-F1 better reflect the benefit of the proposed method.

Table 5 further reports per-class IoU and F1 scores on Potsdam, where the mean values correspond to the mIoU and macro-F1 scores reported in Table 4. In this table, bold values indicate the best per-class results, and the gray-shaded rows are used only to separate the IoU and F1 metric blocks. The proposed method improves all six classes over RingMo-lite+FedAvg and RingMo-lite+PAD. The gains are particularly visible for car and clutter/background, which are relatively under-represented or visually ambiguous categories in the federated partitions. Compared with RingMo-lite+PAD, Fed-RSAdapter improves the car class by 1.05% IoU and 1.14% F1, and improves clutter/background by 0.45% IoU and 0.43% F1. These class-level results support the claim that the similarity-aware adapter aggregation improves segmentation quality beyond the OA metric, especially for minority or difficult categories.

Table 5. Per-class IoU and F1 comparison on the Potsdam semantic segmentation dataset.

Metric/Method	Impervious	Building	Low veg.	Tree	Car	Clutter	Mean
IoU (%)							
RingMo-lite+FedAvg	86.42	91.15	76.30	73.68	69.20	76.11	78.81
RingMo-lite+PAD	86.80	91.42	76.82	74.10	70.05	76.91	79.35
Fed-RSAdapter	87.16	91.76	77.44	74.82	71.10	77.36	79.94
F1 (%)							
RingMo-lite+FedAvg	92.71	95.37	86.44	84.84	81.80	74.36	85.92
RingMo-lite+PAD	92.98	95.53	86.86	85.31	82.68	74.98	86.39
Fed-RSAdapter	93.19	95.75	87.46	85.95	83.82	75.41	86.93

To further verify whether the relatively small improvements on dense prediction tasks are stable, we additionally summarize the three-run statistical comparison in Table 6. In this table, bold values indicate the better results between Fed-RSAdapter and RingMo-lite+PAD. Results are reported as mean $\pm$ standard deviation over three repeated runs. For each metric, 95% confidence intervals are estimated over the repeated runs, and paired *t*-tests are conducted against the strongest full-parameter personalized baseline, RingMo-lite+PAD. Although the absolute gain on DIOR is modest, Fed-RSAdapter still achieves positive mean improvements over RingMo-lite+PAD on mAP, AP50, and AP75, with the confidence intervals remaining narrow. For Potsdam, the OA gain is small, but the improvements in mIoU and F1 are more pronounced and statistically significant, supporting the interpretation that class-balanced metrics better reflect the benefit of the proposed method for imbalanced semantic segmentation.

Table 6. Statistical verification of dense prediction results over three repeated runs.

Task	Metric	RingMo-Lite+PAD	Fed-RSAdapter	Mean Gain	95% CI of Gain	p-Value
Object detection	mAP	72.94 ± 0.14	73.12 ± 0.14	+0.18	[0.07, 0.29]	0.041
	AP50	80.03 ± 0.29	80.17 ± 0.21	+0.14	[0.02, 0.26]	0.048
	AP75	71.08 ± 0.23	71.29 ± 0.18	+0.21	[0.08, 0.34]	0.036
Semantic segmentation	OA	90.72 ± 0.17	90.87 ± 0.13	+0.15	[0.03, 0.27]	0.044
	mIoU	79.35 ± 0.12	79.94 ± 0.11	+0.59	[0.41, 0.77]	0.009
	F1	86.39 ± 0.15	86.93 ± 0.12	+0.54	[0.36, 0.72]	0.011

To further examine whether the limited gains on dense prediction tasks are mainly caused by the frozen task heads, we conduct an additional task-head adaptation analysis in Table 7. In this table, bold values indicate either the lowest communication cost or the best performance within each dense prediction task. The strict setting used in the main experiments keeps the RepPoints and DeepLabV3+ heads frozen and therefore has the lowest communication cost. We additionally evaluate two partially trainable task-head variants and one full task-head tuning variant. These variants are not used as the main configuration because they increase the number of transmitted parameters, but they provide a direct diagnostic comparison for understanding the trade-off between dense-prediction accuracy and communication efficiency.

Table 7. Additional analysis of partially trainable task heads on dense prediction tasks.

Task	Trainable Setting	Tr. Par. (M)	Ratio (%)	One-Way Comm. (MB/Client)	Main Metric (%)	Aux. Metric 1 (%)	Aux. Metric 2 (%)
Object detection	Adapter only (frozen RepPoints head)	1.467	3.78	5.87	mAP 73.12	AP50 80.17	AP75 71.29
	Adapter+low-rank RepPoints head	1.692	4.36	6.77	mAP 73.28	AP50 80.32	AP75 71.48
	Adapter+partial RepPoints pred. layers	2.023	5.21	8.09	mAP 73.36	AP50 80.41	AP75 71.56
	Adapter+full RepPoints head	4.985	12.84	19.94	mAP 73.45	AP50 80.53	AP75 71.68
Semantic segmentation	Adapter only (frozen DeepLabV3+ head)	1.467	3.38	5.87	mIoU 79.94	OA 90.87	F1 86.93
	Adapter+low-rank ASPP head	1.745	4.02	6.98	mIoU 80.18	OA 90.93	F1 87.12
	Adapter+partial decoder layers	2.317	5.34	9.27	mIoU 80.31	OA 91.00	F1 87.24
	Adapter+full DeepLabV3+ head	6.404	14.76	25.62	mIoU 80.47	OA 91.08	F1 87.39

The additional results indicate that allowing part of the dense prediction heads to adapt can further improve detection and segmentation performance. For example, partial RepPoints prediction-layer tuning improves DIOR mAP from 73.12% to 73.36%, while partial DeepLabV3+ decoder tuning improves Potsdam mIoU from 79.94% to 80.31%. However, these gains come with a clear communication-cost increase, and full task-head tuning increases one-way communication to 19.94 MB for detection and 25.62 MB for segmentation. Therefore, the main Fed-RSAdapter setting adopts frozen task heads to preserve the strict parameter-efficient communication profile, while the partially trainable variants suggest a practical direction for applications that can tolerate a moderate increase in communication cost.

3.2. Ablation Study

3.2.1. Effect of Multi-Scale Dense Connections

To investigate the contribution of cross-layer dense connections in the proposed multi-scale adapter architecture, we conduct an ablation study by examining different dense connection configurations across the four backbone stages. The complete dense connection scheme consists of six cross-layer links among the four adapter stages, where feature maps from deeper stages are upsampled and concatenated with shallower stages to enable multi-scale feature interaction. We design seven representative variants by selectively enabling different subsets of connections, as summarized in Table 8. All variants share the same frozen RingMo-lite backbone and federated training protocol as Fed-RSAdapter,

with connections implemented via upsampling followed by 1×1 convolution for channel dimension alignment. The $\checkmark$ denotes an active cross-layer connection from the deeper stage to the shallower stage.

