MVP: Meta Visual Prompt Tuning for Few-Shot Remote Sensing Image Scene Classification

Junjie Zhu, Yiying Li, Ke Yang, Naiyang Guan, Zunlin Fan, Chunping Qiu, and Xiaodong Yi

***Abstract*— Vision transformer (ViT) models have recently emerged as powerful and versatile tools for various visual tasks. In this article, we investigate ViT in a more challenging scenario within the context of few-shot conditions. Recent work has achieved promising results in few-shot image classification using pretrained ViT models. However, this work uses full fine-tuning for the downstream tasks, leading to significant overfitting and storage issues, especially in the remote sensing domain. To tackle these issues, we turn to the recently proposed parameter-efficient tuning (PETuning) methods, which update only the newly added parameters while keeping the pretrained backbone frozen. Inspired by these methods, we propose the meta visual prompt tuning (MVP) method. Specifically, we integrate the prompt-tuning-based PETuning method into the meta-learning framework and tailor it for remote sensing datasets, resulting in an efficient framework for few-shot remote sensing scene classification (FS-RSSC). Moreover, we introduce a novel data augmentation scheme that exploits patch embedding recombination to enhance data diversity and quantity. This scheme is generalizable to any network that uses the ViT architecture as its backbone. Experimental results on the FS-RSSC benchmark demonstrate the superior performance of the proposed MVP over existing methods in various settings, including various-way-various-shot, various-way-one-shot, and cross-domain adaptation.**

***Index Terms*— Few-shot learning, meta-learning, parameter-efficient fine-tuning, prompt tuning, remote sensing.**

I. Introduction

FEW-SHOT remote sensing scene classification (FS-RSSC) [1], [2] aims to classify remote sensing images into different categories using only a few labeled examples per category. This is a challenging but important machine learning task for practical applications such as land use classification and environmental monitoring, where obtaining large-scale and high-quality labeled datasets is expensive and time-consuming. Transfer learning [3], meta-learning [4], and metric learning [1] have been used for FS-RSSC tasks. These methods primarily use convolutional neural networks (CNNs) that are typically restricted to smaller models with fewer parameters, such as Conv4 [5], ResNet12 [6], and ResNet18 [6].

Manuscript received 20 July 2023; revised 12 December 2023; accepted 16 January 2024. Date of publication 29 January 2024; date of current version 23 February 2024. This work was supported in part by the National Natural Science Foundation of China under Grant 62006241, Grant 91948303, Grant 62206307, Grant 62106280, and Grant 42201513; and in part by the Science and Technology Innovation 2030 Major Project under Grant 2019AAA0104800. *(Junjie Zhu and Yiying Li contributed equally to this work.) (Corresponding authors: Ke Yang; Naiyang Guan; Xiaodong Yi.)*

The authors are with the National Innovation Institute of Defense Technology, Beijing 100071, China, and also with the Intelligent Game and Decision Lab (IGDL), Beijing 100094, China (e-mail: yangke13@nudt.edu.cn; nyguan@sina.com; xdong_yi@163.com).

Digital Object Identifier 10.1109/TGRS.2024.3359599

Recently, vision transformers (ViTs) have yielded remarkable achievements in various visual tasks. The prospect of applying them to the field of few-shot learning is highly attractive and holds significant appeal. For instance, PMF [9], a ViT-based method consisting of a three-stage learning pipeline, has made significant progress in few-shot classification tasks. PMF pretrained the ViT model on unsupervised external data, then meta-trained the model on base categories, and finally, fine-tuned the model on a novel task. They showed that this simple transformer-based pipeline yields surprisingly good performance on standard benchmarks such as Mini-ImageNet [5], CIFAR-FS [10], CDFSL [11], and Meta-Dataset [12].

During the three stages of learning, PMF uses full fine-tuning to update network weights. However, this brings about two significant issues, especially in remote sensing applications. First, compared with the few-shot classification task in natural images, the remote sensing domain faces a more severe issue of sample scarcity, and thus, fully fine-tuning ViT with a large number of weights can lead to severe overfitting problems. Second, training a ViT model for each remote sensing task is unfeasible due to storage limitations on the satellite or drone platforms where the algorithm is deployed. Our experiment results also indicate that ViT models based on a full fine-tuning strategy exhibit lower efficacy in solving FS-RSSC tasks.

To address these issues, we turn to explore a fine-tuning method for ViT models that is suitable for FS-RSSC tasks. A potential solution is the parameter-efficient tuning (PETuning) [13] methods, which have received considerable attention in natural image recognition recently. In the PETuning paradigm, only a small number of newly added parameters are updated during training, while the pretrained backbone is kept frozen. For instance, visual prompt tuning (VPT) [14] is a recently proposed visual PETuning method that adds prompt tokens to the input space and only updates the newly added parameters. The tuned parameters for each downstream task are less than 1% of model parameters, and thus, it can reduce the storage demand and effectively alleviate the model overfitting issues of remote sensing applications.

Taking inspiration from VPT, we propose the meta visual prompt tuning (MVP) method as an effective approach to address FS-RSSC tasks. Within the framework of meta-learning, MVP leverages prompt tuning to adapt a pretrained ViT model to new tasks with limited data and computational resources. Unlike previous methods that fine-tune the entire ViT model, MVP only updates the newly added prompt parameters while keeping the pretrained ViT backbone

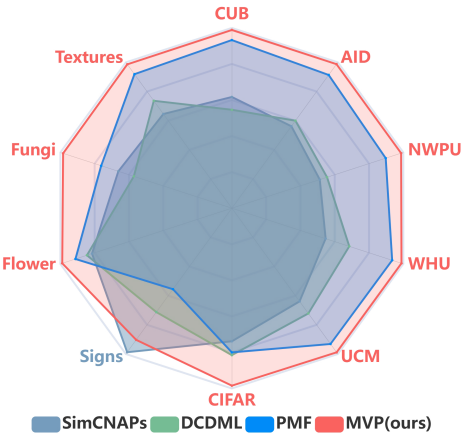


Fig. 1. MVP versus SoTA FS-RSSC algorithms (i.e., SimCNAPs [7], DC-DML [8], PMF [9]): a succinct assessment of their accuracy performance on the AIFS-DATASET [8].

networks fixed. Specifically, MVP embeds the prompt parameters into a novel parameter-efficient meta-learning framework. In the meta-training phase, MVP learns to learn a good initialization for the newly added prompt parameters on multiple sets of FS-RSSC source tasks. We denote the optimal initialization prompt parameters by $\boldsymbol{\theta}$. In the meta fine-tuning phase, MVP fine-tunes $\boldsymbol{\theta}$ with a few gradient steps on the target task and then makes predictions for remote sensing scene categories.

In addition, we design a novel data augmentation method for meta fine-tuning that is based on the ViT architecture. Our method is motivated by the observation that remote sensing scene images of the same category tend to have high consistency, which may lead to model overfitting and poor generalization to variations in imaging conditions. To address this issue, we propose to enhance the diversity of remote sensing scenes by embedding image patches of other categories into the current image, inspired by image patch recombination research [15]. Specifically, a new image patch recombination method based on the ViT network is designed, which operates on image patch embeddings after the linear projection of ViT. Moreover, we randomly select and swap some patch embeddings of an input image with those from other images in the same batch. It is worth noting that this data augmentation method is also applicable to all other models based on the ViT architecture. Experiments show that our data augmentation method can effectively enhance the generalization performance to new categories in the meta fine-tuning stage.

Real-world FS-RSSC tasks exhibit two key characteristics: the number of categories and samples in new tasks varies, and the data distribution is unpredictable, thus necessitating cross-domain adaptation [8]. The large-scale FS-RSSC benchmark AIFS-DATASET [8] meets both the criteria as the benchmark for evaluation. We thoroughly evaluated our proposed MVP model on the AIFS-DATASET through comprehensive experiments, under various-way-various-shot, various-way-one-shot, and cross-domain adaptation settings. Fig. 1 shows an overview of the comparison of the results and Fig. 2 depicts the pipeline of the MVP model. Our MVP demonstrated superior performance on various challenging in-domain and cross-domain benchmarks, significantly surpassing the existing methods. Our contributions can be summarized as follows.

1) To the best of our knowledge, our proposed MVP is the first study to explore PETuning on remote sensing applications.
2) We integrate PETuning into the meta-learning framework using the meta-learning paradigm to initialize newly added prompt parameters, facilitating rapid adaptation to new FS-RSSC tasks.
3) Our proposed MVP empowers transformer models to perform well in situations where there are only limited data available, significantly alleviating the overfitting issue.
4) We propose a data augmentation method tailored explicitly for the ViT-based models to enhance their adaptability to remote sensing scenes.
5) Our MVP demonstrates exceptional performance on the challenging FS-RSSC dataset.

II. Related Work

A. Scene Classification With Few-Shot Learning

In practical scenarios, well-labeled training data are limited, and FS-RSSC attempts to solve this challenge. Classical methods can be roughly divided into two categories: metric-based and meta-learning-based [16]. The metric-based methods learn a feature space or distance function to measure the similarity between classes [1], [17], [18]. For instance, RS-MetaNet [17] introduces a novel balance loss to provide better linear segmentation planes for scenes in different categories. Another example is DLA-MatchNet [18], which proposes an approach to automatically discover discriminative regions. MCMNet [19] takes this a step further by proposing a multiscale covariance network to optimize the manifold space.

On the other hand, meta-learning-based methods [20], [21] aim to learn a meta-model that can quickly adapt to new tasks with few gradient updates. For instance, MetaRS [21] explores the use of meta-learning to improve the generalization capability of deep neural networks (DNNs) on remote sensing scene classification with limited training data. Another example is PTMeta [22], which applies parameter transfer to fix the parameters in a DNN to relax the problem of training a large number of parameters within a meta-learning framework.

A review of these methods reveals that most use shallow CNNs as their backbone network. While this approach can mitigate overfitting when training data are limited, it also constrains further improvements in classification performance. Recently, ViTs have demonstrated promising results in visual tasks [9], [23], and there have been efforts to apply ViT-based approaches to large-scale remote sensing classification [24], [25]. However, research on the application of ViT-based methods to FS-RSSC tasks remains limited.

