Few-Shot Transfer Learning for SAR Image Classification Without Extra SAR Samples

Yuan Tai, Yihua Tan, *Member, IEEE*, Shengzhou Xiong, Zhaojin Sun, and Jinwen Tian

***Abstract*—Deep learning-based synthetic aperture radar (SAR) image classification is an open problem when training samples are scarce. Transfer learning-based few-shot methods are effective to deal with this problem by transferring knowledge from the electro–optical (EO) to the SAR domain. The performance of such methods relies on extra SAR samples, such as unlabeled novel class's samples or labeled similar classes samples. However, it is unrealistic to collect sufficient extra SAR samples in some application scenarios, namely the extreme few-shot case. In this case, the performance of such methods degrades seriously. Therefore, few-shot methods that reduce the dependence on extra SAR samples are critical. Motivated by this, a novel few-shot transfer learning method for SAR image classification in the extreme few-shot case is proposed. We propose the connection-free attention module to selectively transfer features shared between EO and SAR samples from a source network to a target network to supplement the loss of information brought by extra SAR samples. Based on the Bayesian convolutional neural network, we propose a training strategy for the extreme few-shot case, which focuses on updating important parameters, namely the accurately updating important parameters. The experimental results on the three real-SAR datasets demonstrate the superiority of our method.**

***Index Terms*—Bayesian convolutional neural network (Bayesian-CNN), few-shot transfer learning, synthetic aperture radar (SAR).**

I. Introduction

SYNTHETIC aperture radar (SAR) imaging benefits from propagating radar signals during occluded weather or at night. Radar signals sent from mobile antennas and reflection signals have been collected for subsequent signal processing to produce high-resolution images regardless of weather conditions and shielding. Therefore, SAR imaging is a powerful technique in many applications, such as continuous environmental monitoring, large-scale surveillance [1], Earth remote sensing [2], and military investigation. Image classification is one of the basic tasks in these applications.

Recently, with a large number of labeled training samples, DL is a popular and effective solution to SAR image classification. However, we can only obtain few labeled SAR samples of the interesting targets in some application scenarios because of collection difficulty. In such scenarios, the problem has to be studied by few-shot learning that trains the network with few labeled SAR samples (less than 20) of each class. Existing few-shot learning methods for SAR are mainly divided into two types.

1) Transfer learning-based (TL-based) methods that match features between the electro–optical (EO) and SAR domains with extra similar SAR samples. For example, Rostami *et al.* [3] proposed to minimize the distance of two feature distributions of unlabeled SAR and EO samples, and then fine-tuned the network with few labeled SAR samples. The performance of this method relies on the extra unlabeled SAR samples, which are required to be similar classes as the testing samples.
2) Meta-learning-based methods that learn from similar labeled SAR samples without using EO samples. For example, Wang *et al.* [4] pretrained the network with hundreds of labeled samples of seven supporting categories in the MSTAR dataset before the network fine-tunes procedure with few labeled samples in the three target categories. However, lots of the extra labeled SAR samples are required in such methods.

In general, extra SAR samples, including unlabeled novel classes samples or labeled SAR samples of similar categories (in [4], the novel classes and supporting classes were different subclasses of the tank) are necessary to achieve good performances for the existing TL-based few-shot methods and meta-learning-based few-shot methods. However, in some extreme application scenarios, such as surveillance, it is unrealistic to collect extra similar SAR samples, which will result in a severe decline in the performance of existing few-shot learning methods. Therefore, a few-shot learning method for SAR image classification that can mitigate the difficulty of the scarcity of extraSAR samples is critical, which we name as the extreme few-shot learning method. In this article, we propose a TL-based extreme few-shot learning method that can reduce the dependence on the extra similar SAR samples.

In fact, two core reasons make extra SAR samples critical to the performance of existing few-shot TL methods. First, the big shift between the EO and SAR samples results in parts of features extracted by the network pretrained with EO samples being unsuitable for SAR image classification. For example, Fig. 1 shows the comparison between samples of aircraft and vehicles in EO and SAR domains. We can see that their shapes are similar,

Manuscript received December 5, 2021; revised February 9, 2022; accepted February 25, 2022. Date of publication March 1, 2022; date of current version March 21, 2022. This work was supported in part by the National Natural Science Foundation of China under Grant 41371339 and in part by The Fundamental Research Funds for the Central Universities under Grant 2017KFYXJJ179. *(Corresponding author: Yihua Tan.)*

The authors are with the School of Artificial Intelligence and Automation, State Key Laboratory of Multispectral Information Processing Technology, Huazhong University of Science and Technology, Wuhan 430074, China (e-mail: t_y_@hust.edu.cn; yhtan@hust.edu.cn; xiongsz@hust.edu.cn; zhaojin_sun@hust.edu.cn; jwtian@mail.hust.edu.cn).

Digital Object Identifier 10.1109/JSTARS.2022.3155406

This work is licensed under a Creative Commons Attribution 4.0 License. For more information, see https://creativecommons.org/licenses/by/4.0/

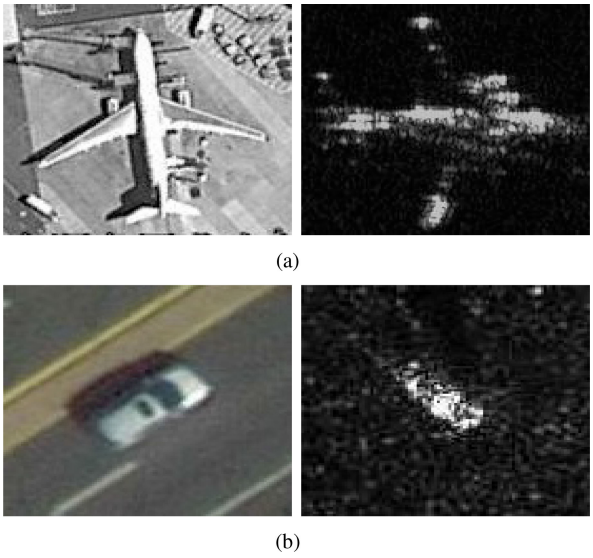


Fig. 1. Comparison of samples between the EO domain and the SAR domain. (a) Comparision of the aircrafts in the EO and SAR domain. (b) Comparision of the vehicles in the EO and SAR domain.

but their textures are quite different. This phenomenon indicates that SAR and EO samples of the same category share "common features," such as shape, and present distinct "individual features," such as texture. Therefore, common features of SAR samples extracted by a network pretrained with EO samples are also very effective for SAR image classification. On the other hand, the extraction of individual features of SAR samples is transferred from that of EO samples, which is harmful to SAR image classification. Existing methods have to learn from extra SAR samples of similar classes to supplement those individual features of targets in the SAR domain. Second, extra SAR samples can first update the network parameters to a suitable initial point so that the network is easier to be well-trained with few labeled SAR samples. Based on the aforementioned analyses, we must take measures to compensate for the loss of two advantages brought by extra SAR samples since we have only few labeled samples in the extreme few-shot case.

For the first loss of supplementation of individual features, if we can strengthen common features and suppress the transferring of individual features of EO samples, it is probably to make the network able to classify SAR samples more accurate based on the enhanced common features. This aim can be realized by transferring common features from a source network to a target network, and connecting features between them is a popular and effective way [5]–[7]. In this article, we transfer common features from a complicated source network to a simplified target network also by connecting features between both networks. Differently, we propose a novel transferring structure for the extreme few-shot case. Specifically, we construct a complex source network to capture rich features and a small target network to be more easily trained in the extreme few-shot case. The transferring structure connecting the source and target network is in charge of enhancing common features transferring and depressing individual features transferring. However, to achieve the goal, following two factors need to be considered.

1) Which features of the source network are common features and individual features.
2) Different layers of the target network are adjusted to receive the two kinds of feature with suitable weights.

It can be implemented by connecting all the feature channels in the source network to each layer of the target network, in which each connection is corresponding to a weight combo. It includes two types of weights that indicate the transferring extent in the layer level and feature channel level. However, it is not easy to determine both of them artificially, so that the transferring structure needs to be learnable. An attention mechanism is a popular way to learn the weights of the features we care about by designing various attention modules [8]–[10]. However, most attention mechanism-based methods construct the attention module with fully-connected (FC) layers, which bring a large number of parameters due to the high dimensions of features, increasing the difficulty of attention module training, making such attention modules difficult to be well-trained in the extreme few-shot case. Therefore, in this article, we propose a novel attention module, which avoids the large number of parameters brought by the FC layer by using the learnable vector to replace the FC layer, namely connection-free attention module.

For the second problem of loss of good initial point, we have to design an appropriate parameter update strategy with few labeled samples. Considering that the features extracted by different parameters of the network are of different importance to classification ability, it is reasonable to infer that training those important parameters more accurately can give the network better generalization ability. Concentrating on updating important parameters, we probably mitigate the optimization problem to avoid training all the parameters equally. Based on this point, we need to find a way to measure the importance of the parameters of the network first. Bayesian convolutional neural network (Bayesian-CNN) [11] models the uncertainty on the parameters, which provides a basis for measuring the importance of the parameters. Thus, we can regard the higher uncertainty parameters as less important because the important parameters should be stable for a well-trained network. Therefore, the Bayesian-CNN is introduced as the target network to measure the importance of each parameter according to its uncertainty. Furthermore, to train these important parameters more accurately, based on the Bayesian-CNN, we propose a training strategy for the extreme few-shot case, namely accurately updating important parameters (AUIPs). Specifically, first, we pretrain the target network by EO samples with the initial learning rate, and then we train the target network by few labeled SAR samples with adaptive learning rates for each parameter according to their uncertainty. Generally, the structure of our method is shown in Fig. 2, which consist of following three parts.

1) A complex CNN is introduced as the source network to capture rich features of EO samples.
2) Several connection-free attention modules are constructed to selectively transfer common features from the source network to the target network.
3) A small Bayesian-CNN is introduced as the target network to capture effective features for SAR image classification.