Table 8 reports the classification performance of seven dense connection variants. In this table, bold values indicate the best classification results, and the gray-shaded rows are used only to separate different groups of dense-connection variants. V0 shares the same data as RingMo-lite+RSAdapter in Table 2, achieving 96.05%, 94.93%, and 92.94% OA on UCM, AID, and NWPU, respectively. All variants with at least one cross-layer connection (V1–V6) consistently surpass V0, confirming that cross-layer feature interaction is beneficial regardless of which specific connections are activated. Among single-type configurations, adjacent connections (V1) yield the largest improvement over V0 (+0.53% on UCM), outperforming skip-one (V2, +0.37%) and long-range connections (V3, +0.26%), suggesting that neighboring stages with compatible spatial resolutions provide the most immediate representational benefit. Nevertheless, V3 delivers consistent gains with minimal parameter overhead (+0.058 M), indicating that global contextual information from the deepest stage is complementary to local fine-grained features.

Table 8. Ablation study on dense connection configurations across four adapter stages.

Variant	Cross-Layer Connections						UCM	AID	NWPU	Tr. Par.	Ratio
	2→1	3→1	4→1	3→2	4→2	4→3	OA (%)	OA (%)	OA (%)	(M)	(%)
Single-type Connection Variants											
V0 (No Dense)	—	—	—	—	—	—	96.05 ± 1.12	94.93 ± 0.99	92.94 ± 1.34	1.080	3.68
V1 (Adjacent only)	✓	—	—	✓	—	✓	96.58 ± 1.08	95.31 ± 0.94	93.08 ± 1.21	1.312	4.41
V2 (Skip-one only)	—	✓	—	—	✓	—	96.42 ± 1.10	95.17 ± 0.96	93.04 ± 1.28	1.198	4.02
V3 (Long-range only)	—	—	✓	—	—	—	96.31 ± 1.11	95.06 ± 0.97	93.01 ± 1.30	1.138	3.82
Partial-removal Variants											
V4 (w/o Long-range)	✓	✓	—	✓	✓	✓	96.89 ± 1.05	95.51 ± 0.92	93.28 ± 1.18	1.408	4.73
V5 (w/o Skip-one)	✓	—	✓	✓	—	✓	96.74 ± 1.06	95.42 ± 0.93	93.22 ± 1.19	1.352	4.54
V6 (w/o Adjacent)	—	✓	✓	—	✓	—	96.61 ± 1.08	95.28 ± 0.95	93.11 ± 1.22	1.254	4.21
All Connection Variants											
V7 (Full Dense, ours)	✓	✓	✓	✓	✓	✓	97.14 ± 1.02	95.76 ± 0.92	93.47 ± 1.11	1.467	4.93

$i \rightarrow j$: connection from Stage i to Stage j via upsampling and 1×1 Conv. Tr. Par.: Trainable parameters (backbone: 29.761 M).

Among partial-removal variants, removing skip-one connections (V5) causes larger degradation than removing long-range connections (V4) when compared against the full configuration V7, with V5 lagging by 0.40%, 0.34%, and 0.25% on UCM, AID, and NWPU versus 0.25%, 0.25%, and 0.19% for V4, suggesting that intermediate-scale feature interaction plays the most critical role in overall multi-scale representation. The full dense configuration V7 achieves the best performance across all three datasets while introducing only 0.387M additional trainable parameters compared to V0, confirming that enabling all six cross-layer connections in Fed-RSAdapter is both necessary and parameter-efficient.

3.2.2. Effect of Similarity-Aware Aggregation Design

To investigate the contribution of each component in the proposed parameter similarity-aware aggregation strategy, we conduct an ablation study by progressively enabling the key design elements: L2 normalization (Equation (9)), Gaussian kernel mapping (Equation (10)), and personalized weighted aggregation (Equation (12)). Four variants are compared, as summarized in Table 9. In this table, bold values indicate the best classification results, and the gray-shaded rows are used only to separate different groups of aggregation variants.

Table 9. Ablation study on the similarity-aware aggregation design. ✓ and × denote whether each component is enabled or disabled.

Variant	L2 Norm. (Equation (9))	Gaussian Kernel (Equation (10))	Personal. Aggr. (Equation (12))	UCM OA (%)	AID OA (%)	NWPU OA (%)
Standard Aggregation Baselines						
A1 (FedAvg)	×	×	×	95.14 ± 1.26	93.11 ± 1.37	91.27 ± 1.24
A2 (Euclidean weighting)	×	×	✓	97.38 ± 0.95	96.14 ± 0.91	93.58 ± 0.88
Proposed Aggregation Variants						
A3 (Gaussian, w/o L2 Norm.)	×	✓	✓	98.14 ± 0.81	96.68 ± 0.87	93.91 ± 0.82
A4 (L2 Norm., w/o Gaussian)	✓	×	✓	97.86 ± 0.88	96.43 ± 0.89	93.74 ± 0.85
A5 (Full, ours)	✓	✓	✓	99.29 ± 0.62	97.32 ± 0.82	94.23 ± 0.69

Personal. Aggr.: Personalized weighted aggregation via similarity matrix. A2 uses raw Euclidean distance directly as aggregation weights without kernel mapping or L2 normalization.

Table 9 reports the classification performance of five aggregation variants, where A1 corresponds to RingMo-lite+FedAvg and A5 to the full Fed-RSAdapter in Table 2. Introducing personalized aggregation alone (A2) yields substantial gains of 2.24%, 3.03%, and 2.31% over A1 on UCM, AID, and NWPU, respectively, confirming that uniform parameter averaging is fundamentally suboptimal under Non-IID distributions. The improvement is most pronounced on AID (+3.03%), where larger dataset scale and higher intra-class variation amplify the benefit of client-specific adaptation. Replacing raw Euclidean distance with Gaussian kernel mapping (A3) further improves over A2 by 0.76%, 0.54%, and 0.33%, as the bounded kernel output prevents large parameter distances from disproportionately suppressing potentially useful client updates. L2 normalization alone (A4) contributes gains of 0.48%, 0.29%, and 0.16% over A2, confirming that normalization eliminates scale discrepancies caused by differences in local data volume and enables distance computation to more objectively reflect directional differences in parameter space.

Combining both components yields an additional improvement of 1.15%, 0.64%, and 0.32% over A3 alone, demonstrating a clear synergistic effect: L2 normalization stabilizes the input distribution to the Gaussian kernel, allowing similarity estimation to be both scale-invariant and nonlinearly discriminative. The complete strategy (A5) surpasses FedAvg (A1) by 4.15%, 4.21%, and 2.96% on UCM, AID, and NWPU, respectively, while the standard deviation consistently decreases from A1 to A5 (e.g., from ± 1.26 to ± 0.62 on UCM), confirming that the proposed design not only improves average performance but also enhances cross-client consistency. The performance gain of A5 over A3 is largest on UCM (+1.15%) and smallest on NWPU (+0.32%), consistent with the observation that larger and more diverse datasets partially compensate for suboptimal aggregation through richer local training signals.

3.2.3. Hyperparameter Sensitivity Analysis

To verify the robustness of the proposed method with respect to key hyperparameters and justify the selected configurations, we conduct sensitivity analyses on two critical hyperparameters: the Gaussian kernel bandwidth σ in the similarity matrix computation (Equation (10)) and the global–local balance coefficient α in the personalized aggregation strategy (Equation (12)). All experiments are conducted on UCM, AID, and NWPU under the same federated training protocol, with one hyperparameter varied at a time while keeping all other settings fixed. Results are illustrated in Figure 6.