B. Efficient Tuning for Visual Transformer

One of the key challenges in machine learning is how to effectively reuse the existing models and fine-tune them for downstream tasks. Conventionally, when the task data distribution matches the pretraining data distribution, the model training can be done by freezing the backbone network and

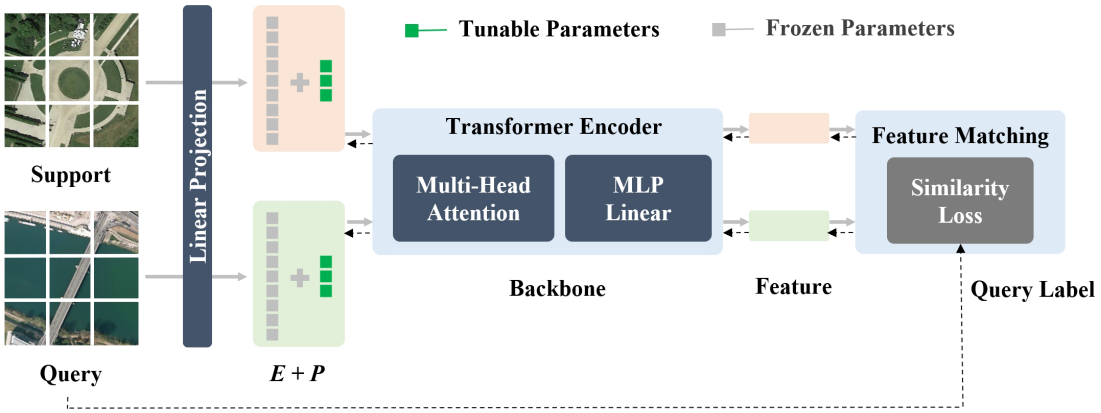


Fig. 2. Meta prompt tuning framework for remote sensing scene classification. Query in the dashed box: actual image in meta-training and pseudoquery generated via a novel data augmentation method in meta fine-tuning. "E" represents original backbone embedding tokens and "P" represents newly added prompt tokens.

only fine-tuning the classification head [3]. However, when the downstream task data distribution differs significantly, the main fine-tuning method is to update all the parameters of the model. Recent work also explores adding new network structures (such as FiLM layers [7]) on top of the backbone network and then fully fine-tuning all the model parameters for the downstream task.

PETuning [26], [27] is an effective way to adapt large-scale transformer models to downstream tasks with minimal parameter size, data size, and storage space. PETuning addresses the challenge of data scarcity that hinders the full fine-tuning of large models. Depending on the data distribution of the task, different PETuning variants can be used. For instance, sparse fine-tuning [28] only updates the bias terms of the model when the task data are similar to the pretraining data. For tasks with different data distributions, side-tuning [29], [30] and prompt-tuning [14] are suitable methods. These methods freeze the whole model backbone and only fine-tune a small fraction (usually less than 1%) of newly added parameters, which are plug-and-play and do not alter the original model structure and parameters. Side-tuning (e.g., adapter [31] and LoRA [32]) uses the same input as the original model and combines the output of the original model with the output of the side module to form new feature representations. The prompt-tuning-based methods can change the input of each transformer block by adding special tokens to induce different feature activations. In this article, we propose a novel PETuning method that leverages meta-learning to adapt large-scale transformer models to the FS-RSSC task.

C. Data Augmentation

In both general and few-shot image classification, data augmentation expands the number of available images per class and generates novel classes and tasks [33]. Techniques range from simple rotations [34] and crops [35] to more refined strategies such as cutmix [36] and mixup [37]. Some methods use GAN networks to emulate the target data distribution [38]. Recent research has shown that data augmentation has different effects on the meta-training and meta-testing stages of the meta-learning pipeline [39]. For instance, increasing the number of query samples and tasks during meta-testing improves the performance of meta-learners more than increasing the number of support samples during meta-training. Compared with the general few-shot datasets, the FS-RSSC datasets have a smaller volume [1]. To overcome this data scarcity problem, several methods have been proposed to augment the training data for FS-RSSC in different ways. For example, the quad-patch method [15] generates synthetic samples by cutting and reassembling patches from existing images, while the spatial vector enhancement method [40] simulates the distribution of neighboring classes to enrich the feature space. However, these methods do not consider the specific properties of ViT as the backbone network. In this article, we present a novel data augmentation method that is customized for the structural features of ViT and the attributes of remote sensing images and fully unleashes the potential of the ViT architecture.

III. Methodology

A. Overview

1) Problem Definition: FS-RSSC is a task that requires a model to quickly and accurately classify unseen scene images with only a few annotated samples [1]. This task is motivated by the challenge of domain adaptation in remote sensing images, which are often collected from different sensors, regions, and seasons, resulting in a large domain gap between the source and target domains. Moreover, the number of annotated samples varies significantly across different tasks, making it necessary to train models that can cope with different numbers of annotated samples. Formally, the annotated dataset, referred to as the support set S, consists of C categories (way) with K samples (shot) per category, where $C \in [5, \text{MAXWAY}]$ and $K \in [1, \text{MAXSHOT}]$. The model is trained on the support set and then used to predict the categories of the query set Q, which contains unlabeled images from the same categories as the support set.

2) Meta-Learning Process: Meta-learning methods have shown promise for the FS-RSSC task. The basic meta-learning process consists of two stages: meta-training and meta fine-tuning [4], [21]. With the development of ViT, the PMF [9]

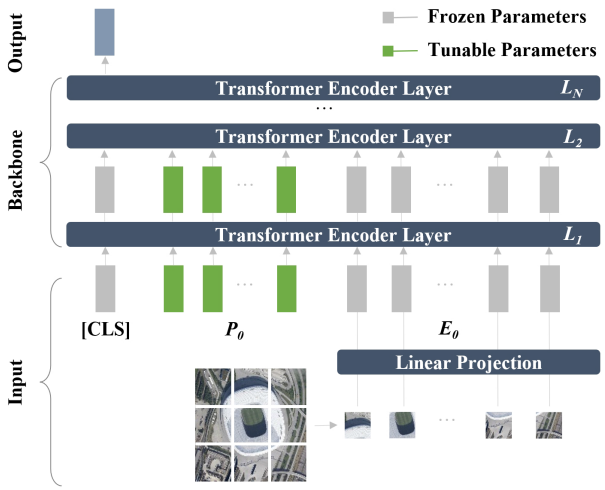


Fig. 3. Network structure of the proposed MVP model.

method enhances the meta-learning process by adding a pretraining phase. PMF proposes a new three-stage pipeline: pretraining, meta-training, and meta fine-tuning. The backbone network is first pretrained on a large-scale external dataset such as ImageNet [41], then meta-trained on multiple source datasets, and finally, meta fine-tuned on a target dataset with limited annotated support samples. Note that the source and target datasets have nonoverlapping domains.

3) Prompt-Based Meta-Learning: During the three stages of learning, PMF uses full fine-tuning to update the backbone network. However, this brings about two significant challenges, especially in remote sensing applications. One challenge is how to avoid overfitting when fine-tuning the whole ViT model on limited support samples. Another challenge is how to reduce the storage space required to store different ViT models for different tasks. To address these issues, we propose a meta-visual prompt tuning (MVP) framework. Our research aims to efficiently fine-tune the pretrained ViT backbone models for the FS-RSSC task.

Given a pretrained ViT backbone network, MVP introduces a new prompt module into the input space of the pretrained ViT model (as shown in Fig. 3). The prompt parameters do not need to undergo the pretraining stage as in PMF. During the meta-training and meta fine-tuning stages, MVP only updates the prompt parameters to fit the FS-RSSC task features, while the whole ViT network is frozen.

In the meta-training stage of MVP, we optimize the prompt parameters on multiple source datasets that are domain-disjoint from the target dataset. We use a meta-learning algorithm that mimics the FS-RSSC task by sampling episodes from the source datasets. An episode is a few-shot learning task that contains a support set and a query set with the same categories but different images. We use a prototypical loss function [42] to evaluate the classification performance of the MVP model on the query set and update the prompt parameters using gradient descent.

After initializing the prompt parameters in meta-training, the MVP model is able to adapt to the target dataset of remote sensing scenes in the meta fine-tuning phase. These target scenes are completely new and different from the source dataset. The MVP model performs meta fine-tuning on auxiliary tasks that are based on support data, using a novel data augmentation method. After meta fine-tuning, the MVP model can classify all the remaining unlabeled query data.

B. Model Architecture

1) ViT Backbone Networks: The standard ViT [23] is used as our backbone network to address the FS-RSSC task. As input to the ViT backbone network, the image $x \in \mathbb{R}^{3\times H\times W}$ is initially partitioned into m fixed-size patches $\{I_j \in \mathbb{R}^{3\times h\times w} \mid j \in \mathbb{N}, 1 \le j \le m\}$. Subsequently, each patch is projected into d-dimensional feature embedding with positional encoding [23]

$$\boldsymbol{e}_0^j = \texttt{Embed}\left(I_j\right) \tag{1}$$

where $\boldsymbol{e}_0^j \in \mathbb{R}^d$. Following this, the collection of image patch embeddings $\boldsymbol{E}_i = \{\boldsymbol{e}_i^j \in \mathbb{R}^d \mid i \in \mathbb{N}, 1 \le i \le N\}$ is used as the inputs to the $(i+1)$th transformer layer L_{i+1}. Formally, the entire ViT backbone networks can be articulated as

$$[\boldsymbol{CLS}_i, \boldsymbol{E}_i] = L_i\left(\left[\boldsymbol{CLS}_{i-1}, \boldsymbol{E}_{i-1}\right]\right) \tag{2}$$

$$f_\theta(x) = \boldsymbol{CLS}_N \tag{3}$$

where $\boldsymbol{CLS}_{i-1} \in \mathbb{R}^d$ refers to the class token in the input sequence of L_i. Furthermore, $\boldsymbol{CLS}_N$ in the output of the final layer is used as the feature representation $f(x)$ of the input image x. Moreover, θ represents the model parameters of ViT.