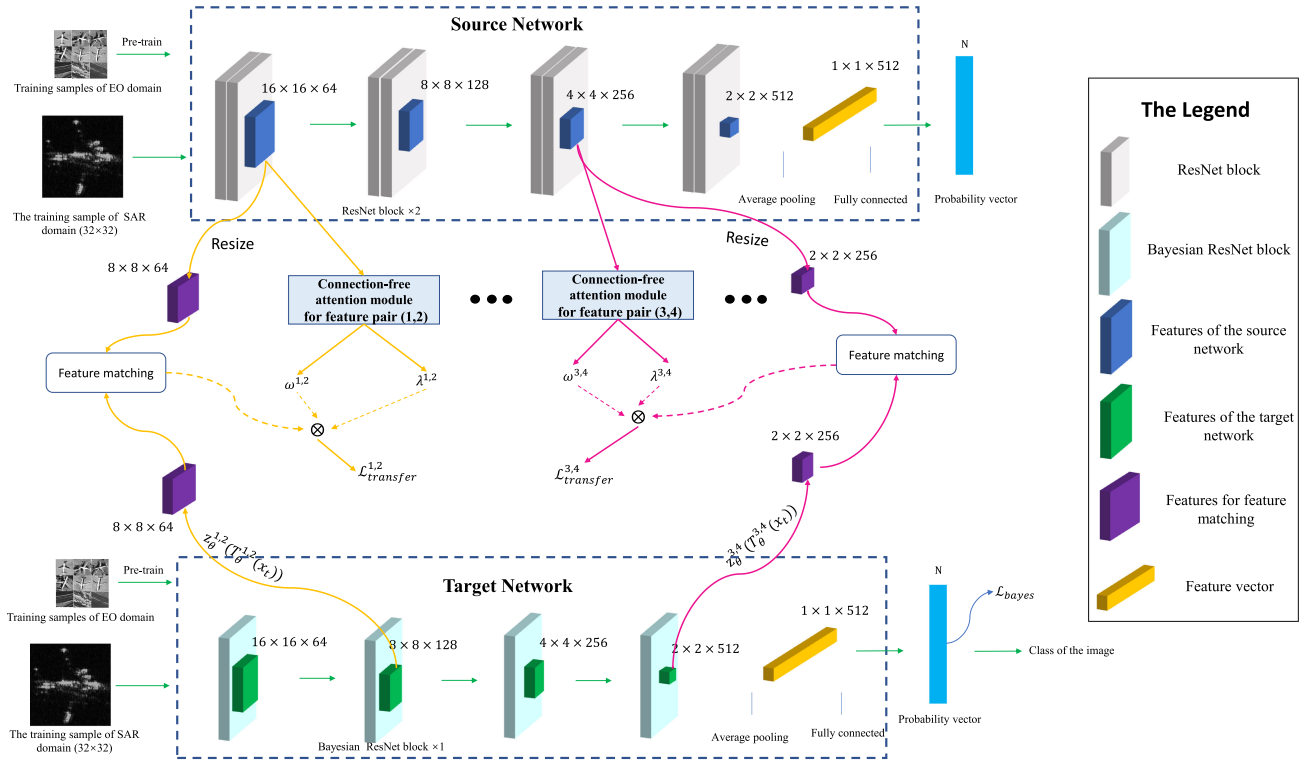


Fig. 2. Training process. Each connection-free attention module generates a $w^{m,n}$ and a $\lambda^{m,n}$ to weight feature channels and feature connection when transferring features from the mth layer of the source network to the nth layer of the target network. The target network is responsible for classifying SAR samples. The Bayesian ResNet block, represents the parameters of the ResNet block, is defined by a Gaussian distribution. Note that during the testing process, only the target network works.

The training details are introduced in Section III-F and Algorithm 1.

In summary, the contributions of this article are as follows.

1) A novel few-shot transfer learning method for SAR image classification is proposed in the extreme few-shot case. It focuses on the application scenarios in which no extra similar SAR samples are available.
2) The connection-free attention module is proposed to selectively transfer common features from a complex source network to a small target network in the extreme few-shot case.
3) The Bayesian-CNN is introduced as the target network to measure the importance of its parameters. Based on this, the training strategy of AUIPs is proposed to solve the problem that the number of training samples is insufficient to update all parameters to a suitable value in the extreme few-shot case.

II. Realated Work

A. Few-Shot Learning in SAR Image Classification

Few-shot classification methods learn a classifier with only few labeled training samples of each class [12]. The TL strategy is widely used in the SAR domain, which transfers knowledge from extra similar SAR samples. Huang *et al.* [13] used an unsupervised learning method to generate many unlabeled SAR samples to train the depth autoencoder. Zhang *et al.* [14] transferred knowledge from another SAR task, where labeled data were easy to obtain. Shang *et al.* [15] modified a CNN with an information recorder used to store spatial features of labeled samples to label the unlabeled samples according to the spatial similarity. However, in the extreme few-shot case, the extra SAR samples are unavailable, causing a decline in the performance of these methods.

On the other hand, the meta-learning strategy in image classification [16] has become popular in few-shot learning, which predicts novel classes based on few labeled samples and the meta-dataset containing thousands of base classes with many labeled samples. In the SAR domain, for the lack of base classes, Wang *et al.* [4] repeatedly chose seven classes of the MSTAR [17], a thousand times to construct the meta-dataset, and then fine-tuned the network with few labeled samples of the three target classes. The base classes are very similar to the target classes in this method to ensure performance. However, in the extreme few-shot case, the base classes are unavailable.

B. Feature Transferring

Transferring features from a source network to a target network became popular in feature transferring, and recently, attention-based methods have been proven effective. Jang *et al.* [7] constructed an attention-based meta-network consisting of two kinds of attention modules to selectively transfer features

from a source network to a target network. Ji *et al.* [18] also proposed an attention-based meta-network with a different structure from that of Jang *et al.* [7]. This meta-network learns relative similarities between features and applies identified similarities to control distillation intensities of all possible pairs. Zagoruyko *et al.* [6] computed statistics of features across the channel dimension to construct a spatial attention module to features from a source network to a target network. These methods utilized attention modules to selectively transfer effective features from a source network to a target network, enhancing the feature extraction ability of the target network. However, these attention-based methods are unsuitable for the extreme few-shot case because of the large number of parameters brought by the attention modules in them. Thus, in this article, a light-weight connection-free attention module is proposed to transfer common features in the extreme few-shot case.

C. Bayesian Convolutional Neural Network

The Bayesian approach has been studied in the field of learning neural networks for a few decades [19]. Several methods have been proposed for Bayesian-CNN, such as the Laplace approximation [20], variational inference [21], [22], and probabilistic back-propagation [23]. Recently, several problems of different fields have been studied with Bayesian-CNN. Kendall and Gal [24] modeled the uncertainty with the Bayesian-CNN to study the confidence of the output of the network. Kendall *et al.* [25] decided weights among tasks in the multitask learning problem according to the uncertainty for each task. Ebrahimi *et al.* [26] applied the Bayesian-CNN into continual learning and achieve state-of-the-art performance in several open datasets. However, the potential of Bayesian-CNN in the field of few-shot transfer learning has not been exploited.

III. Problem Formulation and Proposed Method

A. Problem Formulation

This article aims to learn a network without extra SAR samples but only few labeled SAR samples, namely extreme few-shot learning. Specifically, let $D_T = (X_T, Y_T)$ be the few labeled SAR samples of target classes and D'_T be the extra SAR samples of similar classes, including labeled or unlabeled samples. X and Y are the images and the corresponding class labels, respectively. Compared with the common few-shot TL methods that train a network by both $D_T = (X_T, Y_T)$ and D'_T, the extreme few-shot learning method train a network only by the $D_T = (X_T, Y_T)$. In general, TL-based few-shot methods belong to semisupervised algorithms because they use unlabeled SAR samples to support training, and meta-learning-based few-shot methods follow the episode training strategy, which needs extra labeled SAR samples of similar classes as the target class. Differently, the extreme few-shot learning methods are supervised algorithms and do not follow the episode training strategy.

B. Overview of the Proposed Method

In this section, we first overview the structure of the source network and the target network, and then, we briefly introduce the training process.

As shown in Fig. 2, the source network contains eight ResNet blocks, namely 8-Resblock-based CNN. The target network is a Bayesian-CNN, which only contains four Bayesian ResNet blocks, namely 4-Resblock-based Bayesian-CNN. ResNet block is the basic unit of ResNet [27], which contains three convolution layers and one down-sampling layer. The Bayesian ResNet block represents each parameter of the ResNet block as a Gaussian distribution. The details of the connection-free attention module and Bayesian-CNN will be introduced in Sections III-C and III-D, respectively.

The training process generally divides into two parts.

1) The source network and the target network are pretrained independently from scratch with EO samples $D_S = (X_S, Y_S)$.
2) The target network and connection-free attention modules are trained with few labeled SAR samples $D_T = (X_T, Y_T)$.

As shown in Fig. 2, the cores of the training process are as follows.

1) $\lambda^{m,n}$ and $w^{m,n}$ are learned by a connection-free attention module to weight the feature connection (S^m, T^n) and the feature channels in the feature pair $(S^m(x_t), T^n(x_t))$, respectively, where S^m and T^n represent the mth layer of the source network and the nth layer of the target network, respectively, $S^m(x_t)$ and $T^n(x_t)$ are the features of the mth layer of the source network and the nth layer of the target network with the inputs x_t, respectively.
2) The target network are trained with the AUIP training strategy. The details of the training scheme are shown in Algorithm 1.

C. Learn to Transfer Common Features With Connection-Free Attention Module

We aim to enhance the common features and suppress the individual features of EO samples of the target network T_θ parameterized by θ, through transferring common features from the source network S. To achieve this aim, we connect all the feature channels in the source network to each layer of the target network with different weights. Besides, the connection-free attention module is proposed to learn the weights for both feature channels in the feature pair $(S^m(x_t), T^n_\theta(x_t))$ and the feature connection (S^m, T^n) when transferring $S^m(x_t)$ to $T^n_\theta(x_t)$.

Specifically, following two operations are needed for transferring $S^m(x_t)$ to $T^n_\theta(x_t)$.

1) The number of feature channels of the feature pair $(S^m(x_t), T^n(x_t))$ are equaled by a 1×1 convolution function $z^{m,n}_\theta$.
2) The F-norm of the difference of $S^m(x_t)$ and $z^{m,n}_\theta(T^n_\theta(x_t))$ is minimized. This operation is also called "feature matching" [5].

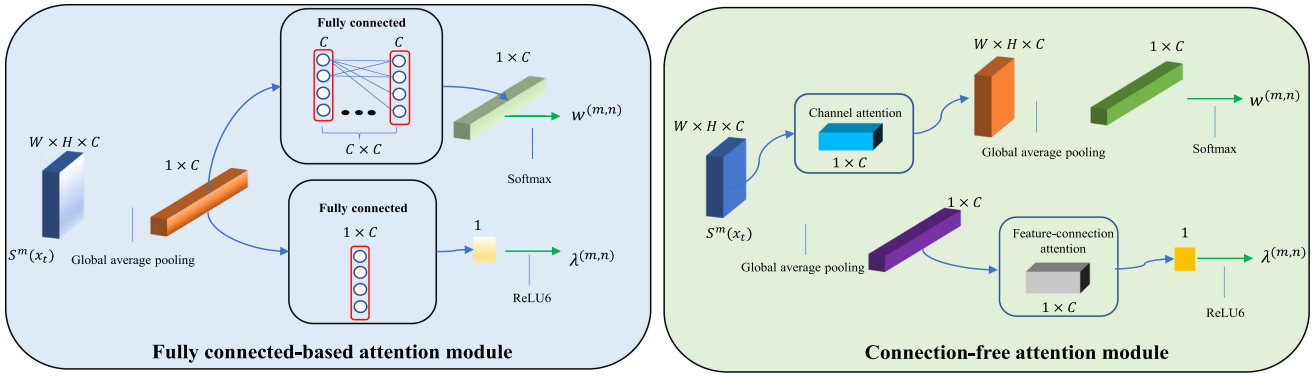


Fig. 3. Comparison between the FC-based attention module and our connection-free attention module. The FC-based attention module contains two FC layers with $(C+1)\times C$ parameters, and the attention module contains channel attention and feature-connection attention with $2\times C$ parameters.