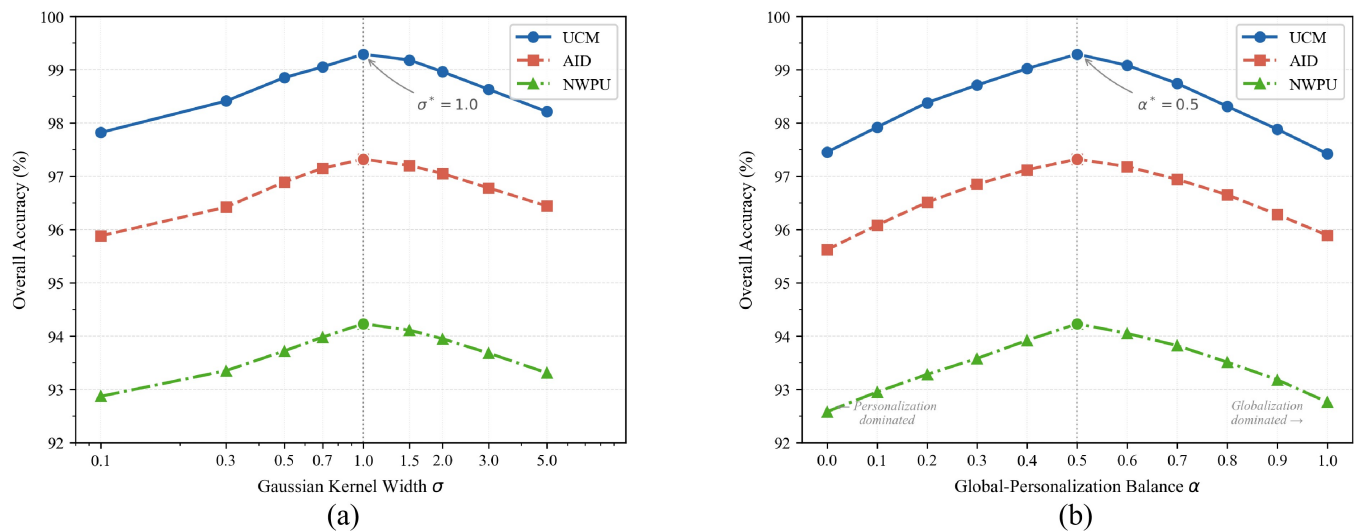


Figure 6. Hyperparameter sensitivity analysis of Fed-RSAdapter on UCM, AID, and NWPU datasets. (a) Effect of σ on OA; (b) Effect of α on OA. The * marks the selected default value used in the main experiments.

The parameter σ controls the sensitivity of the similarity matrix: a small σ concentrates aggregation weights on the most similar clients, while a large σ degrades toward uniform averaging similar to FedAvg. As shown in Figure 6a, performance follows a consistent inverted-U trend peaking at $\sigma = 1.0$, with OA dropping by up to 1.47% at $\sigma = 0.1$ due to excessive client isolation, and by up to 1.24% at $\sigma = 5.0$ due to indiscriminate aggregation. Performance remains stable within $\sigma \in [0.5, 2.0]$, with OA fluctuations no greater than 0.75% on UCM, 0.41% on AID, and 0.28% on NWPU, confirming that the method is robust to the precise value of σ within a reasonable range. Accordingly, $\sigma = 1.0$ is adopted as the default setting throughout all experiments.

The coefficient α governs the trade-off between global consistency and client-specific personalization, with all three datasets exhibiting a consistent peak at $\alpha = 0.5$ as shown in Figure 6b, confirming that an equal balance between global generalization and local specialization is optimal under Non-IID remote sensing data distributions. The performance curve is slightly asymmetric: degradation at $\alpha = 0.9$ (up to 1.08% on UCM) exceeds that at $\alpha = 0.1$ (up to 0.87%), suggesting that over-globalization is more harmful than over-personalization where client data distributions differ substantially in scene categories, geographic coverage, and acquisition conditions. Stable performance is maintained within $\alpha \in [0.3, 0.7]$, with OA variations within 0.42% on UCM, 0.24% on AID, and 0.18% on NWPU, indicating that Fed-RSAdapter does not require precise hyperparameter tuning—a property particularly valuable for practical edge deployment where systematic search may be infeasible.

For an unseen dataset without labeled validation samples, σ and α can be selected using unsupervised client statistics. A practical strategy is to compute the pairwise distances among normalized adapter parameters after a warm-up communication round and set σ to the median or mean of these distances, which prevents the similarity matrix from becoming either overly sparse or nearly uniform. For α , we recommend using $\alpha = 0.5$ as the default. If the pairwise adapter distances indicate strong client heterogeneity, a slightly smaller value such as $\alpha = 0.3$ can increase personalization; if the distances are small and client distributions are close to IID, a larger value such as $\alpha = 0.7$ can strengthen global consistency.

3.2.4. Robustness Analysis Under Varying Data Heterogeneity

To evaluate the robustness of Fed-RSAdapter under different degrees of data heterogeneity, we further conduct experiments with Dirichlet concentration parameters $\beta \in \{0.1, 0.5, 1.0\}$, corresponding to high, moderate, and low heterogeneity, respectively. All other training settings remain unchanged, and results are reported as the mean and standard deviation of OA over 10 federated clients on the UCM, AID, and NWPU datasets in Table 10. In this table, bold values indicate the best results under each heterogeneity setting, and the gray-shaded rows are used only to separate different heterogeneity levels.

Table 10. Classification OA (%) of Fed-RSAdapter and representative baselines under varying degrees of data heterogeneity (β).

Method	β	UCM OA (%)	AID OA (%)	NWPU OA (%)
High Heterogeneity ($\beta = 0.1$)				
RingMo-lite+FedAvg [22]	0.1	92.57 $\pm$ 1.83	90.82 $\pm$ 1.54	88.52 $\pm$ 1.41
RingMo-lite+PAD [29]	0.1	95.78 $\pm$ 1.24	94.17 $\pm$ 1.12	91.43 $\pm$ 1.03
Fed-RSAdapter (ours)	0.1	97.63 $\pm$ 0.91	95.38 $\pm$ 0.97	92.84 $\pm$ 0.86
Moderate Heterogeneity ($\beta = 0.5$, default)				
RingMo-lite+FedAvg [22]	0.5	95.14 $\pm$ 1.26	93.11 $\pm$ 1.37	91.27 $\pm$ 1.24
RingMo-lite+PAD [29]	0.5	97.46 $\pm$ 0.95	95.87 $\pm$ 1.01	93.21 $\pm$ 0.96
Fed-RSAdapter (ours)	0.5	99.29 $\pm$ 0.62	97.32 $\pm$ 0.82	94.23 $\pm$ 0.69
Low Heterogeneity ($\beta = 1.0$)				
RingMo-lite+FedAvg [22]	1.0	97.18 $\pm$ 0.94	95.76 $\pm$ 0.87	93.37 $\pm$ 0.84
RingMo-lite+PAD [29]	1.0	98.12 $\pm$ 0.77	96.47 $\pm$ 0.79	93.91 $\pm$ 0.73
Fed-RSAdapter (ours)	1.0	99.47 $\pm$ 0.49	97.58 $\pm$ 0.67	94.46 $\pm$ 0.61

Results for $\beta = 0.5$ are directly reproduced from Table 2 for reference. A smaller β induces greater data heterogeneity across federated clients.