2) Prompt-Based ViT Networks: In line with the VPT [14] model and given a pretrained ViT backbone network, a set of prompt tokens $\boldsymbol{P}_i = \{\boldsymbol{p}_i^t \in \mathbb{R}^d \mid t \in \mathbb{N}, 1 \le t \le p\}$ are concatenated into the input space of the transformer layer. Formally, the ViT architecture in (2) can be replaced by

$$[\boldsymbol{CLS}_i, \boldsymbol{P}_i, \boldsymbol{E}_i] = L_i\left(\left[\boldsymbol{CLS}_{i-1}, \boldsymbol{P}_{i-1}, \boldsymbol{E}_{i-1}\right]\right) \tag{4}$$

$$f_{\theta'}(x) = \boldsymbol{CLS}_N \tag{5}$$

where $[\boldsymbol{x}_{i-1}, \boldsymbol{P}_{i-1}, \boldsymbol{E}_{i-1}] \in \mathbb{R}^{(1+p+m)\times d}$. θ' denotes the model parameters of ViT, as well as the additional prompt parameters θ^P. The network structure of our proposed model, MVP, is illustrated in Fig. 3.

Throughout the meta-training and meta fine-tuning phases, only the newly added prompt parameters θ^P are updated, while all other parameters of the ViT backbone network remain unchanged. Therefore, a classification task involving the prediction of label y can be represented as follows:

$$\theta^* = \arg\max_{\theta^P} \sum_{\boldsymbol{x}} \log p\left(y \mid f_{\theta'}(\boldsymbol{x}); \theta^P\right) \tag{6}$$

where θ^* represents the optimal value for the prompt parameters θ^P that maximizes the sum of logarithmic probabilities.

C. Meta Fine-Tuning Process

Meta fine-tuning for target few-shot tasks requires effectively using a small amount of labeled support data and achieving meta fine-tuning in a few steps. A common solution is to use data augmentation to expand the support set [9].

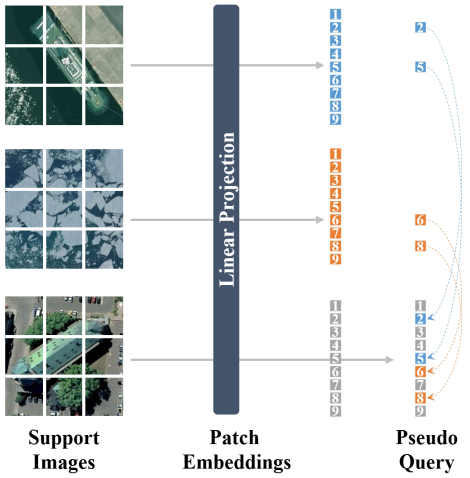


Fig. 4. RPR-aug: our novel data augmentation method.

For a few-shot task $T = \{S, Q\}$, where S is a set of labeled support images, and Q are other unlabeled query images. The method is to create an auxiliary task $T' = \{S, Q'\}$, where the pseudoquery $Q' = \text{augment}(S)$ is composed of augmented support images. This auxiliary task-based meta fine-tuning process enhances the model's adaptability to novel tasks by leveraging augmented data.

In this work, we propose a novel data augmentation method called random patch recombination (RPR-aug), which is designed specifically for ViT-based models. As shown in Fig. 4, the RPR-aug method is applied after the linear projection of the data. Unlike traditional methods [9], [39] that augment the data before feeding it into the backbone networks, our method can fully use the structure of ViT. Specifically, for the patch embeddings $\boldsymbol{E} = \{\boldsymbol{e}^{\text{pos}} \in \mathbb{R}^d \mid \text{pos} \in \mathbb{N}, 1 \leq \text{pos} \leq m\}$ of a given image in a support set S, we select a subset of $\boldsymbol{E}$ and denote their corresponding positions as $\{\textbf{pos} \in \mathbb{R}^{m'}, 1 \leq m' \leq m\}$, with a recombination rate of α. Then, we replace the selected patches with patches from other images in the support set S that have the same positions **pos**. Algorithm 1 summarizes the process in detail.

Algorithm 1 PyTorch Pseudocode for RPR-Aug

```
# Inputs: Patch embeddings of images in a support set.
# Outputs: Recombination of input patch embeddings.

def random_recombine(emb_x, rate):
    # emb_x: [bs, pos, dim]
    # rate: selection rate of \alpha
    pos = emb_x.size\hyperref[eq:patch]{(\ref{eq:patch})}
    rate = int(pos * rate)
    # iterate over all image patch embeddings
    for bs_i, _ in enumerate(emb_x):
        select_pos = random.sample(range(0, pos), rate)
        emb_x = replace(emb_x, bs_i, select_pos)

def replace(emb_x, bs_i, select_pos):
   # iterate over all selected positions
   bs = emb_x.size(0)
   for pos_i in select_pos:
       # note: probability of p(i=j) is approximately 0
       bs_j = random.randint(0, bs-1)
       emb_x[bs_i, pos_i, :] = emb_x[bs_j, pos_i, :]
   return emb_x
```

D. Loss Functions

With the prompt-based ViT backbone networks, we discuss how to define our training objective. We denote the feature representation of support and query image as $f_{\theta'}(x^s)$ and $f_{\theta'}(x^q)$ and write them as $f^s_{\theta'}$ and $f^q_{\theta'}$, for simplicity. Following the prototypical networks [42], the prototype vectors of support data are denoted as $\boldsymbol{\Omega} = \{\boldsymbol{\mu}_c \in \mathbb{R}^d \mid c \in \mathbb{N}, 1 \leq c \leq C\}$. Here, $\boldsymbol{\mu}_c = (1/|S_c|) \sum_{i:y_i^s=c} f^{s_i}_{\theta'}$ is the prototype of class c and $|S_c| = \sum_{i:y_i^s=c} 1$. Then the probability of a query image x^q is defined as a function of its similarity to the prototypes of support data

$$p\left(y^q = c|x^q\right) = \frac{\exp\left(-d\left(f^q_{\theta'}, \boldsymbol{\mu}_c\right)\right)}{\sum_{c'} \exp\left(-d\left(f^q_{\theta'}, \boldsymbol{\mu}_{c'}\right)\right)} \tag{7}$$

where d is the cosine distance. Finally, the training objective $\mathcal{L}$ of our proposed MVP is to minimize the negative log-likelihood and $\mathcal{L} = -\log p(y^q = c|x^q)$, which can be further written as

$$\mathcal{L} = \frac{1}{|S_c|}\left[d\left(f^q_{\theta'}, \boldsymbol{\mu}_c\right) + \log \sum_{c'} \exp -d\left(f^q_{\theta'}, \boldsymbol{\mu}_{c'}\right)\right]. \tag{8}$$

IV. Experiments

In this section, we first introduce the evaluation dataset and the implementation details briefly but comprehensively. Then, we present ablation experiments that demonstrate the significant effectiveness of the MVP design. Finally, we contrast the proposed MVP with the state-of-the-art (SoTA) peer competitors.

A. Experimental Setup

1) Datasets: We evaluate our methods and SoTA algorithms using a challenging FS-RSSC benchmark named AIFS-DATASET [8]. This benchmark is a variant of the META-DATASET [12] that includes a collection of remote sensing datasets. The AIFS-DATASET is composed of two subsets: in-domain and out-of-domain sets. The in-domain set consists of six open-source datasets that do not feature remote sensing scenarios: CUB-200-2011, Describable Textures, Fungi, VGG Flower, Traffic Signs, and CIFAR100. These six datasets are partitioned such that approximately 70%, 15%, and 15% of data are assigned to training, validation, and testing sets, respectively. The out-of-domain set comprises four remote sensing datasets, namely, NWPU-RESISC45, UC-Merced, WHU-RS19, and AID, all of which are exclusively used for testing purposes. Table I displays the specific dataset partitioning.

2) Benchmarking Methods: In this article, we compare our method with several SoTA methods for FS-RSSC, such as Finetune [43], MatchingNet [5], ProtoNet [42], fo-MAML [20], RelationNet [44], fo-Proto-MAML [20], DeepBDC [45], CNAPs [46], SimpleCNAPs [7], and DC-DML [8]. These methods have been evaluated on the AIFS-DATASET [8], the large-scale benchmark for FS-RSSC. In addition, we use PMF [9] as another baseline model, since it has achieved SoTA performance on various few-shot

TABLE I
DATA COMPOSITION AND SPLIT OF AIFS-DATASET [8]

Dataset	Domain	Total	Train	Val	Test
CUB-200-2011	In	200	140	30	30
Describable Textures		47	33	7	7
Fungi		1394	994	200	200
VGG Flower		102	71	15	16
Traffic Signs		43	30	6	7
CIFAR100		100	72	13	15
UCMerced	Out	21	0	0	21
WHU-RS19		19	0	0	19
NWPU-RESISC45		45	0	0	45
AID		30	0	0	30

learning benchmarks. We follow the official code and settings of PMF to reproduce its results on the AIFS-DATASET. For the training settings, all the above methods use the full fine-tuning method. In contrast, our proposed MVP method uses the prompt fine-tuning method.

3) Implementation Details: In terms of data organization, we follow the same setup as the AIFS-DATASET [8]. Specifically, the number of classes included in each task was randomly selected from the range [5, MAXWAY], while the number of support samples per class was randomly selected from the range [1, MAXSHOT]. The sampling algorithm used in this process is based on uniform sampling. In the subsequent text, we use MW and MS to represent MAXWAY and MAXSHOT, respectively. To comprehensively evaluate model performance in our experiments, we set MW to 5/10/20 and MS to 1/5/10/20. As for the pretraining phase, consistent with PMF [9], we also use ViT as the backbone network. ViT is pretrained on the ImageNet1K dataset using the classical self-supervised method DINO [47] method. In our experiments, we demonstrated results based on ViT-tiny and ViT-small. During the meta-training and meta fine-tuning phases, the parameters of ViT are fixed, and only the newly added prompt parameters are updated.