Based on the two operations, the following equation is given:

$$\mathcal{L}_{\text{match}}^{m,n}(\theta \mid x_t) = \sum_c \|(S^m(x_t)_c - (z_\theta^{m,n}(T_\theta^n(x_t)))_c)\|_F^2 \quad (1)$$

where m and n represent the mth layer of the source network and the nth layer of the target network, respectively, c is the cth feature channel, and the size of $S^m(x_t)$ is resized to be the same as $T_\theta^n(x_t)$. Note that θ contains parameters of both T_θ and z_θ, but z_θ is not included in the target network (see Fig. 2).

To achieve the aim of weighting feature channels and the feature connection when transferring $S^m(x_t)$ to $T_\theta^n(x_t)$, Jang *et al.* [7] constructed a meta-network implemented by FC layers to obtain $w^{m,n}$ and $\lambda^{m,n}$ for weighting feature channels and the feature connection, which is an attention module indeed. Thus, the meta-network is renamed as the FC-based attention module for easier understanding in this article. In fact, the number of the parameters of the FC-based attention module reaches 10^6 because of the FC layer, making it difficult to be well-trained in the extreme few-shot case. Therefore, to avoid the large number of parameters brought by the FC layers, we propose the connection-free attention module consisting of channel attention and feature-connection attention. The number of parameters of our attention module only reaches 10^3. The comparison between the meta-network and our attention module is shown in Fig. 3.

Specifically, the $w^{m,n}$ for weighting feature channels in feature pair $(S^m(x_t), T_\theta^n(x_t))$ is computed by the channel attention by

$$w^{m,n} = \text{softmax}(\text{AvgPooling}(\mathbf{f}(S^m(x_t), \text{CA}))) \quad (2)$$

where CA represents the channel attention, which is a $1\times C$ vector, where C is the number of feature channels of $S^m(x_t)$, $\mathbf{f}(a,b)$ represents the each element in the cth channel of a perform pixel-wise operation with the each element in the cth channel of b, AvgPooling is the operation of global average pooling, softmax is the softmax function to make $\sum_c w_c^{m,n} = 1$, and $w^{m,n}$ is a $1\times C$ vector.

Besides, the $\lambda^{m,n}$ for weighting feature connection, (S^m, T^n) is computed by the feature-connection attention by

$$\lambda^{m,n} = \text{ReLU6}(\mathbf{g}(\text{AvgPooling}(S^m(x_t), \text{FCA})) \quad (3)$$

where FCA represents the feature-connection attention, which is a $C\times 1$ vector, $\mathbf{g}(a,b)$ represents the a and b perform matrix multiplication, ReLU6 [28] is a function to prevent $\lambda^{m,n}$ from being too large, and $\lambda^{m,n}$ is a value. Note that each parameter of both channel attention and feature-connection attention is learnable, and we use ϕ to represent the parameters of the attention module. The comparison results between the meta-network and our attention are given in Table IX, and the detailed difference between our method and the method in [7] is discussed in Section IV-F.

Based on the outputs of our attention module, the objective function for transferring features from the source network to the target network is defined as follows:

$$\mathcal{L}_{\text{transfer}}(\theta, \phi \mid x_t) = \sum\nolimits_{m,n\in\mathcal{Q}} \lambda^{m,n} \sum\nolimits_c \mathbf{f}(\mathcal{F}_c^{m,n}, w_c^{m,n}) \quad (4)$$

where $\mathcal{Q}$ is the set of candidate feature connections, $\mathbf{f}$ is the same as that in (2), and $\mathcal{F}^{m,n} = \|(S^m(x_t) - z_\theta^{m,n}(T_\theta^n(x_t)))\|_F^2$ is a $1\times C$ vector.

D. Learn to Accurately Update Important Parameters

To supplement the loss of good initial points brought by extra SAR samples, we design a training strategy that gives the network better generalization ability by training important parameters more accurately, namely AUIPs. Therefore, we need to measure the importance of the parameters of the network first. To achieve this goal, the Bayesian-CNN [11] is introduced as the target network because it models the uncertainty on the parameters, which can provide a basis for measuring the importance. First, we briefly introduce the Bayesian-CNN.

Bayesian-CNN models the parameter uncertainty by initializing each parameter with a Gaussian distribution. Specifically, the ith parameter $\theta_i \in \theta$ is defined as a Gaussian distribution with the mean μ_i and the standard deviation σ_i, represented as $\theta_i \sim N(\mu_i, \sigma_i)$. When the Bayesian-CNN propagates forward, θ_i is sampled as

$$\xi_i = \mu_i + \log(1+\exp(\sigma_i)) \circ \epsilon, \epsilon \sim N(0,1) \quad (5)$$

where ξ_i participates in the calculation of network output as the value of sampling θ_i, $\circ$ means the point multiplication, and ϵ is a random value sampled from the Gaussian distribution $N(0,1)$.

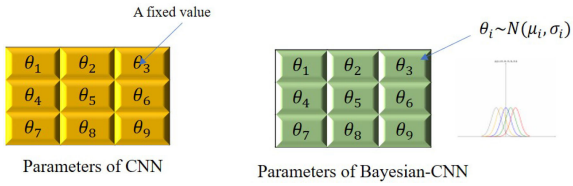


Fig. 4. Comparison of parameters of CNN (left-hand side) and Bayesian-CNN (right-hand side).

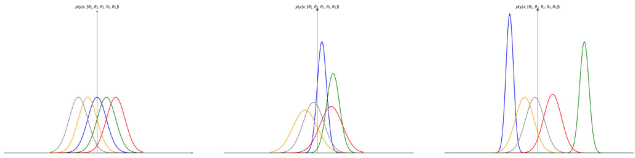


Fig. 5. Diagram of the evolution of parameters distributions through TL from EO domain to SAR domain. (a) Parameters initialized by Gaussian distributions. (b) Posterior distribution after pretraining with training samples of the EO domain. (c) Posterior distribution after training with training samples of the SAR domain.

To visualize the difference between CNN and Bayesian-CNN, Fig. 4 shows a comparative example of the 3×3 convolution kernel in the CNN and the Bayesian-CNN. As shown in Fig. 4, each parameter in the CNN is a fixed value, but in the Bayesian-CNN is a Gaussian distribution.

In fact, the output of the Bayesian-CNN during the training process is obtained by averaging the network output of multiple sampling the parameters

$$\text{output} = \frac{1}{N}\sum_{j=1}^{N} B(x_t, \xi^j) \tag{6}$$

where B represents the Bayesian-CNN, output is the output of the Bayesian-CNN, representing the score for each class of a training sample x_t, N is the sampling times, and ξ^j is the jth sampling result of network parameters. Intuitively, the important features of the Bayesian-CNN should be stable because the output of a well-trained Bayesian-CNN to a certain sample should be relatively fixed in different sampling times. Therefore, the parameters extracting these important features should be less volatile in each sampling time. Based on the abovementioned analysis, it is reasonable to infer that the parameters with low standard deviation significantly impact the network output, showing the importance of such parameters to the output of the network. Besides, Ebrahim *et al.* [26] verified this inference in the continuous learning task [29]. They focused on those unimportant parameters of the network to solve the problem of "forgetting." Unlike [26], this article focuses on those important parameters by increasing their learning rates and reducing that of unimportant parameters. Specifically, the learning rate β_i of $\theta_i \in \theta$ is scaled according to standard deviation σ_i by

$$\beta_i \leftarrow \beta * \gamma_i \tag{7}$$

where $\gamma_i = \frac{1}{\log(1+e^{\sigma_i})}$ is to ensure the positive value and β is the initial learning rate. Fig. 5 illustrates how important and unimportant parameters change while transferring from the EO domain to the SAR domain [see Fig. 5(b) and (c)]. The value of μ_i and σ_i of important parameters (the blue one and the green one, respectively) makes a more significant change. Note that Fig. 5 is not the actual result but just a diagram.

E. Objective Function

Up to now, we can learn the target network in the extreme few-shot case by enhancing the common features of the target network and updating important parameters of the target network more accurately. However, learning the Bayesian-CNN needs to estimate its parameters' posterior distribution, but the general loss function for image classification, such as the cross-entropy loss L_{CE}, cannot learn the posterior distribution. Instead, a popular method for training the Bayesian-CNN is to learn an approximating distribution $q(\theta|\iota)$ parameterized by ι to minimize Kullback–Leibler (KL) divergence with the true Bayesian posterior on the parameters

$$\theta^* = \arg\min_{\theta} \text{KL}(q(\boldsymbol{\theta}|\iota) \| P(\boldsymbol{\theta}|x_t, y_t)). \tag{8}$$

This objective function can be deduced as

$$\begin{aligned}&\mathcal{L}_{\text{bayes}}(\theta|x_t, y_t, \iota)\\ &\quad = \text{KL}[q(\theta|\iota)\|P(\theta)] - E_{q(\theta|\iota)}[\log(P(y_t|x_t, \theta))]\end{aligned} \tag{9}$$

where $p(\theta)$ represents the prior distribution we set and $q(\theta|\iota)$ is the variational posterior distribution.

Further, (9) can be approximated using N Monte Carlo samples from the variational posterior [30]

$$\begin{aligned}&\mathcal{L}_{\text{bayes}}(\theta|x_t, y_t, \iota)\\ &\approx \sum\nolimits_{i=1}^{N} \log q\left(\boldsymbol{\theta}^i|\iota\right) - \log P\left(\boldsymbol{\theta}^i\right) - \log\left(P\left(y_t|x_t, \theta^i\right)\right)\end{aligned} \tag{10}$$

where N is the sampling times.

Finally, based on the (10), the total objective function is given by

$$\mathcal{L}_{\text{total}}\left(\theta, \phi|x_t, y_t, \iota\right) = \mathcal{L}_{\text{transfer}}\left(\theta, \phi \mid x_t\right) + \mathcal{L}_{\text{bayes}}(\theta|x_t, y_t, \iota) \tag{11}$$

where $\mathcal{L}_{\text{transfer}}\left(\theta, \phi \mid x_t\right)$ is the objective function for transferring features and $\mathcal{L}_{\text{bayes}}(\theta|x_t, y_t, \iota)$ is the objective function for the classification task.

F. Training Scheme

First, we pretrain the source and target networks with EO samples to obtain rich features and reduce transferring difficulty. Then, we learn the parameters of the attention modules and the target network with few labeled SAR samples. In each epoch, we update parameters with $\mathcal{L}_{\text{transfer}}$ to transfer common features first, then update parameters with $\mathcal{L}_{\text{bayes}}$ to learn to classify images. Finally, we update parameters with both objective functions to learn to transfer common features and classify images together. Algorithm 1 shows the detail of our training scheme.