As β decreases from 1.0 to 0.1, client distributions become increasingly heterogeneous, and the advantage of Fed-RSAdapter correspondingly becomes more pronounced. Under high heterogeneity ($\beta = 0.1$), the performance gain of Fed-RSAdapter over RingMo-lite+FedAvg reaches 5.06%, 4.56%, and 4.32% on UCM, AID, and NWPU, respectively, exceeding the gains observed under the default $\beta = 0.5$ setting. This trend demonstrates that the similarity-aware aggregation mechanism becomes increasingly effective when client distributions diverge substantially, as the Gaussian kernel similarity matrix can more accurately differentiate client relationships and assign personalized aggregation weights. Moreover, although OA variance increases for all methods as heterogeneity intensifies, the increase is significantly smaller for Fed-RSAdapter than for RingMo-lite+FedAvg, indicating stronger cross-client stability under challenging Non-IID conditions. In contrast, when heterogeneity is reduced ($\beta = 1.0$), all methods benefit from more balanced data distributions and the performance gaps among conventional federated methods narrow, which is consistent with the common observation that uniform averaging becomes more competitive under near-IID settings. Nevertheless, Fed-RSAdapter still maintains consistent improvements over RingMo-lite+FedAvg across all three datasets, suggesting that its performance gain under low heterogeneity mainly originates from the proposed multi-scale adapter design rather than the aggregation strategy alone. Furthermore, Fed-RSAdapter consistently outperforms RingMo-lite+PAD across all β settings while transmitting only 4.93% of PAD's trainable parameters, demonstrating that the proposed framework achieves robust efficiency advantages throughout the full heterogeneity spectrum.

4. Discussion

4.1. Analysis of Performance Patterns

The experimental results across three remote sensing interpretation tasks reveal several consistent performance patterns that merit deeper analysis beyond the numerical comparisons presented in Section 3.

Task-dependent performance gains. Fed-RSAdapter achieves markedly different levels of improvement across the three tasks: an average OA gain of 3.75% over federated baselines in scene classification, 0.74% in mAP for object detection, and 0.25% in OA for semantic segmentation. This pronounced disparity can be attributed to three interacting factors. First, the degree of task head freezing differs across tasks: in classification, only a lightweight fully connected head is appended and updated, leaving the adapter modules as the primary driver of task-specific adaptation; in detection and segmentation, the RepPoints and DeepLabV3+ heads, respectively, are also frozen during federated fine-tuning, which substantially constrains the overall adaptation capacity of the model and dilutes the relative contribution of the adapter design. Second, task complexity introduces additional variability: object detection and semantic segmentation require spatial reasoning across multiple scales and dense per-pixel or per-anchor predictions, making them inherently more sensitive to the quality of the frozen task head than to adapter-level feature enhancement. Third, the spatial uniformity of the Potsdam dataset, where dominant classes such as impervious surfaces and buildings occupy the majority of pixels, compresses the OA range across all methods and makes marginal improvements less visible in absolute terms, even when class-level metrics such as mIoU and F1 reveal more substantial gains. Therefore, the framework is most beneficial for scene classification in terms of absolute accuracy improvement, while its value in object detection and semantic segmentation lies mainly in maintaining or slightly improving dense-prediction performance with much lower communication cost. This trade-off is still meaningful in federated remote sensing scenarios where transmitting full detection or segmentation models is expensive, but we do not claim that the current adapter-only design fully solves dense prediction adaptation. The additional task-head adaptation analysis in Table 7 further shows that partially trainable dense prediction heads can improve detection and segmentation accuracy at the cost of increased communication.

Advantage amplification under data scarcity. Across the three classification datasets, the performance advantage of Fed-RSAdapter over RingMo-lite+FedAvg grows consistently as the per-client training data decrease: the improvement is 4.15% on UCM (80% training ratio), 4.21% on AID (50%), and 2.96% on NWPU (20%). Although the absolute improvement is slightly smaller on NWPU due to the higher task difficulty imposed by its 45-category label space and lower per-class sample count, the relative advantage of Fed-RSAdapter over full-parameter baselines remains consistent across all three settings. This pattern confirms that the proposed adapter design and personalized aggregation strategy are particularly effective when per-client data are scarce and label distributions are highly imbalanced, which precisely characterize realistic federated remote sensing deployments on satellite or UAV platforms with limited onboard storage and heterogeneous coverage areas.

Progressive variance reduction. A consistent reduction in the standard deviation of OA across clients is observed as components are progressively added to the proposed framework. On the UCM dataset, the standard deviation decreases monotonically from ± 1.52 (SwinT+FedAvg) to ± 1.26 (RingMo-lite+FedAvg), ± 1.12 (with RSAdapter), ± 1.02 (with dense connections), ± 0.96 (with FedAvg aggregation), and finally, ± 0.62 for the full Fed-RSAdapter, representing a reduction of 59.2% in cross-client performance variance compared to the SwinT+FedAvg baseline. A similar trend is observed on AID and NWPU,

confirming that each proposed component contributes not only to improving average performance but also to enhancing cross-client consistency. This variance reduction is particularly meaningful in practical federated remote sensing scenarios, where performance stability across heterogeneous edge devices is as important as aggregate accuracy: an edge model that performs well on average but poorly on specific clients with atypical data distributions may fail to meet operational requirements in real-world deployment.

4.2. Visualization Analysis

To further verify that the proposed method effectively learns task-relevant sample features during the fine-tuning process, this part conducts a qualitative visualization analysis. Representative samples are selected from each classification dataset, and the feature activation heatmaps produced by different methods, as well as by the backbone network layers and adapter layers, are extracted and compared. The comparison focuses on the activation intensity and spatial focus of different layers, as illustrated in Figure 7.

For RingMo-lite (column b) and the frozen backbone network (column c), since no task-specific feature adaptation is performed during fine-tuning, the networks are still able to respond to coarse category-related regions. However, their attention is relatively diffuse, and the localization of discriminative regions is imprecise. In contrast, the adapter layers (column d), which are updated during the fine-tuning stage, explicitly learn features associated with the target samples. As a result, they can accurately focus on salient regions corresponding to the semantic objects of interest. The learned representations from the adapter layers further influence the final network output (column e), where the heatmaps exhibit more precise and concentrated activation over the relevant regions. These results demonstrate that the proposed method leverages adapter layers to control the output of fine-tuned sample features, enabling efficient task adaptation while preserving the original feature learning of the backbone network.

In addition, visualization of detection results on the DIOR dataset is conducted to evaluate the interpretation performance of the proposed method after fine-tuning, as shown in Figure 8. For standard scenes (column a), Fed-RSAdapter is able to effectively detect both densely and sparsely distributed targets with relatively high confidence scores. In more complex scenes with significant background clutter (column b), the detection results exhibit reduced confidence and occasional false alarms. For scenarios with large variations in target scale (column c), the method suffers from noticeable missed detections, such as small aircraft in airports or ships in harbors. These observations indicate that, although the proposed fine-tuning strategy yields consistent performance gains across standard scenarios, challenges remain in handling complex backgrounds and extreme scale variations. The missed detections in airport and harbor scenes are mainly caused by two factors. First, extremely small objects may occupy only a few pixels after feature downsampling, so part of their discriminative information can be weakened before it reaches the adapter-enhanced stages. Second, the RepPoints head is fixed in the current parameter-efficient setting, which limits the model's ability to adapt localization points and classification boundaries to client-specific small-object distributions. Thus, the proposed multi-scale adapter improves general feature adaptation but remains insufficient for extreme scale variation. Future extensions may combine adapter tuning with lightweight task-head adaptation or small-object-aware losses.