B. Ablation Study

In this section, we conduct an ablation study to analyze the effectiveness of the proposed MVP method. First, we compare the performance of meta-visual prompt-tuning and fully fine-tuning and present the results in Table II. Table III shows the number of learnable parameters that need to be updated using different fine-tuning methods. Second, we evaluate the effectiveness and computational efficiency of the RPR-aug method proposed in this article. Finally, we investigate how the number of prompt tokens affects classification performance and computational efficiency.

1) Is PMF Effective in the Field of Remote Sensing?: To investigate the performance of the full fine-tuning method in the FS-RSSC task, we conducted an ablation study on the AIFS-DATASET using the PMF framework. We compared four settings: 1) M1, which is the pretrained model without any meta-learning process; 2) M2, which is the pretrained model + only the meta-training process; 3) M3, which is the pretrained model + only the meta fine-tuning process; and 4) M4, which is the complete PMF [9] model. From the results in Table II, we can draw a conclusion that the complete PMF (full fine-tuning) does improve the performance over the pretrained model that without any meta-learning process (comparing models M4, M3, M2, with M1). These suggest that full fine-tuning might not be a good solution for FS-RSSC tasks.

TABLE II
EFFECT OF UPDATING BACKBONE OR PROMPT PARAMETERS ACROSS DIFFERENT META-LEARNING PHASES ON THE AVERAGE CLASSIFICATION ACCURACY (%) OF THE AIFS-DATASET USING VIT-SMALL AS THE BACKBONE NETWORK UNDER MW5 MS5 AND MW10 MS10 SCENARIOS."–" INDICATES THAT THIS STAGE WILL NOT BE CARRIED OUT

Training Configuration			Benchmark Results	
Model	Meta Train	Meta Finetune	MW5 MS5	MW10 MS10
M1	–	–	75.5 ± 0.1	76.4 ± 0.3
M2	Backbone	–	75.6 ± 0.2	76.6 ± 0.2
M3	–	Backbone	74.9 ± 0.1	77.2 ± 0.3
M4	Backbone	Backbone	76.6 ± 0.2	78.1 ± 0.1
M5	Prompt	–	79.8 ± 0.3	80.4 ± 0.2
M6	–	Prompt	76.3 ± 0.2	78.6 ± 0.2
M7	Backbone	Prompt	75.6 ± 0.1	78.9 ± 0.2
M8	Prompt	Backbone+Prompt	78.3 ± 0.2	80.1 ± 0.4
M9	Prompt	Prompt	79.6 ± 0.3	81.2 ± 0.2

TABLE III
NUMBER OF TRAINABLE PARAMETERS FOR RN18, VIT, AND PROMPT VIT WITH 200 PROMPT TOKENS

Backbone	Image size	Trainable Params (M)
RN18	224×224	11.28
ViT-tiny	224×224	5.52
ViT-small	224×224	21.66
ViT-base	224×224	85.79
Prompt ViT-tiny	224×224	0.46
Prompt ViT-small	224×224	0.92
Prompt ViT-base	224×224	1.84

2) Effectiveness of MVP for FS-RSSC and Which Stage to Apply MVP?: To evaluate the effectiveness of the meta visual prompt-tuning (MVP) method for FS-RSSC, we performed ablation studies under three settings: 1) M5, which only applied MVP for meta-training; 2) M6, which only applied MVP for meta fine-tuning; and 3) M9, which applied MVP for both meta-training and meta fine-tuning. The results in Table II revealed that: 1) the model with MVP that completes either meta-training or meta fine-tuning process has a better performance than the PMF model with the same process (M5 versus M2, and M6 versus M3); 2) moreover, the model with MVP that completes either the meta-training or meta fine-tuning process even outperforms the PMF model that completes both the processes (M5 versus M4, and M6 versus M4); and 3) the complete MVP model significantly surpasses the complete PMF model (M9 versus M4). These findings indicate that MVP is an effective and superior method for FS-RSSC, as it can learn from a few examples more efficiently and accurately than PMF.

3) Combining MVP and Full Fine-Tuning: To validate whether MVP and full fine-tuning can work together to achieve better results, we carried out experiments under three settings:

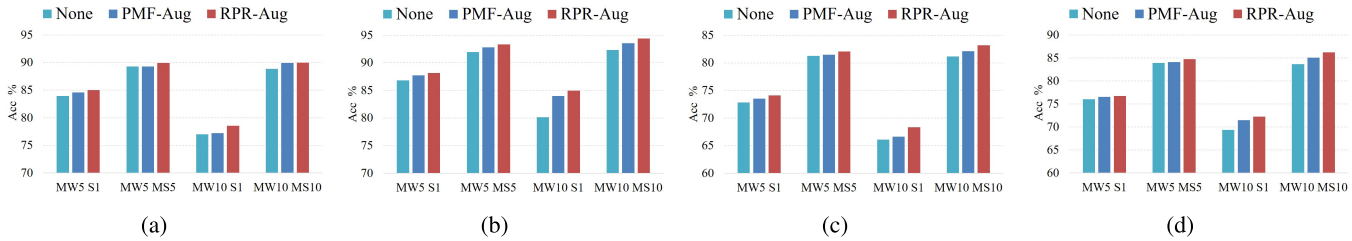


Fig. 5. Comparison of different data augmentation methods on various-way-various-shot learning for four out-of-domain datasets. (a) Results on the UCM dataset. (b) Results on the WHU dataset. (c) Results on the NWPU dataset. (d) Results on the AID dataset. None denotes no data augmentation, RPR-aug denotes random patch recombination, and PMF-aug denotes the PMF data augmentation method.

1) M7 applies full fine-tuning in the meta-training stage and MVP tuning in the meta fine-tuning stage; 2) M8 applies MVP tuning in the meta-training stage and full fine-tuning for all the parameters in the meta fine-tuning stage; and 3) M9 applied MVP tuning for both meta-training and meta fine-tuning. The results in Table II revealed that: 1) M7 does not show a significant improvement over M4, indicating that performing MVP only in the meta fine-tuning stage is effective but not remarkable; 2) M8 achieves a substantial improvement over M4, indicating that MVP can help the model obtain a better initialization point which leads to better generalization to new tasks; and 3) M9 achieves the SoTA result, demonstrating that using MVP in both the meta-training and meta fine-tuning stages can attain the maximum benefit improvement.

4) Effectiveness of RPR-Aug: This study proposes a novel data augmentation method called RPR-aug, which is detailed in Section III-C. This section mainly verifies the performance of the RPR-aug method compared with other data augmentation methods. We use the validated PMF [9] data augmentation (PMF-aug) as the main comparison method. PMF-aug includes popular techniques such as mixup, cutmix, color-jitter, translation, and cutout, and it activates one or more of these methods based on probability. Similar to PMF, we apply the proposed RPR method to construct pseudoqueries based on support images, used as auxiliary tasks in the meta fine-tuning phase. However, unlike PMF, these auxiliary tasks are only used to update prompt parameters while freezing the backbone network. Furthermore, we also automatically selected the learning rate lr and the recombination rate α for each task. We used MVP to choose the optimal lr and α from the ranges $lr \in [1e-4, 1e-3, 1e-2, 0.1]$ and $\alpha \in [0.05, 0.1, 0.2, 0.25]$, and then performed meta fine-tuning with them.

To evaluate the efficacy of RPR-aug, we conducted a series of experiments on the four remote sensing datasets of AIFS (UCM, WHU, NWPU, and AID) for both the various-way-various-shot and various-way-one-shot scenarios and compared the results. The outcomes of these experiments are presented in Fig. 5. Our findings revealed that overall, RPR-aug outperformed PMF-aug significantly. Specifically, when considering the average results across all the four tasks (MW5 S1, MW5 MS5, MW10 S1, and MW10 MS10), RPR-aug yielded improvements of 0.62%, 0.7%, 0.98%, and 0.68% compared with PMF-aug on UCM, WHU, NWPU, and AID, respectively. Notably, in one-shot learning tasks, RPR-aug achieved particularly significant enhancements compared with o PMF-aug, with the MW10 S1 task indicating improvements of 1.36%, 0.97%, 1.71%, and 0.74% on UCM, WHU, NWPU, and AID, respectively. In summary, the proposed RPR-aug technique outperforms the PMF-aug significantly in few-shot image classification tasks, especially in one-shot learning scenarios.

TABLE IV
COMPARISON OF RUNNING TIME FOR DATA AUGMENTATION METHODS ON FS-RSSC TASKS WITH VIT BACKBONE NETWORKS

Backbone	Method	Avg Time (s)
ViT-tiny	PMF-Aug	3.16
	RPR-Aug	**0.67**
ViT-small	PMF-Aug	3.08
	RPR-Aug	**0.97**
ViT-base	PMF-Aug	3.03
	RPR-Aug	**1.61**

TABLE V
COMPARISON OF PARAMETERS, AVERAGE ACCURACY, AND RUNNING TIME FOR DIFFERENT NUMBERS OF TOKENS

Token	Params	Avg Acc (%)	Time/Iteration (s)
10	1,150	81.9 ± 0.1	2.70
20	1,210	80.3 ± 0.3	5.87
50	1,385	81.8 ± 0.2	6.20
200	2,542	76.5 ± 0.4	7.53

5) Efficiency of RPR-Aug: To compare the efficiency of our RPR-aug method and the PMF-aug method, we further conducted experiments on the AIFS-DATASET using different backbone networks of ViT-tiny and ViT-small. We tested six ten-way k-shot tasks, where $k \in [1, 2, 4, 6, 8, 10]$, and repeated 1000 data augmentation experiments for each task. To ensure the fairness of the comparison, we set the recombination rate α of RPR-aug to a maximum value of 0.25 and used the same model and same batch. Table IV shows the total average running time of 1000 experiments for each task. From Table IV, we can see that RPR-aug is three times faster than PMF-aug on average, indicating that our RPR-aug method is more efficient and more suitable for FS-RSSC tasks based on ViT.