Algorithm 1: Training scheme.

Input: Dataset of the EO domain$D_S = (X_S, Y_S)$,
dataset of the SAR domain $D_T = (X_T, Y_T)$,
learning rate of the target network β,
learning rate of the attention modules α;
Pretraining:
Pretrain the source network by D_S with $\mathcal{L}_{\text{CE}}$;
Pretrain the target network by D_S with $\mathcal{L}_{\text{bayes}}$;
for t=1,2,3..... **do**:
Calculate $\mathcal{L}_{\text{transfer}}$ with (4) by D_T;
for i=1 to range of θ **do**:
$\mu_{ti} \leftarrow \mu_{t-1i} - \beta \times \gamma_i \nabla_{\mu_i} \sum_{(x,y)\in D_T} \mathcal{L}_{\text{transfer}}$;
$\sigma_{ti} \leftarrow \sigma_{t-1i} - \beta \times \gamma_i \nabla_{\sigma_i} \sum_{(x,y)\in D_T} \mathcal{L}_{\text{transfer}}$;
end for
$\phi_t \leftarrow \phi_{t-1} - \alpha \nabla_{\phi} \sum_{(x,y)\in D_T} \mathcal{L}_{\text{transfer}}$;
Calculate $\mathcal{L}_{\text{bayes}}$ with (10) by D_T;
for i=1 to range of θ **do**:
$\mu_{ti} \leftarrow \mu_{t-1i} - \beta \times \gamma_i \nabla_{\mu_i} \sum_{(x,y)\in D_T} \mathcal{L}_{\text{bayes}}$;
$\sigma_{ti} \leftarrow \sigma_{t-1i} - \beta \times \gamma_i \nabla_{\sigma_i} \sum_{(x,y)\in D_T} \mathcal{L}_{\text{bayes}}$;
end for
$\phi_t \leftarrow \phi_{t-1} - \alpha \nabla_{\phi} \sum_{(x,y)\in D_T} \mathcal{L}_{\text{bayes}}$;
Calculate $\mathcal{L}_{\text{total}}$ with (11) by D_T;
for i=1 to range of θ **do**:
$\mu_{ti} \leftarrow \mu_{t-1i} - \beta \times \gamma_i \nabla_{\mu_i} \sum_{(x,y)\in D_T} \mathcal{L}_{\text{total}}$;
$\sigma_{ti} \leftarrow \sigma_{t-1i} - \beta_{\times} \gamma_i \nabla_{\sigma_i} \sum_{(x,y)\in D_T} \mathcal{L}_{\text{total}}$;
end for
$\phi_t \leftarrow \phi_{t-1} - \alpha \nabla_{\phi} \sum_{(x,y)\in D_T} \mathcal{L}_{\text{total}}$;

IV. Experiment

A. Data Preparation

1) Ship Dataset: The ship dataset of the EO domain [31] contains 4000 RGB 80×80 images, which are taken from planet satellite imagery of the San Francisco Bay area. All the samples of the dataset are used to pretrain the model.

The ship dataset of the SAR domain comes from a public release dataset [32], which contains three classes of images, including 1596 positive samples of the ship, 3192 false-positive samples of ship-like areas, and 9588 negative samples of ocean areas. Fig. 6 shows nine samples of the three classes. In this article, we define a binary classification problem, where each sample is considered to contain ships (positive data points) or no-ship (false positives and negatives). To balance the number of positive and negative samples, the ship dataset of the SAR domain contains all 1596 positive samples, 798 false-positive samples, and 798 negative samples. The false-positive samples and negative samples are randomly selected. The training set are randomly chosen from the dataset, and the testing set consists of all the other samples.

2) Aircraft Dataset: The aircraft dataset of the EO domain [17] contains 3891 positive samples of aircraft and 8154 negative background samples in 408 images.

We build an aircraft dataset of the SAR domain. The dataset contains 224 aircraft from TerraSAR-X of Singapore National Airport, with a resolution of 3 m, 112 aircraft from Wuxi airport of China, with a resolution of 1 m, and 528 negative samples

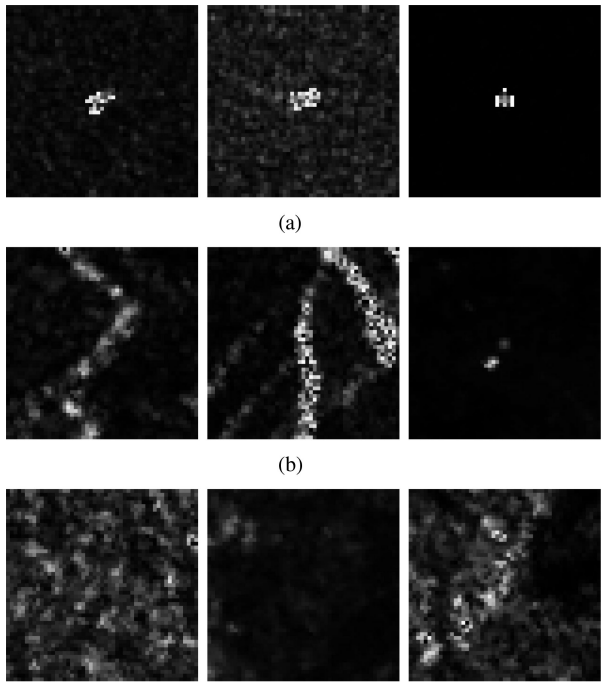


Fig. 6. Samples of the ship dataset in the SAR domain. (a) The samples of positives in images. (b) The samples of false positives in SAR images. (c) The samples of negatives in SAR images.

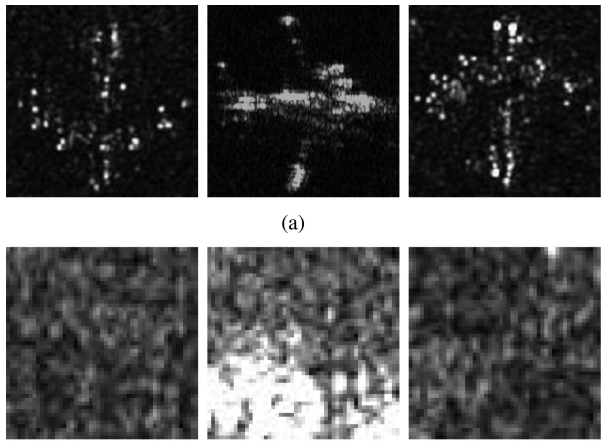


Fig. 7. Samples of the aircraft dataset in the SAR domain. (a) The samples of aircraft in SAR images. (b) The samples of Negtive in SAR images.

come from the background around the aircraft. Fig. 7 shows six examples in the aircraft dataset of the SAR domain. The training set are randomly chosen from the dataset, and the testing set consists of all the other samples.

3) Vehicle Dataset: The vehicle dataset of the EO domain [17] contains 2639 positive samples and 8154 negative samples of background.

The vehicle dataset of the SAR domain is the MSTAR dataset [33], which consists of a training set and a testing set. Fig. 8 shows nine samples of three different classes. The training samples are randomly sampled from the training set, and all the samples of the testing set are used to verify our method.

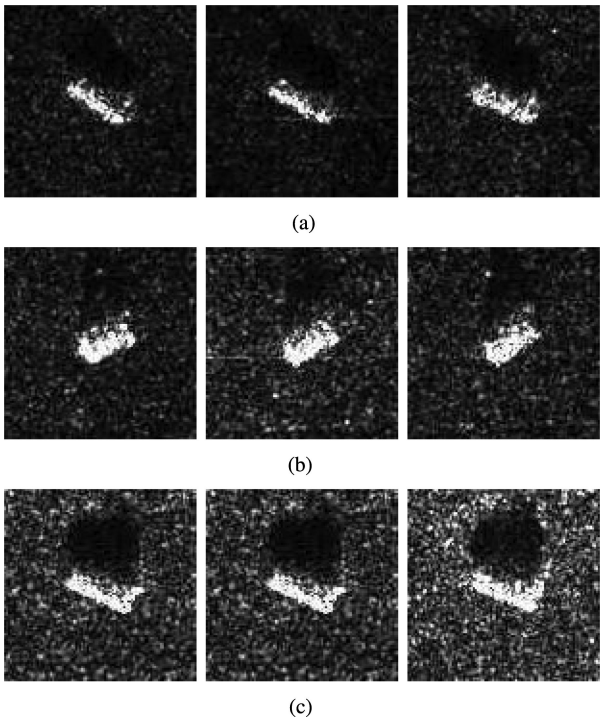


Fig. 8. Samples of the vehicle dataset in the SAR domain. (a) The samples of T62 in SAR images. (b) The samples of D7 in SAR images. (c) The samples of 2S1 in SAR image.

B. Settings

1) Few-Shot Task: The few-shot task assumes that only few labeled data of each category are used to train the network, namely N-way K-shot. For example, in the 2-way 8-shot case, eight examples are selected for each of the two categories. Note that, for the ship dataset, the training set is randomly and equally selected from two kinds of negative samples to balance the number of positive and negative samples. For example, in the 2-way 2-shot case, we randomly take one negative sample from ship-like areas and ocean areas, respectively, when taking two positive samples from ships.

2) Pretraining: All the networks are pretrained by the corresponding EO dataset from scratch until the objective function converges. The batch size is 256 and the learning rate is 0.1. The pretraining takes little time because the EO datasets only contain thousands of samples. All the pretrained parameters are preserved for the tasks of ship and aircraft classification because the number of categories of the EO dataset is the same as that of the SAR dataset. However, for the vehicle classification, pretrained parameters of the FC layer are discarded because the vehicle dataset of the SAR domain contains ten categories of targets, but the EO dataset only has two categories.

3) Training: For the parameters of the attention modules and the target network, the optimizer is Adam with a learning rate of $10e-4$ and 0.1, respectively. The prior distribution is a mixture of the Gaussian distribution whose mean value is 0 and the variance value is 1 and 0.0025. The batch size is all the training samples. We train each model with training samples for 200 epochs and test it in the testing set. The SAR samples are resized to 32×32. We randomly select training samples of the SAR datasets four times, represented as seed 0–3, where "seed" is the value used by the code to generate random numbers. In order to minimize the shift across different imaging methods, the targets of the EO domain and the SAR domain have the same class. For example, the network is pretrained by the ship dataset of the EO domain when the testing sample is the ship in the SAR domain.

C. Compared Methods

Three few-shot learning methods are compared with ours, including the TL-based method [3], the meta-learning-based method [34], and the SAR-pretrained-based method [35]. All the compared methods are trained without extra SAR samples.

1) DTLF: The DTLF [3] is a few-shot learning method that transfers knowledge from the EO domain to the SAR domain with many unlabeled SAR samples.

2) Diversity Transfer Network (DTN): The DTN [34] is a meta-learning-based method that learns to transfer knowledge from similar labeled samples and combine them with support features to generate training samples for novel categories.