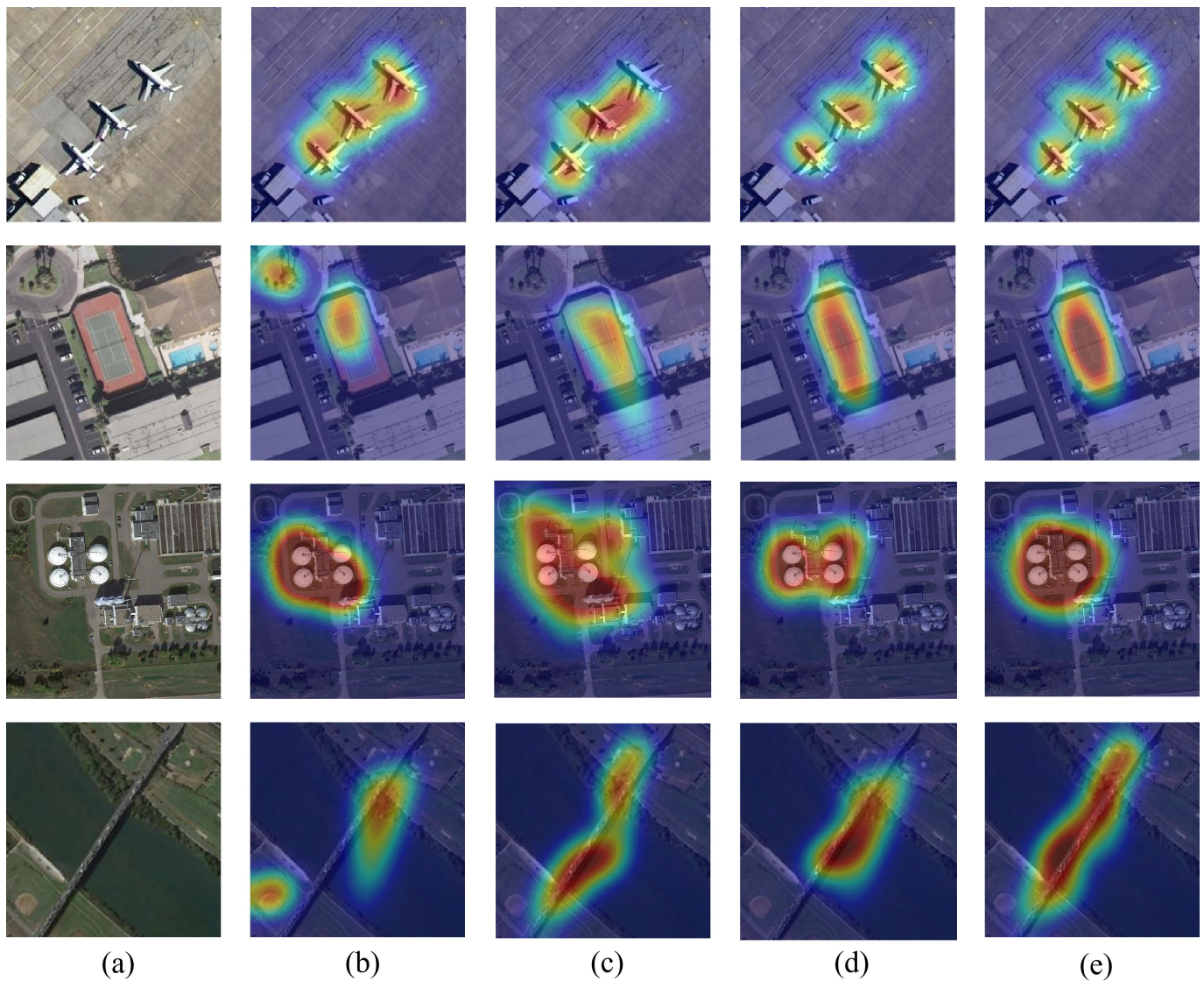


Figure 7. Feature map visualization results of the classification task at each stage. (a) Original image; (b) RingMo-lite; (c) Backbone network output features; (d) Adapter layer output features; (e) Fed-RSAdapter

To further evaluate the segmentation quality of the proposed method on the Potsdam dataset, Figure 9 presents qualitative comparisons of semantic segmentation results across three representative scenes. Orange bounding boxes highlight key regions where differences among methods are most pronounced. The three scenes are selected to represent distinct segmentation challenges: the first two rows feature dense urban structures with fine-grained object boundaries requiring precise detail segmentation, while the third row presents a road intersection scene with large homogeneous regions requiring consistent region-level classification.

In fine-grained boundary regions (rows 1–2), Fed-RSAdapter produces substantially cleaner predictions with more coherent boundary delineation compared to RingMo-lite+FedAvg, which exhibits fragmented misclassifications and over-segmentation artifacts, and RingMo-lite+RSAdapter, which partially recovers boundaries but still misclassifies small vehicle instances. In the large homogeneous region scene (row 3), Fed-RSAdapter achieves the most spatially coherent predictions with minimal spurious classifications, attributable to the similarity-aware personalized aggregation aligning global representations with local data distributions. These qualitative results are consistent with the quantitative

findings in Table 4, and notably the improvements are more salient in mIoU and F1 than in OA, further underscoring the limitation of overall accuracy for evaluating segmentation quality on class-imbalanced datasets.