6) Prompt Tokens Number: This is an important hyperparameter needed to tune for MVP, and we carried experiment to test the effect of the number of prompt tokens on the perfor-

TABLE VI
IN-DOMAIN AND OUT-OF-DOMAIN ACCURACY OF DIFFERENT MODELS WITH MAXWAY = 5 AND MAXSHOT = 5

Model	Backbone	Tuning	In-Domain Accuracy (%)						Out-of-Domain Accuracy (%)				Avg
			CUB	Textures	Fungi	Flower	Signs	CIFAR	UCM	WHU	NWPU	AID	
Finetune [43]	RN18	Full	57.0±1.7	38.3±1.2	45.8±1.8	73.9±1.2	50.1±1.3	54.5±1.4	63.6±1.7	76.1±1.5	55.5±1.6	60.2±1.7	57.5
MatchingNet [5]	RN18	Full	49.7±0.5	36.7±0.4	38.2±0.5	66.1±0.4	52.6±0.4	46.9±0.4	54.2±0.5	65.0±0.5	46.9±0.5	51.3±0.5	50.8
ProtoNet [42]	RN18	Full	44.9±0.5	33.5±0.3	35.0±0.4	56.2±0.5	35.7±0.4	42.8±0.5	51.1±0.5	54.2±0.5	42.2±0.4	44.6±0.4	44.0
fo-MAML [20]	RN18	Full	59.5±0.6	38.5±0.4	44.0±0.5	70.3±0.5	49.0±0.4	49.7±0.5	52.1±0.5	60.8±0.5	44.5±0.5	50.0±0.5	51.8
RelationNet [44]	RN18	Full	60.5±1.9	38.8±1.3	46.8±1.9	72.8±1.4	**79.9±1.2**	50.9±1.6	56.6±1.8	65.4±1.6	49.3±1.5	52.8±1.6	57.4
Proto-MAML [12]	RN18	Full	47.0±1.5	33.3±1.1	35.5±1.4	60.1±1.4	36.6±1.2	41.9±1.3	50.9±1.4	52.6±1.4	41.6±1.4	44.2±1.5	44.4
DeepBDC [45]	RN18	Full	59.4±0.3	39.8±0.4	48.6±0.3	73.6±0.2	47.2±0.5	41.8±0.5	59.9±0.6	68.3±0.3	51.8±0.4	57.7±0.3	54.8
CNAPs [46]	RN18	Full	65.6±0.6	41.5±0.5	46.5±0.5	69.7±0.9	43.2±0.7	55.7±0.5	66.9±0.6	61.8±0.5	49.1±0.5	54.6±0.5	55.5
SimpleCNAPs [7]	RN18	Full	64.0±0.5	44.9±0.9	49.8±0.5	73.1±0.4	48.5±0.4	64.8±0.5	74.6±0.5	74.5±0.5	57.0±0.5	65.2±0.5	61.6
DC-DML [8]	RN18	Full	62.2±0.5	49.5±0.5	50.6±0.6	82.2±0.4	48.7±0.4	59.7±0.5	78.9±0.5	80.9±0.5	60.6±0.5	68.6±0.6	64.2
PMF-tiny † [9]	ViT-t	Full	83.2±0.3	61.5±0.2	55.6±0.5	83.8±0.2	43.5±0.2	54.6±0.4	83.5±0.2	88.2±0.1	75.3±0.3	77.6±0.2	70.7
PMF-small † [9]	ViT-s	Full	84.3±0.2	72.3±0.1	61.8±0.5	86.3±0.1	53.7±0.5	61.3±0.3	89.1±0.1	93.0±0.4	80.8±0.2	83.8±0.1	76.6
MVP-tiny (ours)	ViT-t	Prompt	**87.9±0.3**	66.1±0.2	65.9±0.4	89.1±0.4	55.9±0.3	63.5±0.2	84.1±0.3	89.6±0.4	77.2±0.2	79.6±0.2	75.9
MVP-small (ours)	ViT-s	Prompt	87.2±0.2	**73.4±0.1**	**66.9±0.4**	**91.7±0.3**	60.3±0.4	**67.3±0.2**	**89.3±0.3**	**93.4±0.1**	**81.5±0.2**	**84.7±0.2**	**79.6**

TABLE VII
IN-DOMAIN AND OUT-OF-DOMAIN ACCURACY OF DIFFERENT MODELS WITH MAXWAY = 10 AND MAXSHOT = 10

Model	Backbone	Tuning	In-Domain Accuracy (%)						Out-of-Domain Accuracy (%)				Avg
			CUB	Textures	Fungi	Flower	Signs	CIFAR	UCM	WHU	NWPU	AID	
Finetune [43]	RN18	Full	37.4±1.1	31.9±1.2	33.7±1.3	56.1±1.3	46.8±1.1	37.6±1.2	48.5±1.5	61.1±1.3	41.8±1.3	46.8±1.4	44.2
MatchingNet [5]	RN18	Full	42.3±1.5	33.5±1.1	31.5±1.3	58.1±1.4	52.9±1.3	39.7±1.3	48.9±1.5	58.3±1.3	40.0±1.4	46.0±1.3	45.1
ProtoNet [42]	RN18	Full	35.4±1.2	29.1±1.1	24.2±1.1	42.1±1.3	28.4±1.0	34.8±1.2	42.3±1.3	45.9±1.3	33.9±1.3	35.1±1.2	35.1
fo-MAML [20]	RN18	Full	51.3±1.5	35.6±1.3	37.5±1.4	61.5±1.3	43.2±1.2	40.3±1.3	50.1±1.4	58.2±1.4	39.2±1.4	44.8±1.3	46.2
RelationNet [44]	RN18	Full	44.9±1.5	33.1±1.2	35.7±1.7	58.3±1.5	**73.4±1.3**	36.8±1.4	49.1±1.5	59.8±1.3	40.2±1.4	46.0±1.5	47.7
Proto-MAML [12]	RN18	Full	37.5±1.4	28.3±1.0	27.0±1.0	48.3±1.4	28.4±1.0	33.0±1.2	39.3±1.4	41.6±1.5	32.2±1.1	33.6±1.3	34.9
DeepBDC [45]	RN18	Full	59.3±0.2	43.3±0.1	46.7±0.4	71.6±0.2	52.0±0.2	39.1±0.1	64.4±0.4	69.2±0.3	52.1±0.4	57.7±0.4	55.5
CNAPs [46]	RN18	Full	59.5±0.5	44.4±0.4	46.6±0.5	70.1±0.4	55.8±0.5	52.8±0.5	61.7±0.5	60.6±0.5	44.3±0.5	52.0±0.5	54.8
SimpleCNAPs [7]	RN18	Full	64.6±0.5	40.4±0.4	47.8±0.6	68.6±0.4	54.3±0.5	51.6±0.5	62.1±0.5	61.5±0.5	43.6±0.5	52.7±0.5	54.7
DC-DML [8]	RN18	Full	56.3±0.5	45.7±0.5	43.1±0.6	71.9±0.4	47.5±0.4	53.7±0.5	69.4±0.6	72.2±0.5	49,7±0.6	56.8±0.6	57.4
PMF-tiny † [9]	ViT-t	Full	85.9±0.1	67.6±0.4	54.4±0.4	89.1±0.1	51.5±0.4	60.2±0.6	86.2±0.2	90.4±0.5	78.2±0.3	80.5±0.3	74.4
PMF-small † [9]	ViT-s	Full	86.1±0.1	75.5±0.1	60.4±0.4	87.2±0.1	58.3±0.3	62.7±0.4	89.8±0.3	93.8±0.2	81.8±0.1	84.8±0.1	78.1
MVP-tiny (ours)	ViT-t	Prompt	**89.1±0.3**	68.9±0.1	66.0±0.3	89.0±0.2	56.9±0.3	64.4±0.4	86.8±0.1	90.0±0.4	78.6±0.3	82.1±0.2	77.3
MVP-small (ours)	ViT-s	Prompt	88.8±0.2	**76.2±0.2**	**67.4±0.1**	**92.4±0.3**	64.9±0.4	**69.3±0.3**	**89.9±0.2**	**94.4±0.3**	**83.2±0.4**	**85.1±0.1**	**81.2**

TABLE VIII
IN-DOMAIN AND OUT-OF-DOMAIN ACCURACY OF DIFFERENT MODELS WITH MAXWAY = 20 AND MAXSHOT = 20

Model	Backbone	Tuning	In-Domain Accuracy (%)						Out-of-Domain Accuracy (%)				Avg
			CUB	Textures	Fungi	Flower	Signs	CIFAR	UCM	WHU	NWPU	AID	
Finetune [43]	RN18	Full	33.3±1.0	40.4±0.7	27.5±0.7	54.6±1.0	47.8±1.0	37.0±1.0	46.9±1.0	56.1±1.0	37.1±1.0	41.4±1.0	42.2
MatchingNet [5]	RN18	Full	38.5±0.6	41.8±0.4	27.4±0.4	63.9±0.5	58.8±0.5	40.5±0.5	44.7±0.6	54.6±0.5	35.9±0.5	40.6±0.5	44.7
ProtoNet [42]	RN18	Full	30.8±0.6	40.0±0.4	20.1±0.4	45.2±0.6	31.5±0.4	32.8±0.6	38.2±0.6	41.7±0.7	29.4±0.6	30.5±0.6	34.1
fo-MAML [20]	RN18	Full	45.2±0.6	42.9±0.4	31.4±0.5	63.7±0.5	48.1±0.5	39.5±0.6	42.4±0.6	48.8±0.6	32.4±0.5	36.3±0.5	43.1
RelationNet [44]	RN18	Full	35.3±0.6	38.0±0.4	27.6±0.5	65.5±0.5	85.7±0.3	34.3±0.5	40.9±0.6	50.7±0.6	31.9±0.5	37.5±0.6	44.7
Proto-MAML [12]	RN18	Full	35.4±1.1	39.4±0.7	22.7±0.7	49.8±1.1	33.8±1.0	34.0±1.0	36.4±1.1	40.5±1.1	28.4±1.0	30.7±1.0	35.1
DeepBDC [45]	RN18	Full	55.7±0.4	48.9±0.2	42.6±0.5	71.9±0.1	59.6±0.2	40.3±0.1	59.5±0.4	67.8±0.2	49.0±0.3	55.8±0.4	55.1
CNAPs [46]	RN18	Full	54.2±0.6	48.9±0.4	41.5±0.6	77.0±0.4	62.7±0.5	51.2±0.6	57.8±0.5	52.5±0.6	38.8±0.6	44.2±0.6	52.9
SimpleCNAPs [7]	RN18	Full	54.9±0.6	47.8±0.4	41.7±0.6	74.2±0.4	71.1±0.4	48.0±0.6	55.6±0.6	50.3±0.7	40.0±0.6	45.5±0.6	52.9
DC-DML [8]	RN18	Full	48.6±0.7	54.5±0.4	36.0±0.6	76.9±0.4	51.3±0.4	53.0±0.6	62.8±0.6	63.0±0.6	43.3±0.7	48.6±0.7	53.8
PMF-tiny † [9]	ViT-t	Full	84.4±0.2	69.1±0.3	49.5±0.6	84.5±0.6	40.8±0.4	52.7±0.5	81.3±0.2	86.2±0.1	70.3±0.4	74.9±0.4	69.4
PMF-small † [9]	ViT-s	Full	84.1±0.3	80.3±0.4	55.7±0.2	85.6±0.2	74.7±0.3	60.4±1.5	88.2±0.3	93.5±0.1	77.8±0.2	83.0±1.1	78.3
MVP-tiny (ours)	ViT-t	Prompt	87.2±0.3	72.2±0.3	61.3±0.4	89.9±0.4	64.4±0.5	63.4±0.5	85.6±0.2	90.1±0.1	76.9±0.3	79.9±0.3	77.1
MVP-small (ours)	ViT-s	Prompt	**88.9±0.2**	**80.9±1.4**	**67.1±0.3**	**94.3±0.6**	**78.7±0.5**	**67.1±0.4**	**89.7±0.1**	**93.7±0.2**	**81.8±0.4**	**84.3±0.2**	**82.1**