3) NWPU: The NWPU [35] is a SAR-pretrained model that pretrain the network on an annotated EO dataset with a top two smooth loss function to tackle label noise and imbalanced class problems.

4) L2T: The L2T [7] is a TL method, which uses attention module to decide what features should be transferred to which layer of the network. It makes a good performance in the limited training samples case.

5) TransMatch: The TransMatch [36] is a semisupervised few-shot learning, which transfer knowledge from unlabeled data. In this article, we replace these unlabeled by few labeled SAR samples.

D. Results

In this article, the results are represented as the accuracy for the image classification. Each accuracy is represented with the mean and standard deviation of ten experiments and the subscript represents the standard deviation.

1) Comparative Results With Other Few-Shot Learning Methods: We compare our method with three few-shot learning methods on all datasets, including the TL-based few-shot learning method [3], the meta-learning-base few-shot learning method [34], and the SAR-pretrained-based method [35]. All the compared methods are trained without extra SAR samples. The results are given in Tables I–III. In these tables, the boldface number means the best performance. So does in Tables IV–IX. We can see that our method achieves the highest accuracy in all cases.

2) Comparative Results With Shallow Networks: Some recent works prove that the shallow network make a similar performance as those complex methods with a deep network in some application scenarios [37]–[40]. Therefore, we compare our method with several shallow networks trained from scratch for the N-way 8-shot case, including A-Net [37], LeNet [41], and VGG-11 [42]. We adjust the hyperparameters to get the best performance of these shallow networks. The results are

TABLE I
COMPARATIVE RESULTS ON THE SHIP DATASET WITH OTHER METHODS FOR 2-WAY K-SHOT IN ACCURACY (IN %)

Shot	8-shot					
Methods	TransMatch	L2T	DTLF	DTN	NWPU	Ours
seed 0	$81.46_{\pm 2.56}$	$80.23_{\pm 2.14}$	$78.97_{\pm 1.55}$	$69.26_{\pm 4.78}$	$74.52_{\pm 2.97}$	**89.24**$_{\pm 1.23}$
seed 1	$78.67_{\pm 2.08}$	$78.54_{\pm 1.86}$	$76.11_{\pm 3.68}$	$68.08_{\pm 3.73}$	$66.21_{\pm 0.25}$	**85.97**$_{\pm 2.72}$
seed 2	$80.97_{\pm 2.48}$	$80.73_{\pm 2.13}$	$79.97_{\pm 3.93}$	$77.05_{\pm 2.96}$	$69.29_{\pm 0.94}$	**86.29**$_{\pm 1.60}$
seed 3	$78.31_{\pm 2.51}$	$78.47_{\pm 1.94}$	$76.25_{\pm 3.19}$	$76.19_{\pm 2.84}$	$75.24_{\pm 0.68}$	**87.16**$_{\pm 2.20}$
	4-shot					
seed 0	$78.05_{\pm 2.38}$	$77.23_{\pm 2.14}$	$77.07_{\pm 2.31}$	$64.73_{\pm 1.73}$	$68.21_{\pm 0.75}$	**81.58**$_{\pm 0.79}$
seed 1	$77.23_{\pm 2.14}$	$71.54_{\pm 2.43}$	$70.78_{\pm 1.26}$	$68.42_{\pm 3.12}$	$67.31_{\pm 0.97}$	**82.14**$_{\pm 1.65}$
seed 2	$71.14_{\pm 2.97}$	$70.45_{\pm 2.69}$	$69.55_{\pm 3.54}$	$66.69_{\pm 7.65}$	$66.92_{\pm 0.93}$	**72.33**$_{\pm 2.19}$
seed 3	$71.95_{\pm 2.76}$	$71.56_{\pm 2.58}$	$69.01_{\pm 4.73}$	$70.53_{\pm 4.87}$	$70.21_{\pm 0.30}$	**81.73**$_{\pm 1.21}$
	2-shot					
seed 0	$73.15_{\pm 2.96}$	$72.99_{\pm 2.44}$	$77.25_{\pm 1.73}$	$69.34_{\pm 4.22}$	$66.53_{\pm 0.28}$	**75.76**$_{\pm 2.14}$
seed 1	$72.06_{\pm 2.73}$	$70.65_{\pm 3.13}$	$71.85_{\pm 2.65}$	$65.35_{\pm 2.88}$	$69.23_{\pm 0.34}$	**79.32**$_{\pm 1.12}$
seed 2	$70.56_{\pm 3.15}$	$71.56_{\pm 2.68}$	$71.91_{\pm 5.04}$	$67.49_{\pm 5.51}$	$66.12_{\pm 0.04}$	**76.92**$_{\pm 2.91}$
seed 3	$68.08_{\pm 3.57}$	$68.12_{\pm 3.14}$	$66.22_{\pm 0.42}$	$66.98_{\pm 1.71}$	$71.21_{\pm 0.12}$	**72.76**$_{\pm 2.23}$

TABLE II
COMPARATIVE RESULTS ON THE AIRCRAFT DATASET WITH OTHER METHODS FOR 2-WAY K-SHOT IN ACCURACY (IN %)

Shot	8-shot					
Methods	TransMatch	L2T	DTLF	DTN	NWPU	Ours
seed 0	$74.33_{\pm 2.78}$	$75.14_{\pm 2.13}$	$74.23_{\pm 1.21}$	$66.12_{\pm 2.43}$	$71.25_{\pm 1.52}$	**84.62**$_{\pm 0.44}$
seed 1	$71.73_{\pm 2.57}$	$71.65_{\pm 1.65}$	$70.70_{\pm 2.15}$	$63.95_{\pm 2.13}$	$70.42_{\pm 0.65}$	**73.12**$_{\pm 0.52}$
seed 2	$72.06_{\pm 2.96}$	$71.37_{\pm 1.95}$	$71.21_{\pm 1.98}$	$64.23_{\pm 2.07}$	$68.32_{\pm 2.76}$	**76.52**$_{\pm 1.43}$
seed 3	$71.84_{\pm 2.04}$	$70.98_{\pm 2.32}$	$70.85_{\pm 2.56}$	$67.85_{\pm 3.50}$	$70.23_{\pm 0.89}$	**75.44**$_{\pm 0.73}$
	4-shot					
seed 0	$71.85_{\pm 3.76}$	$71.54_{\pm 2.62}$	$71.65_{\pm 2.32}$	$63.97_{\pm 2.21}$	$68.45_{\pm 0.69}$	**75.21**$_{\pm 0.32}$
seed 1	$69.18_{\pm 2.45}$	$68.53_{\pm 2.44}$	$68.23_{\pm 2.19}$	$66.21_{\pm 2.23}$	$67.87_{\pm 0.88}$	**70.15**$_{\pm 1.32}$
seed 2	$70.83_{\pm 2.83}$	$70.15_{\pm 2.33}$	$69.43_{\pm 2.71}$	$68.06_{\pm 2.24}$	$69.30_{\pm 1.03}$	**72.13**$_{\pm 1.73}$
seed 3	$69.26_{\pm 2.95}$	$69.13_{\pm 2.25}$	$68.34_{\pm 2.80}$	$66.73_{\pm 3.45}$	$68.22_{\pm 0.31}$	**70.95**$_{\pm 2.75}$
	2-shot					
seed 0	$66.92_{\pm 3.96}$	$66.13_{\pm 2.78}$	$65.22_{\pm 2.20}$	$61.23_{\pm 2.16}$	$66.35_{\pm 1.20}$	**74.21**$_{\pm 2.12}$
seed 1	$61.26_{\pm 3.79}$	$61.87_{\pm 2.54}$	$61.74_{\pm 2.68}$	$61.02_{\pm 2.37}$	$62.95_{\pm 1.34}$	**66.95**$_{\pm 1.76}$
seed 2	$63.16_{\pm 3.69}$	$63.13_{\pm 2.95}$	$62.78_{\pm 2.28}$	$61.93_{\pm 2.26}$	$63.52_{\pm 1.43}$	**68.17**$_{\pm 1.06}$
seed 3	$64.15_{\pm 3.94}$	$63.53_{\pm 2.45}$	$63.01_{\pm 3.24}$	$61.32_{\pm 3.26}$	$63.26_{\pm 1.12}$	**68.91**$_{\pm 1.65}$

TABLE III
COMPARATIVE RESULTS ON THE VEHICLE DATASET WITH OTHER METHODS FOR 10-WAY K-SHOT IN ACCURACY (IN %)

Shot	8-shot					
Methods	TransMatch	L2T	DTLF	DTN	NWPU	Ours
seed 0	$51.16_{\pm 2.96}$	$52.32_{\pm 3.37}$	$42.98_{\pm 4.13}$	$47.91_{\pm 3.79}$	$49.39_{\pm 3.52}$	**65.42**$_{\pm 1.64}$
seed 1	$44.52_{\pm 3.58}$	$43.14_{\pm 3.36}$	$42.70_{\pm 5.88}$	$42.53_{\pm 6.04}$	$47.00_{\pm 4.31}$	**63.79**$_{\pm 1.39}$
seed 2	$53.85_{\pm 3.86}$	$53.83_{\pm 2.13}$	$43.08_{\pm 1.89}$	$51.62_{\pm 3.07}$	$50.59_{\pm 1.75}$	**66.45**$_{\pm 1.12}$
seed 3	$33.86_{\pm 4.16}$	$32.94_{\pm 5.12}$	$46.37_{\pm 3.58}$	$45.72_{\pm 1.50}$	$44.24_{\pm 1.67}$	**60.45**$_{\pm 1.91}$
	4-shot					
seed 0	$40.14_{\pm 2.47}$	$39.84_{\pm 1.96}$	$39.28_{\pm 2.12}$	$36.64_{\pm 2.85}$	$39.96_{\pm 1.69}$	**41.87**$_{\pm 1.74}$
seed 1	$35.94_{\pm 3.81}$	$34.63_{\pm 5.80}$	$37.73_{\pm 2.19}$	$41.02_{\pm 3.37}$	$40.13_{\pm 1.97}$	**43.06**$_{\pm 1.56}$
seed 2	$41.84_{\pm 1.96}$	$29.45_{\pm 8.87}$	$37.43_{\pm 4.71}$	$38.03_{\pm 2.49}$	$40.30_{\pm 2.93}$	**43.55**$_{\pm 2.06}$
seed 3	$38.19_{\pm 3.20}$	$37.10_{\pm 1.55}$	$37.92_{\pm 4.80}$	$36.73_{\pm 1.77}$	$37.22_{\pm 1.30}$	**40.21**$_{\pm 2.64}$
	2-shot					
seed0	$27.07_{\pm 2.91}$	$26.64_{\pm 7.95}$	$30.79_{\pm 2.20}$	$31.04_{\pm 2.06}$	$32.53_{\pm 1.28}$	**35.23**$_{\pm 1.80}$
seed1	$30.59_{\pm 3.16}$	$27.47_{\pm 8.50}$	$32.91_{\pm 1.68}$	$31.02_{\pm 3.37}$	$29.56_{\pm 2.34}$	**34.21**$_{\pm 1.70}$
seed2	$28.14_{\pm 3.85}$	$27.27_{\pm 6.73}$	$32.57_{\pm 4.28}$	$28.23_{\pm 2.19}$	$33.12_{\pm 1.04}$	**38.51**$_{\pm 1.01}$
seed3	$28.95_{\pm 3.50}$	$26.74_{\pm 7.47}$	$29.77_{\pm 2.24}$	$30.76_{\pm 3.26}$	$31.09_{\pm 3.12}$	**33.82**$_{\pm 1.46}$

given in Table IV. We can see that the A-Net outperforms the other two shallow networks and gets a similar performance as ours in seed 0 of the vehicle dataset. This is because the A-Net is specially designed to classify SAR images with limited training samples. However, by comprehensively analyzing the experimental results on three datasets, our method has obvious performance advantages. Besides, the simple structure of A-Net reduces the ability for fitting data, which limits its performance potential.