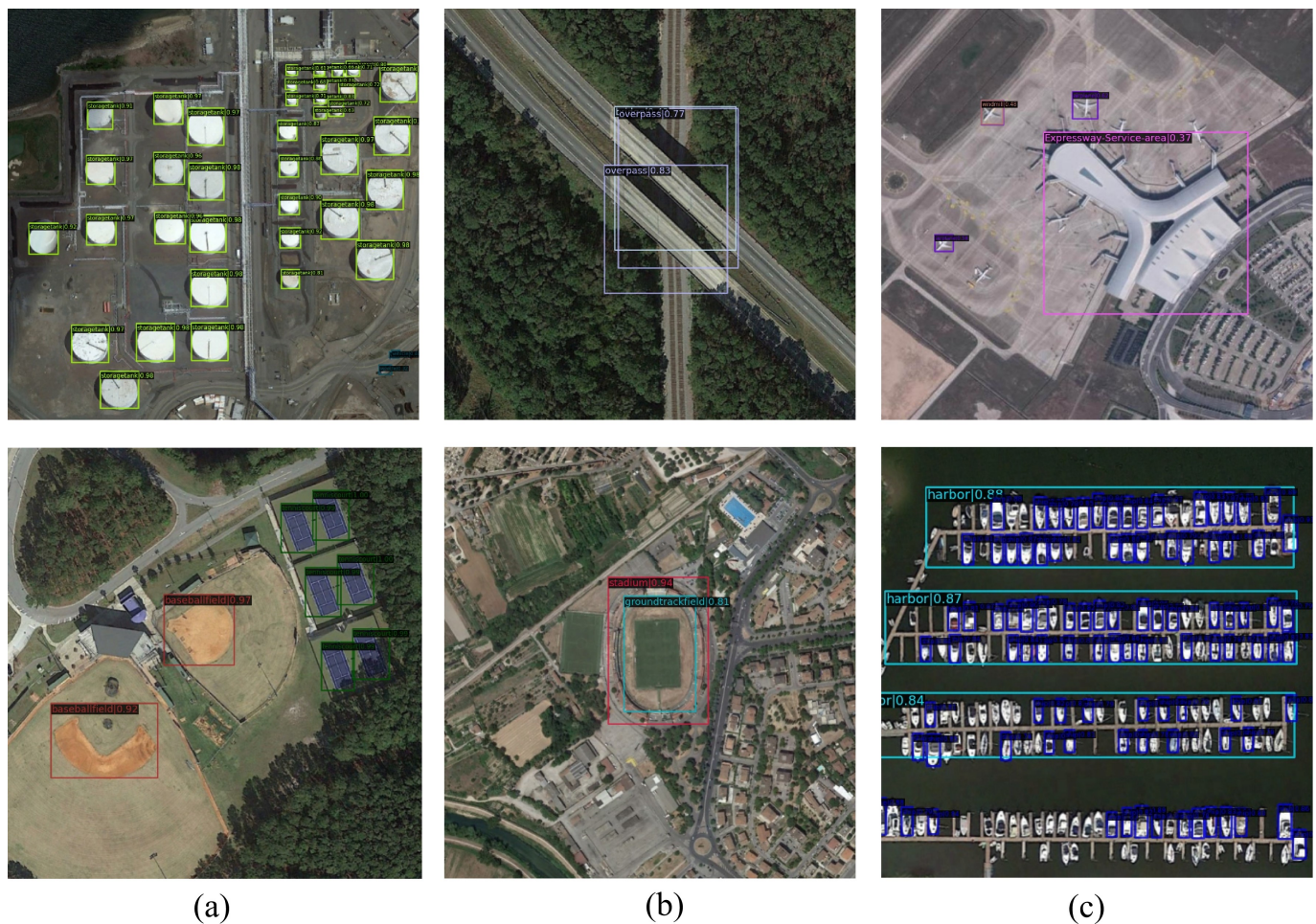


Figure 8. Visualization results of the Fed-RSAdapter detection task in the DIOR dataset. (a) Normal processing result; (b) False alarm result; (c) Missed detection result

4.3. Limitations

Despite the promising results demonstrated across multiple tasks and datasets, several limitations of the proposed Fed-RSAdapter framework should be acknowledged.

Constrained adaptation in dense prediction tasks. The performance improvement of Fed-RSAdapter is considerably more modest for object detection (+0.74% mAP) and semantic segmentation (+0.25% OA) compared to scene classification (+3.75% OA). A primary contributor to this limitation is the freezing of task-specific heads (RepPoints for detection, DeepLabV3+ for segmentation) during federated fine-tuning, which prevents the model from fully adapting its prediction layers to client-specific data distributions. While freezing the task head is necessary to maintain the lightweight communication profile of the framework, it fundamentally limits the expressiveness of client-side adaptation for spatially complex tasks. The supplementary task-head adaptation experiment confirms that partially trainable heads can improve dense prediction performance, but the improvement is accompanied by additional communication overhead. Future work may further optimize low-rank or sparsely updated task-head designs to expand adaptation capacity while keeping the communication cost close to the adapter-only setting. In addition, although the proposed method substantially reduces transmitted parameter volume and shows lower training memory and runtime in the A40-based profiling experiment, the experiments

in this paper are still conducted on server-grade GPUs for controlled comparison. We therefore regard actual runtime, memory, and energy evaluation on embedded platforms such as Jetson or RK3588 as future work. The present conclusions should be interpreted as evidence for communication-efficient and training-efficient federated fine-tuning, not as direct proof of end-to-end deployment efficiency on all edge hardware.

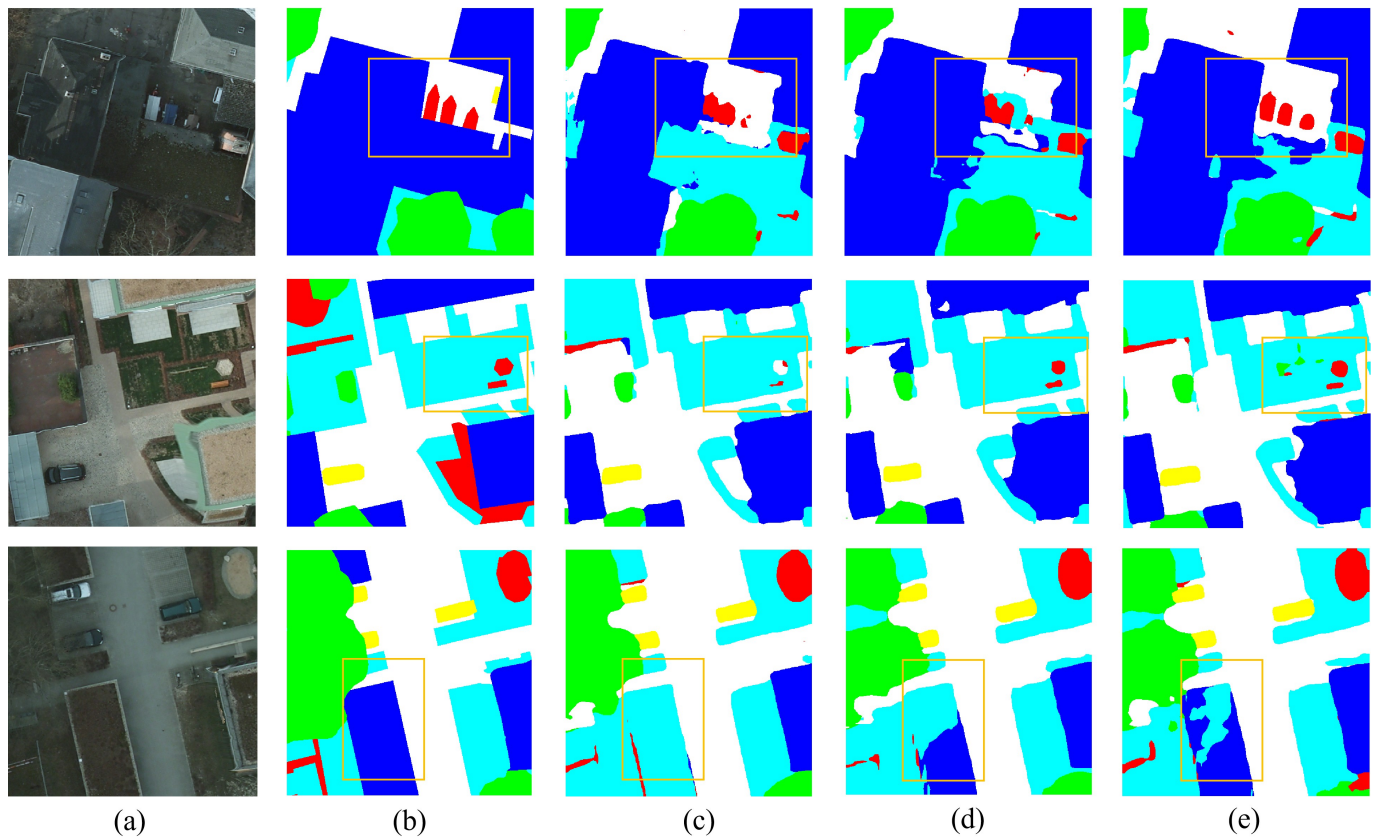


Figure 9. Visualization results of the semantic segmentation task on the Potsdam dataset. (a) Original image; (b) Ground truth; (c) RingMo-lite+FedAvg; (d) RingMo-lite+RSAadapter; (e) Fed-RSAadapter. Orange boxes highlight regions of notable difference. Colors denote land-cover classes: ■ impervious surface, ■ low vegetation, ■ building, ■ car, ■ tree, others is clutter/background.