mance and efficiency of the model. To conduct comparative experiments, we designed our experimental setup following the principle of controlling variables. We used the NWPU dataset of the AIFS-DATASET as a benchmark and evaluated the performance of the MVP model on four RS-FSSC tasks, including five-way one-shot, five-way five-shot, ten-way one-shot, and ten-way ten-shot. Table V shows a comparison of the performance of the MVP model with different numbers of prompt tokens loaded in these four tasks. As can be seen, when the number of tokens is set to 10, the MVP model achieves the highest classification accuracy in most tasks. Moreover, we observed that as the number of tokens increases, so does the time consumption of the model. In summary, considering the tradeoff between performance accuracy and computational efficiency, we fixed the number of prompt tokens for the MVP model at ten in this article and applied this setting consistently across all the tasks.

C. Comparison With SoTA Methods

1) Main Results and Comparisons: We present the experimental results obtained under three distinct evaluation benchmarks, where MW and MS were set to 5, 10, and 20, respectively, in Tables VI–VIII. "†" denotes the results

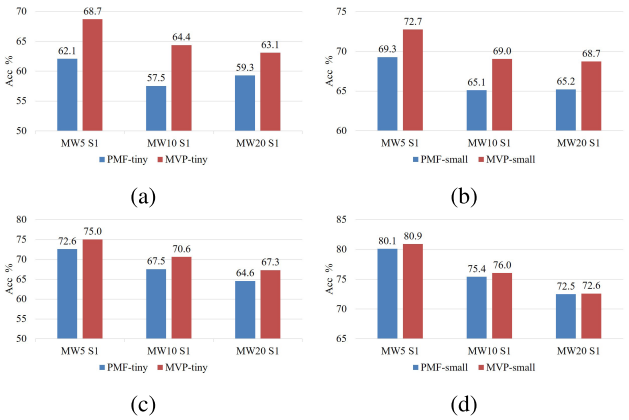


Fig. 6. Comparative performance evaluation of PMF [9] and our MVP model in various-way-one-shot learning scenario. (a) AIFS-DATASET: results of ViT-tiny; (b) AIFS-DATASET: results of ViT-small; (c) out-of-domain AIFS-DATASET: results of ViT-tiny; and (d) out-of-domain AIFS-DATASET: results of ViT-small.

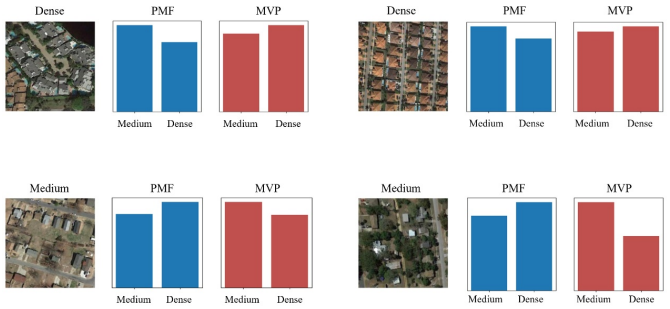


Fig. 7. Qualitative results of few-shot classification for PMF and our MVP.

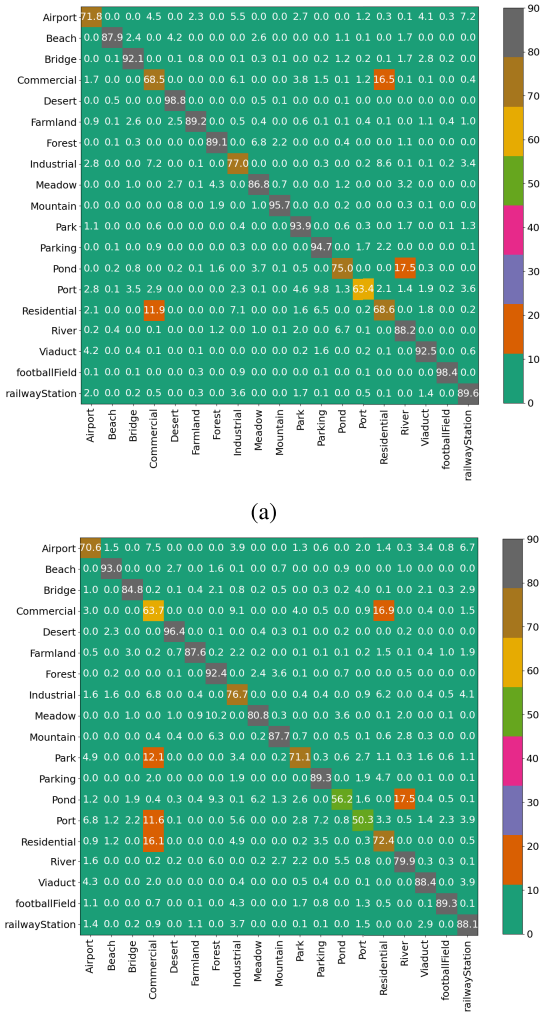


Fig. 8. Confusion matrix results of MVP and PMF [9] methods for 19 remote sensing scene classification tasks in the WHU dataset. (a) Results of MVP. (b) Results of PMF.

that we reproduced. **BOLD** denotes the best results, and underline denotes the second-best results. Comparing these results, we can deduce that MVP outperforms the SoTA methods in terms of average accuracy. Specifically, our MVP method achieves significant improvements over the strong baseline PMF method, with average increases of 3.0%, 3.1%, and 3.8% at MW and MS of 5/10/20. These findings support our view that prompt-tuning techniques are better suited for few-shot classification tasks than full-tuning techniques.

In the context of in-domain tasks, MVP exhibits considerable advantages over other SoTA methods, particularly in fine-grained evaluation benchmarks such as the CUB, Fungi, and Flower datasets. In these datasets, MVP surpasses all other listed methods in terms of performance. This observation further highlights the superior capabilities of the MVP technology in fine-grained few-shot classification tasks. Notably, when compared with the PMF method, the MVP approach demonstrates the most prominent advantages in the Fungi dataset at MW and MS of 5/10/20, with improvements of 5.1%, 7%, and 11.4%, respectively.

Regarding out-of-domain tasks, our analysis indicates that MVP generally outperforms the second-best PMF method. This trend is particularly notable in the NWPU dataset, where MVP improves by 0.7%, 1.4%, and 4% at MW and MS of 5/10/20. Moreover, these results suggest that MVP has better generalization ability when dealing with new unseen FS-RSSC tasks.

2) Analysis of Model Parameters and Performance: Using Tables III and VI–VIII, we can perform a comprehensive analysis of how model parameters impact the performance and accuracy of our model. Our study demonstrates that with the use of a meta-based prompt-tuning framework, ViT-tiny achieves a significant improvement in performance compared with RN18 despite having similar parameter counts. Specifically, in the out-of-domain FS-RSSC task, our MVP-tiny method shows an average accuracy increase of 19.2% over the DC-DML [8] method based on RN18 across all three evaluation benchmarks. Furthermore, while ViT-small possesses four times as many parameters as ViT-tiny, it does not exhibit a significant increase in accuracy for the FS-RSSC task. Nonetheless, the MVP-small method only shows an average precision increase of 3.9% compared with MVP-tiny across all three evaluation benchmarks. Therefore, our results suggest that deploying applications based on MVP-tiny can provide a more balanced tradeoff between efficiency and performance.

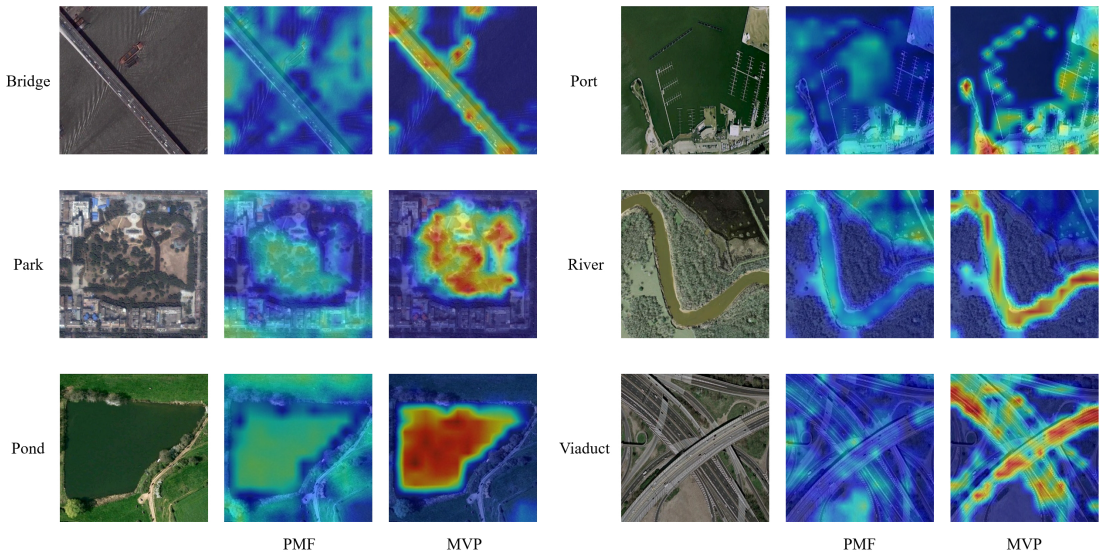


Fig. 9. Feature map visualization results of the MVP and PMF [9] methods for six remote sensing scene classification tasks.