3) Ablation Experiments for the Training Strategy and Target Network: The results are given in Tables V–VII. In these tables, Direct transfer (DT) means the target network is pretrained by the resized EO samples so that all parameters of the network are pretrained. Fine-tune (FT) means the target network is pretrained by the EO samples with the original size so that the parameters of the FC layer of the target network are randomly initialized. Note that only the target network (do not have the source network) is in the abovementioned learning way. Attention module represents that we use connection-free attention modules to transfer features from the source network to the target network. Besides, Bayes represents the target network is 4-Resblock-based Bayesian-CNN, Bayes_AUIP represents that the AUIP training strategy is used to train the network, and $4-$Res represents the target network is 4-Resblock-based CNN.

The results show that the Bayesian-CNN without the AUIP strategy reaches similar performance as the common CNN. It proves that the key to improve the performance is the AUIP training strategy rather than the Bayesian-CNN structure.

4) Ablation Experiments for the Connection-Free Attention Module: To verify the superiority of the connection-free attention module over the FC-based attention module, we conduct ablation experiments on each dataset for the 8-shot case. Table VIII gives that the attention module brings a maximum accuracy improvement of 3.2%, 7.1%, and 4.7% in the ship, aircraft, and vehicle datasets, respectively. This is because the performance depends on the ratio $N^{0.74}/D$ [43], where N is the number of parameters in the model and D means the data size. The FC-based attention module contains 10^6 parameters, theoretically requiring 27 542 training samples and connection-free attention module contains 10^3 parameters, theoretically requiring 165 training samples. By comparison, in the extreme few-shot case, the FC-based attention module exacerbates the overfitting phenomenon.

5) Ablation Experiments for Samples of Pretraining: To verify that our method is not limited to the class of EO samples, we pretrain our model with EO samples of other classes (ship or aircraft) and train it with few labeled SAR vehicle samples. The results are given in Table IX. Compared with the results in Table III, although the best performance occurs when the EO and SAR samples belong to the same class, pretraining our model with EO samples of other classes also makes a better performance than other few-shot learning methods. This is because the common features still exist between the EO ship or aircraft and SAR vehicles. Thus, these common features can be transferred from the source network to the target network, enhancing the ability of the target network to extract common features.

6) Analysis of the Performance Equivalence: To illustrate the performance of our method, the 4-Resblock-based CNN trained from scratch with different amounts of SAR samples is compared with our method. The colored number in Table X means the performance of our method in the 8-shot case of seed 0 is equivalent to that train the network with 800 SAR images.

TABLE IV
COMPARATIVE RESULTS WITH OTHER SHALLOW NETWORKS FOR N-WAY 8-SHOT IN ACCURACY (IN %)

Dataset	Ship				Aircraft				Vehicle			
Methods	A-Net [37]	LeNet [41]	VGG-11 [42]	Ours	A-Net [37]	LeNet [41]	VGG-11 [42]	Ours	A-Net [37]	LeNet [41]	VGG-11 [42]	Ours
seed 0	$79.21_{\pm 0.95}$	$73.53_{\pm 4.25}$	$76.57_{\pm 2.96}$	**89.24**$_{\pm 1.23}$	$78.43_{\pm 4.23}$	$77.73_{\pm 2.12}$	$78.14_{\pm 1.69}$	**84.62**$_{\pm 0.44}$	$63.19_{\pm 1.27}$	$56.60_{\pm 1.57}$	$54.17_{\pm 1.56}$	**65.42**$_{\pm 1.64}$
seed 1	$78.52_{\pm 4.84}$	$73.83_{\pm 5.34}$	$77.47_{\pm 4.41}$	**85.97**$_{\pm 2.72}$	$69.85_{\pm 1.52}$	$68.48_{\pm 1.43}$	$70.12_{\pm 3.54}$	**73.12**$_{\pm 0.52}$	$59.40_{\pm 2.36}$	$55.49_{\pm 1.46}$	$54.33_{\pm 1.42}$	**63.79**$_{\pm 1.39}$
seed 2	$77.92_{\pm 1.52}$	$78.06_{\pm 2.78}$	$75.89_{\pm 2.17}$	**86.29**$_{\pm 1.60}$	$71.42_{\pm 2.43}$	$70.22_{\pm 3.25}$	$72.76_{\pm 2.31}$	**76.52**$_{\pm 0.43}$	$62.42_{\pm 0.41}$	$59.18_{\pm 1.62}$	$57.26_{\pm 2.60}$	**66.45**$_{\pm 1.12}$
seed 3	$74.26_{\pm 1.77}$	$74.20_{\pm 1.12}$	$77.41_{\pm 3.57}$	**87.16**$_{\pm 2.20}$	$71.92_{\pm 3.28}$	$70.21_{\pm 1.32}$	$71.07_{\pm 2.32}$	**75.44**$_{\pm 0.73}$	$53.90_{\pm 1.39}$	$49.16_{\pm 1.33}$	$45.01_{\pm 1.06}$	**60.45**$_{\pm 1.91}$

TABLE V
RESULTS OF THE ABLATION EXPERIMENT ON THE SHIP DATASET WITH THE DT METHOD AND THE FINE-TUNE (FT) METHOD FOR THE 2-WAY 2-SHOT CASE IN ACCURACY (IN %)

Transfer Method	Attention module			Direct transfer			Fine-Tune transfer		
Target Network	4-Res	Bayes	Bayes_AUIP(ours)	4-Res	Bayes	Bayes_AUIP	4-Res	Bayes	Bayes_AUIP
seed 0	$66.71_{\pm 1.52}$	$74.92_{\pm 0.67}$	**75.76**$_{\pm 2.14}$	$65.99_{\pm 0.01}$	$70.30_{\pm 3.65}$	**73.24**$_{\pm 0.95}$	$62.82_{\pm 0.82}$	$50.57_{\pm 1.79}$	**70.19**$_{\pm 0.58}$
seed 1	$76.67_{\pm 2.32}$	$75.62_{\pm 0.53}$	**79.32**$_{\pm 1.12}$	$69.79_{\pm 0.03}$	$62.07_{\pm 5.03}$	**78.03**$_{\pm 1.16}$	$65.37_{\pm 0.90}$	$50.00_{\pm 0.01}$	**78.31**$_{\pm 2.08}$
seed 2	$71.98_{\pm 1.31}$	$75.56_{\pm 2.11}$	**76.92**$_{\pm 2.91}$	$65.92_{\pm 0.11}$	$61.05_{\pm 4.33}$	**75.18**$_{\pm 2.43}$	$67.41_{\pm 1.17}$	$54.16_{\pm 1.95}$	**68.82**$_{\pm 5.77}$
seed 3	$66.72_{\pm 1.42}$	$71.89_{\pm 0.32}$	**72.76**$_{\pm 2.23}$	$70.82_{\pm 0.02}$	$62.93_{\pm 3.25}$	**71.33**$_{\pm 0.81}$	$66.68_{\pm 1.35}$	$55.17_{\pm 2.47}$	**67.03**$_{\pm 1.35}$

TABLE VI
RESULTS OF THE ABLATION EXPERIMENT ON THE SHIP DATASET WITH THE DT METHOD AND THE FINE-TUNE METHOD FOR THE 2-WAY 4-SHOT CASE IN ACCURACY (IN %)

Transfer Method	Attention module			Direct transfer			Fine-Tune transfer		
Target Network	4-Res	Bayes	Bayes_AUIP(ours)	4-Res	Bayes	Bayes_AUIP	4-Res	Bayes	Bayes_AUIP
seed 0	$75.32_{\pm 1.12}$	$79.22_{\pm 0.62}$	**81.58**$_{\pm 0.79}$	$67.72_{\pm 0.14}$	$67.51_{\pm 6.65}$	**78.22**$_{\pm 3.66}$	$65.53_{\pm 0.34}$	$49.99_{\pm 0.03}$	**78.45**$_{\pm 3.92}$
seed 1	$80.02_{\pm 1.41}$	$80.23_{\pm 0.46}$	**82.14**$_{\pm 1.65}$	$66.71_{\pm 0.03}$	$74.26_{\pm 2.71}$	**80.28**$_{\pm 1.08}$	$68.23_{\pm 0.74}$	$50.00_{\pm 0.01}$	**81.54**$_{\pm 0.98}$
seed 2	$66.56_{\pm 3.21}$	$70.74_{\pm 2.12}$	**72.33**$_{\pm 2.19}$	$65.68_{\pm 0.14}$	$59.95_{\pm 2.39}$	**69.96**$_{\pm 8.06}$	$62.41_{\pm 1.98}$	$49.97_{\pm 0.01}$	**66.32**$_{\pm 1.13}$
seed 3	$75.02_{\pm 3.13}$	$79.92_{\pm 0.71}$	**81.73**$_{\pm 1.21}$	$69.97_{\pm 0.03}$	$70.18_{\pm 1.55}$	**80.47**$_{\pm 0.42}$	$63.55_{\pm 0.81}$	$58.66_{\pm 4.04}$	**78.19**$_{\pm 0.96}$

TABLE VII
RESULTS OF THE ABLATION EXPERIMENT ON THE SHIP DATASET WITH THE DT METHOD AND THE FINE-TUNE METHOD FOR THE 2-WAY 8-SHOT CASE IN ACCURACY (IN %)