Quadratic complexity of similarity matrix computation. The parameter similarity-aware aggregation strategy requires the central server to compute pairwise distances among all K client adapter parameter vectors, resulting in $\mathcal{O}(K^2)$ complexity for constructing the similarity matrix at each communication round. In the current experimental setting with $K = 10$ clients, this overhead is negligible relative to the cost of local training. However, in large-scale federated deployments involving tens or hundreds of satellite or UAV platforms, the server-side computational burden may become non-trivial, particularly when communication rounds are frequent. Potential solutions include approximate nearest-neighbor methods for sparse similarity graph construction, or clustering-based aggregation that groups clients by parameter similarity prior to weighted averaging, reducing the effective number of pairwise comparisons.

5. Conclusions

This paper proposes a federated fine-tuning framework for remote sensing imagery based on multi-scale adapters. First, to enable efficient feature adaptation on remote sensing edge devices, a multi-scale adapter-based fine-tuning module is introduced at the edge side. By aligning and densely connecting the outputs of adapter layers at different stages,

the proposed design enhances the capability of adapters to capture discriminative remote sensing features across multiple spatial scales, while preserving the knowledge embedded in the pretrained backbone. Second, to address the difficulty of parameter aggregation during the fine-tuning process, a parameter similarity sharing module is introduced at the central server. The proposed module extends the parameter comparison between local and global models to similarity estimation among all participating clients. By leveraging parameter similarity to reflect data heterogeneity across edge devices, the aggregation weights of returned parameters are adaptively optimized, enabling each edge model to maintain global generalization ability while effectively strengthening task-specific and personalized representations.

Extensive experiments are conducted on scene classification (UCM, AID, and NWPU), object detection (DIOR), and semantic segmentation (Potsdam) tasks. Experimental results demonstrate that the proposed federated fine-tuning method achieves a reduction of trainable parameters by 95.07% in scene classification, with an average OA improvement of 3.75% over federated learning baselines. For object detection, the trainable parameters are reduced by 96.22%, while the mAP is improved by 0.74%. In semantic segmentation, the trainable parameter reduction reaches 96.62%, accompanied by an OA improvement of 0.25% compared with baseline federated learning methods. Overall, across all tasks, the proposed federated fine-tuning strategy yields an average performance gain of 1.57% over full-parameter training, demonstrating its effectiveness in improving interpretation performance under constrained communication and computation conditions. However, the magnitude of improvement differs across tasks. The largest gain is observed in scene classification, whereas object detection and semantic segmentation show smaller absolute improvements because their frozen dense-prediction heads constrain task-specific adaptation. Additional partially trainable head variants show that stronger dense-prediction adaptation is possible with higher communication cost. Future work will investigate more communication-efficient lightweight heads and actual embedded-platform evaluation to further validate the practical deployment potential of Fed-RSAdapter.

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

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The datasets used in this paper are publicly available. The UCM dataset is available at https://figshare.com/articles/dataset/UCM_image_dataset/6085976?file=10967885 (accessed on 8 July 2026), the AID dataset at <https://captain-whu.github.io/AID/> (accessed on 8 July 2026), the NWPU dataset at <https://www.kaggle.com/datasets/happyang/nwpu-data-set> (accessed on 8 July 2026), the DIOR dataset at <https://gcheng-nwpu.github.io/#Datasets> (accessed on 8 July 2026), and the Potsdam dataset at <https://www.isprs.org/resources/datasets/benchmarks/UrbanSemLab/2d-sem-label-potsdam.aspx> (accessed on 8 July 2026).

Acknowledgments: The authors would like to thank the Shanghai Institute of Satellite Engineering and Aerospace Information Research Institute, Chinese Academy of Sciences for providing computational resources and support for this research.