3) Impact of Domain Shift on Classification Accuracy: The cross-domain results shown in Tables VII and VIII indicated that all the algorithms suffered from a varying degree of accuracy degradation when dealing with the MW20 MS20 task compared with the MW10 MS10 task. This suggested that domain shift, especially when the number of categories increased, had a significant effect on classification accuracy. On the other hand, the results also revealed that the MVP model was more robust to domain shift. In particular, the MVP achieved high accuracy on the out-of-domain datasets while effectively reducing the effect of domain shift. On the UCM, WHU, NWPU, and AID datasets, MVP-small and MVP-tiny showed an average accuracy drop of 1.34% and 1.5%, respectively, while PMF-small and PMF-tiny showed an average drop of 1.95% and 5.65%, and DC-DML showed an average drop of 7.6%. Hence, the MVP model outperformed other models in cross-domain FS-RSSC tasks, especially when the number of categories was large.

4) Results for One-Shot Learning: Fig. 6 presents the results of our model on the one-shot learning task with one support image per category. Our MVP method outperforms the strong baseline PMF method in all one-shot learning tasks. Specifically, when ViT-tiny is used as the backbone network, MVP-tiny outperforms PMF-tiny in all the ten evaluation tasks on the AIFS-DATASET, as shown in Fig. 6(a) and (b). On average, MVP-tiny improves by 6.64%, 6.84%, and 3.80% compared with PMF-tiny when MW is 5/10/20, respectively. Similarly, MVP-small improves by 3.46%, 3.94%, and 3.49% compared with PMF-small when MW is 5/10/20, respectively. When considering all the out-of-domain tasks in Fig. 6(c) and (d), MVP-tiny improves by 2.44%, 3.07%, and 2.73% compared with PMF-tiny in these three tasks, respectively. Similarly, MVP-small improves by 0.82%, 0.62%, and 0.12% compared with PMF-small in these three tasks, respectively. These results demonstrate that our MVP method has better generalization ability in the challenging one-shot learning task.

5) Qualitative Analysis of Few-Shot Classification: The aim of this section is to examine the disparities between the outputs generated by the MVP and PMF models. Based on the in-domain results presented in Tables VI–VIII, it was observed that the MVP model demonstrated a notable advantage in processing fine-grained classification tasks. For example, when averaging the three classification tasks of CUB, Fungi, and Flower, our MVP-small model outperforms the PMF-small model by 4.5%, 4.9%, and 8.3% on average under the settings of MW5 MS5, MW10 MS10, and MW20 MS20, respectively. We therefore conjectured that the fine-grained classification abilities of the MVP model may account for its superior performance in downstream tasks. To verify this hypothesis, we designed an experiment where we obtained a ten-way five-shot task from the NWPU dataset, which was distinct from the in-domain AIFS-DATASET that we used for training the models. We subsequently contrasted the outputs generated by each model for every query image. We conducted a rigorous examination on the categories that the MVP model predicted correctly but the PMF model failed to do so. Interestingly, we observed that the MVP model achieved a significantly higher average accuracy than the PMF model when comparing two similar categories: “dense residential” and “medium residential.” To investigate this phenomenon further, we devised a two-way one-shot experiment. Fig. 7 clearly shows that the MVP model is more capable of handling fine-grained classification problems. To sum up, these observations suggest that the MVP model can learn more discriminative features, which enhance the fine-grained classification abilities of the model in downstream tasks.

6) Visualization Analysis Results: To demonstrate the superiority of our method MVP over the strong baseline method PMF in the FS-RSSC task, we performed a comparative analysis of the two methods on the classification performance of 19 remote sensing scenes based on the WHU dataset. We first used confusion matrices to show the average classification accuracy (%) of each category for the MVP and PMF

methods over 1000 experiments per class. Fig. 8(a) and (b) shows the results of the MVP and PMF methods, respectively, where each element of the matrix represents the percentage of predictions for the actual class (row) and the predicted class (column), and the diagonal elements represent the correct classification accuracy of the model for each category. From Fig. 8(a) and (b), we can observe that the MVP method outperforms the PMF method in most categories, especially in Bridge, Commercial, Meadow, Mountain, Park, Parking, Pond, Port, River, Viaduct, and footballField, where the MVP method achieves 7.3%, 4.8%, 6%, 8%, 22.8%, 5.4%, 18.8%, 13.1%, 4.1%, 8.3%, and 9.1% higher accuracy than the PMF method, respectively. The average accuracy of the MVP method over all 19 scenes is 5.4% higher than that of the PMF method.

To further investigate the difference between the MVP and PMF methods in feature extraction, we applied GradCAM [48] model to visualize the feature maps of the two models for the scene categories with higher accuracy by the MVP method, and the results are shown in Fig. 9. We present the feature map visualization results for six representative scenes. We selected the feature maps from the norm1 layer of the last block of the ViT backbone as the input to the GradCAM model. The first column in Fig. 9 shows the original scene images, the second column shows the feature maps of the PMF model, and the third column shows the feature maps of the MVP model. We can see that our MVP model's feature maps are more focused and precise, highlighting the discriminative regions of the scenes; while the PMF model's feature maps are more diffuse and noisy, with background clutter interfering with foreground feature extraction. By combining Figs. 8 and 9, we find MVP has a more robust and effective classification performance than the PMF method and is more suitable for FS-RSSC tasks.

V. Conclusion

In this article, we focused on the challenging and practical scenario of FS-RSSC, where the goal is to classify remote sensing images into different categories using only a few labeled examples for each category. To tackle this problem, we proposed a novel and efficient method called MVP that leverages prompt tuning and meta-learning to adapt a pretrained visual transformer model to new tasks with minimal data and resources. Specifically, MVP has three main components: 1) prompt tuning, a parameter-efficient fine-tuning technique that updates only the newly added prompt parameters and freezes the rest of the model, which reduces the storage demand and mitigates the overfitting risk; 2) meta-learning, a fast adaptation technique that learns to initialize the prompt parameters from multiple source tasks and rapidly adapts them to new tasks with only a few gradient steps, which facilitates cross-domain adaptation; and 3) data augmentation, a novel technique that operates on the patch embeddings of the input tokens of transformer blocks to enhance the scene representation and diversity. We evaluated MVP on a realistic cross-domain FS-RSSC benchmark dataset and demonstrated its superior performance over the existing methods. MVP achieved especially remarkable results on the vary-way, vary-shot, and one-shot tasks, which are more challenging and relevant for real-world applications. Our work paved the way for applying ViT models to FS-RSSC tasks and provided a solution that is suitable for the deployment platform of the FS-RSSC algorithms. In the future, we expect to extend the training data domain to include more data distributions, which may further improve the accuracy and robustness of the prompt-tuning based methods for FS-RSSC tasks, especially for enhancing cross-domain performance.

References

[1] Z. Cui, W. Yang, L. Chen, and H. Li, "MKN: Metakernel networks for few shot remote sensing scene classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 60, 2022, Art. no. 4705611.

[2] X. X. Zhu et al., "Deep learning in remote sensing: A comprehensive review and list of resources," *IEEE Geosci. Remote Sens. Mag.*, vol. 5, no. 4, pp. 8–36, Dec. 2017.

[3] R. P. de Lima and K. Marfurt, "Convolutional neural network for remote-sensing scene classification: Transfer learning analysis," *Remote Sens.*, vol. 12, no. 1, p. 86, 2019.

[4] Y. Li, Z. Shao, X. Huang, B. Cai, and S. Peng, "Meta-FSEO: A meta-learning fast adaptation with self-supervised embedding optimization for few-shot remote sensing scene classification," *Remote Sens.*, vol. 13, no. 14, p. 2776, 2021.

[5] O. Vinyals et al., "Matching networks for one shot learning," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 29, 2016, pp. 3637–3645.

[6] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 770–778.

[7] P. Bateni, R. Goyal, V. Masrani, F. Wood, and L. Sigal, "Improved few-shot visual classification," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 14493–14502.

[8] L. Li, X. Yao, G. Cheng, and J. Han, "AIFS-DATASET for few-shot aerial image scene classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 60, 2022, Art. no. 5618211.

[9] S. X. Hu, D. Li, J. Stühmer, M. Kim, and T. M. Hospedales, "Pushing the limits of simple pipelines for few-shot learning: External data and fine-tuning make a difference," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2022, pp. 9068–9077.

[10] L. Bertinetto, J. F. Henriques, P. H. S. Torr, and A. Vedaldi, "Meta-learning with differentiable closed-form solvers," 2018, *arXiv:1805.08136*.

[11] Y. Guo et al., "A broader study of cross-domain few-shot learning," in *Proc. 16th Eur. Conf. Comput. Vis.* Glasgow, U.K.: Springer, 2020, pp. 124–141.

[12] E. Triantafillou et al., "Meta-dataset: A dataset of datasets for learning to learn from few examples," 2019, *arXiv:1903.03096*.

[13] T. Brown et al., "Language models are few-shot learners," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 33, 2020, pp. 1877–1901.

[14] M. Jia et al., "Visual prompt tuning," in *Proc. Eur. Conf. Comput. Vis.* Springer, 2022, pp. 709–727.