Transfer Method	Attention module			Direct transfer			Fine-Tune transfer		
Network	4-Res	Bayes	Bayes_AUIP(ours)	4-Res	Bayes	Bayes_AUIP	4-Res	Bayes	Bayes_AUIP
seed 0	$83.02_{\pm 1.11}$	$82.34_{\pm 1.23}$	**89.24**$_{\pm 1.23}$	$73.14_{\pm 4.27}$	$76.64_{\pm 3.57}$	**81.98**$_{\pm 1.43}$	$67.84_{\pm 3.18}$	$50.11_{\pm 0.27}$	**80.07**$_{\pm 3.91}$
seed 1	$77.42_{\pm 4.53}$	$81.11_{\pm 0.32}$	**85.97**$_{\pm 2.72}$	$64.37_{\pm 0.12}$	$78.65_{\pm 0.61}$	**80.59**$_{\pm 1.02}$	$61.97_{\pm 1.56}$	$51.68_{\pm 1.54}$	**78.13**$_{\pm 3.83}$
seed 2	$82.04_{\pm 6.13}$	$82.21_{\pm 1.32}$	**86.29**$_{\pm 1.60}$	$67.80_{\pm 0.98}$	$65.76_{\pm 4.98}$	**74.15**$_{\pm 4.72}$	$62.87_{\pm 1.43}$	$50.03_{\pm 0.10}$	**69.47**$_{\pm 3.68}$
seed 3	$80.68_{\pm 2.12}$	$83.54_{\pm 0.74}$	**87.16**$_{\pm 2.20}$	$76.99_{\pm 2.61}$	$76.79_{\pm 2.64}$	**80.83**$_{\pm 0.60}$	$66.03_{\pm 2.37}$	$66.97_{\pm 8.45}$	**78.84**$_{\pm 3.65}$

TABLE VIII
RESULTS OF THE ABLATION EXPERIMENT FOR N-WAY 8-SHOT WITH THE FC-BASED ATTENTION MODULE IN ACCURACY (IN %)

Dateset	Ship		Aircraft		Vehicle	
Method	FC-based attention	connection-free attention	FC-based attention	connection-free attention	FC-based attention	connection-free attention
seed 0	$87.42_{\pm 1.23}$	**89.24**$_{\pm 1.23}$	$77.52_{\pm 0.72}$	**84.62**$_{\pm 0.44}$	$60.75_{\pm 2.11}$	**65.42**$_{\pm 1.64}$
seed 1	$82.81_{\pm 1.16}$	**85.97**$_{\pm 2.72}$	$68.84_{\pm 0.72}$	**73.12**$_{\pm 0.52}$	$60.00_{\pm 2.78}$	**63.79**$_{\pm 1.39}$
seed 2	$84.65_{\pm 1.18}$	**86.29**$_{\pm 1.60}$	$71.37_{\pm 0.76}$	**76.52**$_{\pm 0.43}$	$63.71_{\pm 2.21}$	**66.45**$_{\pm 1.12}$
seed 3	$85.06_{\pm 2.07}$	**87.16**$_{\pm 2.20}$	$69.46_{\pm 1.05}$	**75.44**$_{\pm 0.73}$	$56.97_{\pm 2.40}$	**60.45**$_{\pm 1.91}$

TABLE IX
RESULTS OF THE ABLATION EXPERIMENT ON THE VEHICLE DATASET ON 10-WAY 8-SHOT WITH THE DIFFERENT EO DATASET FOR PRETRAINING IN ACCURACY (IN %)

Shot	8-shot			4-shot			2-shot		
Pre-train	Ship	Aircraft	Vehicle	Ship	Aircraft	Vehicle	Ship	Aircraft	Vehicle
seed 0	$63.99_{\pm 1.61}$	$62.18_{\pm 0.84}$	**65.42**$_{\pm 1.64}$	$40.06_{\pm 1.14}$	$39.14_{\pm 0.58}$	**41.87**$_{\pm 1.74}$	$33.99_{\pm 1.14}$	$33.68_{\pm 1.49}$	**35.23**$_{\pm 1.80}$
seed 1	$62.32_{\pm 1.26}$	$58.23_{\pm 0.58}$	**63.79**$_{\pm 1.39}$	$41.54_{\pm 1.95}$	$40.15_{\pm 1.84}$	**43.06**$_{\pm 1.56}$	$33.15_{\pm 1.75}$	$32.95_{\pm 1.25}$	**34.21**$_{\pm 1.70}$
seed 2	$65.11_{\pm 1.37}$	$63.92_{\pm 0.65}$	**66.45**$_{\pm 1.12}$	$42.15_{\pm 1.38}$	$41.88_{\pm 1.26}$	**43.55**$_{\pm 2.06}$	$37.05_{\pm 1.08}$	$35.89_{\pm 1.79}$	**38.51**$_{\pm 1.01}$
seed 3	$59.36_{\pm 1.27}$	$58.13_{\pm 0.46}$	**60.45**$_{\pm 1.91}$	$38.55_{\pm 2.13}$	$38.01_{\pm 2.06}$	**40.21**$_{\pm 2.64}$	$32.55_{\pm 1.56}$	$32.01_{\pm 1.84}$	**33.82**$_{\pm 1.46}$

TABLE X
COMPARATIVE RESULTS WITH THE 4-RESNET-BASED CNN TRAINED FROM SCRATCH WITH DIFFERENT AMOUNT OF SHIP DATA IN SAR IMAGES IN ACCURACY (IN %)

Method/Shot	50-shot	100-shot	300-shot	400-shot	600-shot	800-shot	ours
ST(seed 0)	$76.38_{\pm 6.06}$	$82.55_{\pm 5.75}$	$83.80_{4.89}$	$84.93_{\pm 4.62}$	$87.48_{\pm 4.18}$	$90.01_{\pm 3.23}$	$89.24_{\pm 1.23}$

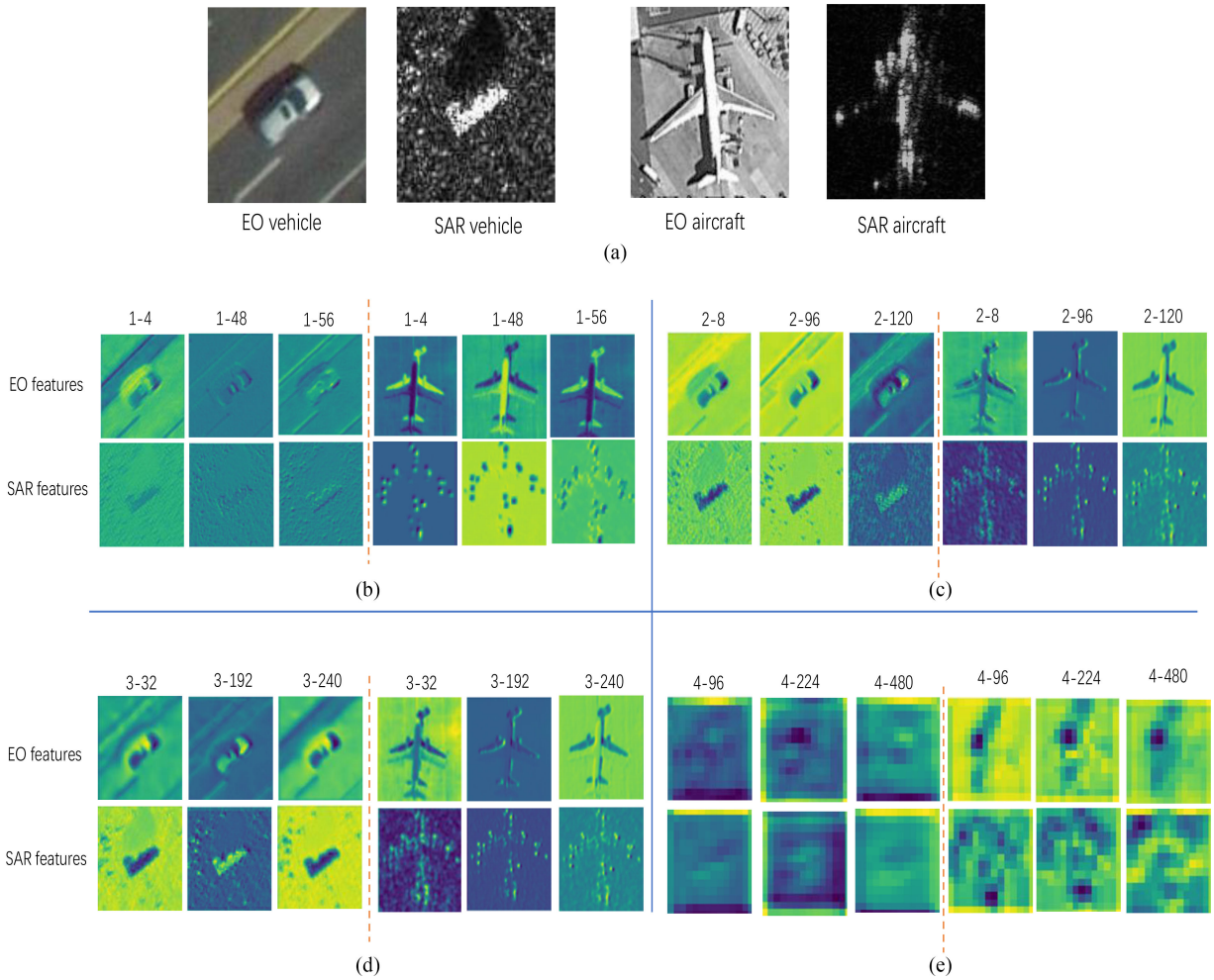


Fig. 9. Comparison of features of EO and SAR samples in the same class extracted from parameters in different layers of the source network. (a) Example samples. (b) Features in the 1st layer. (c) Features in the 2nd layer. (d) Features in the 3rd layer. (e) Features in the 4th layer.

TABLE XI
VALUE OF $\lambda^{m,n}$ OF THE VEHICLE DATASET FOR 10-WAY 8-SHOT IN SEED 0. EACH ROW REPRESENTS THE SOURCE NETWORK LAYER, AND EACH COLUMN REPRESENTS THE LAYER OF THE TARGET NETWORK

S/T	Layer-4	Layer-3	Layer-2	Layer-1
Layer-1	$0.03_{\pm 0.01}$	$0.12_{\pm 0.10}$	$0.11_{\pm 0.07}$	$0.12_{\pm 0.08}$
Layer-2	$0.62_{\pm 0.04}$	$0.76_{\pm 0.05}$	$0.75_{\pm 0.05}$	$0.73_{\pm 0.04}$
Layer-3	$0.66_{\pm 0.02}$	$0.79_{\pm 0.03}$	$0.73_{\pm 0.05}$	$0.72_{\pm 0.05}$
Layer-4	$0.19_{\pm 0.15}$	$0.42_{\pm 0.14}$	$0.34_{\pm 0.04}$	$0.31_{\pm 0.13}$

7) Analysis of Feature-Connection Weight: To understand the process of transferring features, we record the value of $\lambda^{m,n}$ for feature connection in the vehicle dataset for the 10-way 8-shot case in Table XI. Each row represents layers of the source network, and each column represents layers of the target network. As shown in Fig. 9, the features of middle layers are more likely to be transferred to the target model: $\lambda^{3,4} = 0.62$, $\lambda^{3,3} = 0.76$, $\lambda^{3,2} = 0.75$, $\lambda^{3,1} = 0.73$, $\lambda^{2,4} = 0.66$, $\lambda^{2,3} = 0.79$, $\lambda^{2,2} = 0.73$, and $\lambda^{2,1} = 0.72$. The amounts of other feature connections are much smaller than these values. This phenomenon indicates that the features in the middle layers are more likely to be the common features suitable for transferring from the EO domain to the SAR domain.