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

References

1. Yang, X.; Zhang, S.; Duan, S.; Yang, W. An effective and lightweight hybrid network for object detection in remote sensing images. *IEEE Trans. Geosci. Remote Sens.* **2023**, *62*, 5600711.
2. Wang, Y.; Zhang, T.; Zhao, L.; Hu, L.; Wang, Z.; Niu, Z.; Cheng, P.; Chen, K.; Zeng, X.; Wang, Z.; et al. Ringmo-lite: A remote sensing lightweight network with cnn-transformer hybrid framework. *IEEE Trans. Geosci. Remote Sens.* **2024**, *62*, 5608420.
3. Hu, L.; Yu, H.; Lu, W.; Yin, D.; Sun, X.; Fu, K. Airs: Adapter in remote sensing for parameter-efficient transfer learning. *IEEE Trans. Geosci. Remote Sens.* **2024**, *62*, 5605218.
4. Chen, S.; Ge, C.; Tong, Z.; Wang, J.; Song, Y.; Wang, J.; Luo, P. Adaptformer: Adapting vision transformers for scalable visual recognition. *Adv. Neural Inf. Process. Syst.* **2022**, *35*, 16664–16678.
5. Pu, X.; Jia, H.; Zheng, L.; Wang, F.; Xu, F. Classwise-sam-adapter: Parameter efficient fine-tuning adapts segment anything to sar domain for semantic segmentation. *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.* **2025**, *18*, 4791–4804.
6. Wang, Y.; Ghamisi, P. Rsadapter: Adapting multimodal models for remote sensing visual question answering. *IEEE Trans. Geosci. Remote Sens.* **2024**, *62*, 5628313.
7. Yu, B.X.; Chang, J.; Wang, H.; Liu, L.; Wang, S.; Wang, Z.; Lin, J.; Xie, L.; Li, H.; Lin, Z.; et al. Visual tuning. *ACM Comput. Surv.* **2024**, *56*, 1–38.
8. Houshy, N.; Giurgiu, A.; Jastrzebski, S.; Morrone, B.; De Laroussilhe, Q.; Gesmundo, A.; Attariyan, M.; Gelly, S. Parameter-efficient transfer learning for NLP. In *Proceedings of the International Conference on Machine Learning*; PMLR: Cambridge, MA, USA, 2019; pp. 2790–2799.
9. Bapna, A.; Arivazhagan, N.; Firat, O. Simple, scalable adaptation for neural machine translation. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing*; Association for Computational Linguistics: Kerrville, TX, USA, 2019; pp. 1538–1548.
10. Liu, Z.; Lin, Y.; Cao, Y.; Hu, H.; Wei, Y.; Zhang, Z.; Lin, S.; Guo, B. Swin transformer: Hierarchical vision transformer using shifted windows. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, Montreal, QC, Canada, 11–17 October 2021; pp. 10012–10022.
11. Hu, E.J.; Shen, Y.; Wallis, P.; Allen-Zhu, Z.; Li, Y.; Wang, S.; Wang, L.; Chen, W. Lora: Low-rank adaptation of large language models. *Int. Conf. Learn. Represent.* **2022**, *1*, 3.
12. Zhang, Q.; Chen, M.; Bukharin, A.; Karampatziakis, N.; He, P.; Cheng, Y.; Chen, W.; Zhao, T. Adalora: Adaptive budget allocation for parameter-efficient fine-tuning. *Int. Conf. Learn. Represent. arXiv* **2023**, arXiv:2303.10512.
13. Dettmers, T.; Pagnoni, A.; Holtzman, A.; Zettlemoyer, L. Qlora: Efficient finetuning of quantized llms. *Adv. Neural Inf. Process. Syst.* **2023**, *36*, 10088–10115.
14. Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A.N.; Kaiser, Ł.; Polosukhin, I. Attention is all you need. *Adv. Neural Inf. Process. Syst.* **2017**, *30*, 5998–6008.
15. Zhang, Z.; Liu, F.; Liu, C.; Tian, Q.; Qu, H. ACTNet: A dual-attention adapter with a CNN-transformer network for the semantic segmentation of remote sensing imagery. *Remote Sens.* **2023**, *15*, 2363–2379.
16. Shamsolmoali, P.; Zareapoor, M.; Zhou, H.; Wang, R.; Yang, J. Road segmentation for remote sensing images using adversarial spatial pyramid networks. *IEEE Trans. Geosci. Remote Sens.* **2020**, *59*, 4673–4688.
17. Liu, R.; Tao, F.; Liu, X.; Na, J.; Leng, H.; Wu, J.; Zhou, T. RANet: A residual ASPP with attention framework for semantic segmentation of high-resolution remote sensing images. *Remote Sens.* **2022**, *14*, 3109.
18. Hu, Z.; Wang, L.; Lan, Y.; Xu, W.; Lim, E.P.; Bing, L.; Xu, X.; Poria, S.; Lee, R.K.W. Llm-adapters: An adapter family for parameter-efficient fine-tuning of large language models. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*; Association for Computational Linguistics: Kerrville, TX, USA, 2023; pp. 5254–5276.
19. Yin, D.; Hu, L.; Li, B.; Zhang, Y. Adapter is all you need for tuning visual tasks. *arXiv* **2023**, arXiv:2311.15010.
20. Zhang, C.; Xie, Y.; Bai, H.; Yu, B.; Li, W.; Gao, Y. A survey on federated learning. *Knowl.-Based Syst.* **2021**, *216*, 106775–106785.
21. Büyüktas, B.; Sumbul, G.; Demir, B. Federated learning across decentralized and unshared archives for remote sensing image classification: A review. *IEEE Geosci. Remote Sens. Mag.* **2024**, *12*, 64–80.
22. McMahan, B.; Moore, E.; Ramage, D.; Hampson, S.; y Arcas, B.A. Communication-efficient learning of deep networks from decentralized data. In *Proceedings of the Artificial Intelligence and Statistics*; PMLR: Cambridge, MA, USA, 2017; pp. 1273–1282.
23. Li, T.; Sahu, A.K.; Zaheer, M.; Sanjabi, M.; Talwalkar, A.; Smith, V. Federated optimization in heterogeneous networks. *Proc. Mach. Learn. Syst.* **2020**, *2*, 429–450.
24. Karimireddy, S.P.; Kale, S.; Mohri, M.; Reddi, S.; Stich, S.; Suresh, A.T. Scaffold: Stochastic controlled averaging for federated learning. In *Proceedings of the International Conference on Machine Learning*; PMLR: Cambridge, MA, USA, 2020; pp. 5132–5143.
25. T Dinh, C.; Tran, N.; Nguyen, J. Personalized federated learning with moreau envelopes. *Adv. Neural Inf. Process. Syst.* **2020**, *33*, 21394–21405.
26. Collins, L.; Hassani, H.; Mokhtari, A.; Shakkottai, S. Exploiting shared representations for personalized federated learning. In *Proceedings of the International Conference on Machine Learning*; PMLR: Cambridge, MA, USA, 2021; pp. 2089–2099.

27. Zhang, X.; Zhang, B.; Yu, W.; Kang, X. Federated deep learning with prototype matching for object extraction from very-high-resolution remote sensing images. *IEEE Trans. Geosci. Remote Sens.* **2023**, *61*, 5603316.
28. Wang, Y.; Wang, Z.; Cheng, P.; Zeng, X.; Wang, H.; Sun, X.; Fu, K. DCM: A distributed collaborative training method for the remote sensing image classification. *IEEE Trans. Geosci. Remote Sens.* **2023**, *61*, 5605018.
29. Wang, S.; Li, J.; Liu, Z.; Gong, M.; Zhang, Y.; Zhao, Y.; Deng, B.; Zhou, Y. Personalized multi-party few-shot learning for remote sensing scene classification. *IEEE Trans. Geosci. Remote Sens.* **2024**, *62*, 4506115.
30. Han, P.; Wang, S.; Leung, K.K. Adaptive gradient sparsification for efficient federated learning: An online learning approach. In *Proceedings of the 2020 IEEE 40th International Conference on Distributed Computing Systems (ICDCS)*; IEEE: New York, NY, USA, 2020; pp. 300–310.
31. Shlezinger, N.; Chen, M.; Eldar, Y.C.; Poor, H.V.; Cui, S. UVeQFed: Universal vector quantization for federated learning. *IEEE Trans. Signal Process.* **2020**, *69*, 500–514.
32. Yosinski, J.; Clune, J.; Bengio, Y.; Lipson, H. How transferable are features in deep neural networks? *Adv. Neural Inf. Process. Syst.* **2014**, *27*, 1–9.
33. Yang, Y.; Newsam, S. Bag-of-visual-words and spatial extensions for land-use classification. In *Proceedings of the 18th SIGSPATIAL International Conference on Advances in Geographic Information Systems*, San Jose, CA, USA, 2–5 November 2010; pp. 270–279.
34. Xia, G.S.; Hu, J.; Hu, F.; Shi, B.; Bai, X.; Zhong, Y.; Zhang, L.; Lu, X. AID: A benchmark data set for performance evaluation of aerial scene classification. *IEEE Trans. Geosci. Remote Sens.* **2017**, *55*, 3965–3981.
35. Cheng, G.; Han, J.; Lu, X. Remote sensing image scene classification: Benchmark and state of the art. *Proc. IEEE* **2017**, *105*, 1865–1883.
36. Li, K.; Wan, G.; Cheng, G.; Meng, L.; Han, J. Object detection in optical remote sensing images: A survey and a new benchmark. *ISPRS J. Photogramm. Remote Sens.* **2020**, *159*, 296–307.
37. Hsu, T.M.H.; Qi, H.; Brown, M. Measuring the effects of non-identical data distribution for federated visual classification. *arXiv* **2019**, arXiv:1909.06335.
38. Yang, Z.; Liu, S.; Hu, H.; Wang, L.; Lin, S. Reppoints: Point set representation for object detection. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, Seoul, Republic of Korea, 27 October–2 November 2019; pp. 9657–9666.
39. Chen, L.C.; Zhu, Y.; Papandreou, G.; Schroff, F.; Adam, H. Encoder-decoder with atrous separable convolution for semantic image segmentation. In *Proceedings of the European Conference on Computer Vision*, Munich, Germany, 8–14 September 2018; pp. 801–818.

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.