[15] M. Gong, J. Li, Y. Zhang, Y. Wu, and M. Zhang, "Two-path aggregation attention network with quad-patch data augmentation for few-shot scene classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 60, 2022, Art. no. 4511616.

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

[17] H. Li et al., "RS-MetaNet: Deep meta metric learning for few-shot remote sensing scene classification," 2020, *arXiv:2009.13364*.

[18] L. Li, J. Han, X. Yao, G. Cheng, and L. Guo, "DLA-MatchNet for few-shot remote sensing image scene classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 59, no. 9, pp. 7844–7853, Sep. 2021.

[19] Q. Zeng and J. Geng, "Task-specific contrastive learning for few-shot remote sensing image scene classification," *ISPRS J. Photogramm. Remote Sens.*, vol. 191, pp. 143–154, Sep. 2022.

[20] C. Finn, P. Abbeel, and S. Levine, "Model-agnostic meta-learning for fast adaptation of deep networks," in *Proc. Int. Conf. Mach. Learn.*, 2017, pp. 1126–1135.

[21] P. Zhang, Y. Bai, D. Wang, B. Bai, and Y. Li, "A meta-learning framework for few-shot classification of remote sensing scene," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, Jun. 2021, pp. 4590–4594.

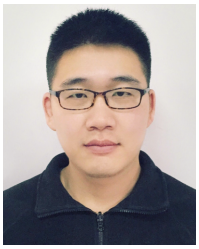
[22] C. Ma, X. Mu, P. Zhao, and X. Yan, "Meta-learning based on parameter transfer for few-shot classification of remote sensing scenes," *Remote Sens. Lett.*, vol. 12, no. 6, pp. 531–541, Jun. 2021.
[23] A. Dosovitskiy et al., "An image is worth 16×16 words: Transformers for image recognition at scale," 2020, *arXiv:2010.11929*.
[24] S. K. Roy, A. Deria, D. Hong, B. Rasti, A. Plaza, and J. Chanussot, "Multimodal fusion transformer for remote sensing image classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 61, pp. 1–20, 2023.
[25] M. Bi, M. Wang, Z. Li, and D. Hong, "Vision transformer with contrastive learning for remote sensing image scene classification," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 16, pp. 738–749, 2023.
[26] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of deep bidirectional transformers for language understanding," 2018, *arXiv:1810.04805*.
[27] R. L. Logan IV, I. Balažević, E. Wallace, F. Petroni, S. Singh, and S. Riedel, "Cutting down on prompts and parameters: Simple few-shot learning with language models," 2021, *arXiv:2106.13353*.
[28] E. Ben Zaken, S. Ravfogel, and Y. Goldberg, "BitFit: Simple parameter-efficient fine-tuning for transformer-based masked language-models," 2021, *arXiv:2106.10199*.
[29] S.-A. Rebuffi, A. Vedaldi, and H. Bilen, "Efficient parametrization of multi-domain deep neural networks," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 8119–8127.
[30] J. O. Zhang, A. Sax, A. Zamir, L. Guibas, and J. Malik, "Side-tuning: A baseline for network adaptation via additive side networks," in *Proc. 16th Eur. Conf. Comput. Vis.* Glasgow, U.K.: Springer, 2020, pp. 698–714.
[31] Z. Chen et al., "Vision transformer adapter for dense predictions," 2022, *arXiv:2205.08534*.
[32] E. J. Hu et al., "LoRA: Low-rank adaptation of large language models," 2021, *arXiv:2106.09685*.
[33] C. Shorten and T. M. Khoshgoftaar, "A survey on image data augmentation for deep learning," *J. Big Data*, vol. 6, no. 1, pp. 1–48, 2019.
[34] S. Gidaris, P. Singh, and N. Komodakis, "Unsupervised representation learning by predicting image rotations," 2018, *arXiv:1803.07728*.
[35] C. Zhang, Y. Cai, G. Lin, and C. Shen, "DeepEMD: Few-shot image classification with differentiable earth mover's distance and structured classifiers," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2020, pp. 12203–12213.
[36] S. Yun, D. Han, S. Chun, S. J. Oh, Y. Yoo, and J. Choe, "CutMix: Regularization strategy to train strong classifiers with localizable features," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 6023–6032.
[37] J.-H. Kim, W. Choo, and H. O. Song, "Puzzle mix: Exploiting saliency and local statistics for optimal mixup," in *Proc. Int. Conf. Mach. Learn.*, 2020, pp. 5275–5285.
[38] K. Li, Y. Zhang, K. Li, and Y. Fu, "Adversarial feature hallucination networks for few-shot learning," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 13470–13479.
[39] R. Ni, M. Goldblum, A. Sharaf, K. Kong, and T. Goldstein, "Data augmentation for meta-learning," in *Proc. 38th Int. Conf. Mach. Learn.*, 2021, pp. 8152–8161.
[40] S. Yang, L. Liu, and M. Xu, "Free lunch for few-shot learning: Distribution calibration," 2021, *arXiv:2101.06395*.
[41] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "ImageNet: A large-scale hierarchical image database," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2009, pp. 248–255.
[42] J. Snell, K. Swersky, and R. Zemel, "Prototypical networks for few-shot learning," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 30, 2017, pp. 4080–4090.
[43] J. Yosinski, J. Clune, Y. Bengio, and H. Lipson, "How transferable are features in deep neural networks?" in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 27, 2014, pp. 3320–3328.
[44] F. Sung, Y. Yang, L. Zhang, T. Xiang, P. H. S. Torr, and T. M. Hospedales, "Learning to compare: Relation network for few-shot learning," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 1199–1208.
[45] J. Xie, F. Long, J. Lv, Q. Wang, and P. Li, "Joint distribution matters: Deep Brownian distance covariance for few-shot classification," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2022, pp. 7972–7981.
[46] J. Requeima, J. Gordon, J. Bronskill, S. Nowozin, and R. E. Turner, "Fast and flexible multi-task classification using conditional neural adaptive processes," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 32, pp. 7959–7970.
[47] M. Caron et al., "Emerging properties in self-supervised vision transformers," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2021, pp. 9650–9660.
[48] R. R. Selvaraju, M. Cogswell, A. Das, R. Vedantam, D. Parikh, and D. Batra, "Grad-CAM: Visual explanations from deep networks via gradient-based localization," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2017, pp. 618–626.

Junjie Zhu received the M.S. degree in computer science and technology from the Academy of Military Sciences, Beijing, China, in 2021. He is currently pursuing the Ph.D. degree with the Research Centre for Artificial Intelligence, National Innovation Institute of Defense Technology (NIIDT), Beijing.

His research interests include computer vision, few-shot learning, and meta-learning.

Yiying Li received the M.S. and Ph.D. degrees from the National University of Defense Technology, Changsha, China, in 2017 and 2021, respectively.

She is currently an Assistant Professor with the Artificial Intelligence Research Center, National Innovation Institute of Defense Technology, Beijing, China. She has authored or coauthored more than 20 research articles on top-tier conferences and journals, including International Conference on Machine Learning (ICML), Neural Information Processing Systems Conference (NeurIPS), Association for the Advancement of Artificial Intelligence (AAAI), and International Conference on Computer Vision (ICCV). Her research interests focus on deep learning algorithms and applications and meta-learning.



Ke Yang received the M.S. and Ph.D. degrees from the National University of Defense Technology, Beijing, China, in 2016 and 2020, respectively.

He is currently an Assistant Professor with the Artificial Intelligence Research Center, National Innovation Institute of Defense Technology. He has authored or coauthored more than 20 research articles on top-tier conferences, including the International Conference on Computer Vision (ICCV), Association for the Advancement of Artificial Intelligence (AAAI), Association for Computing Machinery (ACM), and IEEE Conference on Computer Vision and Pattern Recognition (CVPR). His research interests include computer vision and deep learning.

Naiyang Guan received the M.S. and Ph.D. degrees from the National University of Defense Technology, Changsha, China, in 2006 and 2011, respectively.

He is currently an Associate Professor with the Artificial Intelligence Research Center, National Innovation Institute of Defense Technology. He has authored or coauthored more than 60 research articles on top-tier journals, including the IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE (T-PAMI), IEEE TRANSACTIONS ON NEURAL NETWORKS AND LEARNING SYSTEMS (T-NNLS), IEEE TRANSACTIONS ON IMAGE PROCESSING (T-IP), and IEEE TRANSACTIONS ON SIGNAL PROCESSING (T-SP), and top-tier conferences, including the IEEE International Conference on Data Mining (ICDM), International Joint Conference on Artificial Intelligence (IJCAI), European Conference on Computer Vision (ECCV), and International Joint Conference on Neural Networks (IJCNN). His research interests include machine learning, computer vision, and data mining.

Zunlin Fan received the Ph.D. degree in electrical engineering from Air Force Engineering University, Xi'an, China, in 2018.

He is currently an Assistant Professor with the Artificial Intelligence Research Center, National Innovation Institute of Defense Technology, Beijing, China. His research interests include computer vision, statistical image processing, image denoising, image enhancement, and pattern recognition.

Chunping Qiu received the B.Sc. and M.Sc. degrees in photogrammetry and remote sensing from the Zhengzhou Institute of Surveying and Mapping, Zhengzhou, China, in 2013 and 2016, respectively, and the Dr.-Ing. degree in signal processing in Earth observation from Technische Universität München (TUM), Munich, Germany, in 2020.

In 2019, she was a Guest Researcher with the Telecommunications and Remote Sensing Laboratory, University of Pavia, Pavia, Italy. Since 2021, she has been a Researcher with the PLA Strategic Support Force Information Engineering University, Zhengzhou. She is currently a Post-Doctoral Researcher with the Artificial Intelligence Research Center, National Innovation Institute of Defense Technology, Beijing, China. Her main research interests include deep learning and self-supervised learning for remote sensing applications, and big Earth data management.

Xiaodong Yi received the Ph.D. degree from the National University of Defense Technology, Changsha, China, in 2006.

He is currently a Full Professor with the National Innovation Institute of Defense Technology. He has devoted to Kylin operating systems, which was used in Milkway supercomputers, for over ten years. His research interests include operating systems, high-performance computing, robotics software, and artificial intelligent.