8) Analysis and Visualization of Common Features: To understand the common features, we visualize features of EO and SAR samples in the same class extracted by different parameters of the source network with the method of Zeiler and Fergus [44]. Specifically, we first pretrain the source network with the corresponding EO dataset until the accuracy reaches 99%. Then, we visualize different features of the source network when inputting an image. The results are shown in Fig. 9.

In Fig. 9, the top four images are the input images, and the below them are feature maps extracted from them by different convolution kernels of different layers. For example, the "1–4" in Fig. 9(b) represents the feature map extracted by the fourth convolution kernel of the first layer of the source network (there are four layers in the source network, see Fig. 2). Compared with the EO features in the first layer, which contain rich target information [see Fig. 9(b)], the SAR features are more about the background. Thus, the features in the first layer are unsuitable to be transferred. Besides, the EO features of the fourth layer [see Fig. 9(e)] are in the last layer of the source network, which is usually thought of as the underlying semantic features. Therefore, the SAR features should be as similar as possible to the EO features if they are common features. However, there

seems to be a big difference between them, which indicates the features in the fourth layer are unlikely to be common features.

In contrast, the second and third layers extract the features related to the target in both EO samples and SAR samples, i.e., shape of the target, which indicate these features are more likely to belong to the common features. Besides, the analysis results of common features are consistent with the weight for transferring, indicating that our method transfers common features from the source network to the target network.

Finally, we conclude that the common features are more likely to appear in the middle layers of the network, making middle-layers features suitable for transferring from the EO domain to the SAR domain. This conclusion is consistent with Huang *et al.* [45]. However, they get the conclusion that "middle layers are worth transferring" through experiments without further studying why these layers are valuable.

E. Discussion

We mainly discuss our method from aspects of performance and the relationship with related few-shot learning methods.

1) Performance: First, compared with other few-shot learning methods, our method brings maximum accuracy improvements of 6.3%, 10.4%, and 14.8% in the extreme few-shot case, for the ship, aircraft, and vehicle datasets, respectively. We explain the reasons as follows.

1) 1) We strengthen common features through transferring them from the source network to the target network with the proposed attention module to supplement the loss of individual features brought by extra SAR samples to some extent.
2) We design an appropriate parameter update strategy with few labeled SAR samples by training those important parameters more accurately to supplement the loss of good initial points brought by extra SAR samples.

Second, as given in Table IX, compared with the meta-network, our attention module brings a maximum accuracy improvement of 3.2%, 7.1%, and 4.7% in the ship, aircraft, and vehicle datasets, respectively. This is because the attention module has fewer parameters, making it fully trained to transfer common features to the target network.

Finally, by comparing the results of seed 2 (third row of each table) in Tables V and VI, we find that the network trained with two samples achieves higher accuracy than that of four samples, which means that some samples may play a villain role in the extreme few-shot case.

2) Relationship With Other Methods: Here, we mainly discuss the relationship between our method and two related methods [3], [7].

First, the similarity between our method and the method in [3] is that we both match the features between two networks. The differences are as follows.

1) We do not use extra SAR samples, but the method in [3] needed extra similar unlabeled SAR samples to provide enough knowledge.
2) We first transfer common features to the target network, then we train the network with few labeled SAR samples, which is a TL-based method, but the method in [3] trained the network with EO samples and SAR samples at the same time, which is a domain adaptation method essentially.
3) We selectively transfer different features, but the method in [3] only matched the high-level semantic features.

Second, the similarity between our method and the method in [7] is that we both transfer features from the source network to the target network by the learning way. Differences are as follows.

1) Our method is designed for the extreme few-shot case, but the method in [7] is not for the few-shot case.
2) The method in [7] constructed meta-networks to transfer features, but the number of parameters of meta-networks reaches 10^6, which cause a difficulty to be fully trained in the extreme few-shot case, but our attention module only contains 10^3 parameters, which obtain a better ability to transfer features.
3) The target network of our method is a Bayesian-CNN so that it can train those important parameters more accurately, but the target network of method in [7] is just a CNN.

For other compared methods, DTN [34] and TransMatch [35] are meta-learning-based methods that follow the episode training strategy. However, our method is a transfer-based method that does not follow that strategy. Besides, NWPU [36] is a simple fine-tune method that does not weight features, but our method transfers "common features" by weighting features and training network with the strategy AUIP.

V. Conclusion and Future Work

This article proposed a novel few-shot transfer learning method for the application scenarios that no extra similar SAR samples were available, namely extreme few-shot learning. In this case, the connection-free attention module was proposed to transfer common features from the source network to the target network, and the training strategy of AUIP was proposed to update important parameters of the Bayesian-CNN more accurately. Also, we find that the common features are more likely to appear in the middle layers of the network, which is valuable for designing more targeted transfer learning methods between the EO domain to the SAR domain. In our future work, how to model the quality of samples and how to deal with these samples differently is a problem that needs to be studied.

References

[1] V. C. Koo *et al.*, "A new unmanned aerial vehicle synthetic aperture radar for environmental monitoring," *Prog. Electromagn. Res.*, vol. 122, pp. 245–268, 2012.
[2] C. L. V. Cooke and K. A. Scott, "Estimating sea ice concentration from SAR: Training convolutional neural networks with passive microwave data," *IEEE Trans. Geosci. Remote Sens.*, vol. 57, no. 7, pp. 4735–4747, Jul. 2019.
[3] M. Rostami, S. Kolouri, E. Eaton, and K. Kim, "Deep transfer learning for few-shot SAR image classification," *Remote Sens.*, vol. 11, no. 11, 2019, Art. no. 1374.
[4] L. Wang, X. Bai, R. Xue, and F. Zhou, "Few-shot SAR automatic target recognition based on Conv-BiLSTM prototypical network," *Neurocomputing*, vol. 443, pp. 235–246, 2021.
[5] A. Romero, N. Ballas, S. E. Kahou, A. Chassang, C. Gatta, and Y. Bengio, "FitNets: Hints for thin deep nets," in *Proc. 3rd Int. Conf. Learn. Representations*, 2015, pp. 1–13.

[6] S. Zagoruyko and N. Komodakis, "Paying more attention to attention: Improving the performance of convolutional neural networks via attention transfer," in *Proc. 5th Int. Conf. Learn. Representations*, 2017.

[7] Y. Jang, H. Lee, S. J. Hwang, and J. Shin, "Learning what and where to transfer," in *Proc. 36th Int. Conf. Mach. Learn.*, 2019, pp. 3030–3039.

[8] J. Hu, L. Shen, S. Albanie, G. Sun, and E. Wu, "Squeeze-and-excitation networks," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 42, no. 8, pp. 2011–2023, Aug. 2020.

[9] S. Woo, J. Park, J. Lee, and I. S. Kweon, "CBAM: Convolutional block attention module," in *Proc. 15th Eur. Conf. Comput. Vis.*, 2018, pp. 3–19.

[10] F. Wang *et al.*, "Residual attention network for image classification," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2017, pp. 6450–6458.

[11] C. Blundell, J. Cornebise, K. Kavukcuoglu, and D. Wierstra, "Weight uncertainty in neural networks," in *Proc. Int. Conf. Mach. Learn.*, 2015, pp. 1613–1622.

[12] Y. Wang, Q. Yao, J. T. Kwok, and L. M. Ni, "Generalizing from a few examples: A survey on few-shot learning," *ACM Comput. Surv.*, vol. 53, no. 3, pp. 1–34, 2020.

[13] Z. Huang, Z. Pan, and B. Lei, "Transfer learning with deep convolutional neural network for SAR target classification with limited labeled data," *Remote Sens.*, vol. 9, no. 9, 2017, Art. no. 907.

[14] D. Zhang, J. Liu, W. Heng, K. Ren, and J. Song, "Transfer learning with convolutional neural networks for SAR ship recognition," in *Proc. IOP Conf. Ser. Mater. Sci. Eng.*, 2018, Art. no. 072001.

[15] R. Shang, J. Wang, L. Jiao, R. Stolkin, B. Hou, and Y. Li, "SAR targets classification based on deep memory convolution neural networks and transfer parameters," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 11, no. 8, pp. 2834–2846, Aug. 2018.

[16] D. Ha, A. M. Dai, and Q. V. Le, "Hypernetworks," in *Proc. 5th Int. Conf. Learn. Representations*, 2017. [Online]. Available: https://openreview.net/forum?id=rkpACe1lx

[17] H. Zhu, X. Chen, W. Dai, K. Fu, Q. Ye, and J. Jiao, "Orientation robust object detection in aerial images using deep convolutional neural network," in *Proc. IEEE Int. Conf. Image Process.*, 2015, pp. 3735–3739.

[18] M. Ji, B. Heo, and S. Park, "Show, attend and distill: Knowledge distillation via attention-based feature matching," in *Proc. 35th AAAI Conf. Artif. Intell. 33rd Conf. Innov. Appl. Artif. Intell. 11th Symp. Educ. Adv. Artif. Intel.*, 2021, pp. 7945–7952.

[19] T. By *et al.*, "Bayesian methods for adaptive models,"Ph.D. dissertation, *California Inst. Technol.*, Pasadena, CA, USA, 1992.

[20] D. J. C. MacKay, "A practical Bayesian framework for backpropagation networks," *Neural Comput.*, vol. 4, no. 3, pp. 448–472, 1992.

[21] G. E. Hinton and D. van Camp, "Keeping the neural networks simple by minimizing the description length of the weights," in *Proc. 6th Annu. ACM Conf. Comput. Learn. Theory*, 1993, pp. 5–13.

[22] A. Graves, "Practical variational inference for neural networks," in *Proc. Int. Conf. Neural Inf. Process. Syst.*, 2011, pp. 2348–2356.

[23] J. M. Hernández-Lobato and R. P. Adams, "Probabilistic backpropagation for scalable learning of Bayesian neural networks," in *Proc. 32nd Int. Conf. Mach. Learn.*, 2015, pp. 1861–1869.

[24] A. Kendall and Y. Gal, "What uncertainties do we need in Bayesian deep learning for computer vision?," in *Proc. Int. Conf. Neural Inf. Process. Syst.*, 2017, pp. 5574–5584.

[25] A. Kendall, Y. Gal, and R. Cipolla, "Multi-task learning using uncertainty to weigh losses for scene geometry and semantics," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2018, pp. 7482–7491.

[26] S. Ebrahimi, M. Elhoseiny, T. Darrell, and M. Rohrbach, "Uncertainty-guided continual learning with Bayesian neural networks," in *Proc. 8th Int. Conf. Learn. Representations*, 2020.

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

[28] A. Krizhevsky and G. Hinton, "Convolutional deep belief networks on CIFAR-10," *Unpublished Manuscript*, vol. 40, no. 7, pp. 1–9, 2010.

[29] J. McClelland, B. L. McNaughton, and R. C. O'Reilly, "Why there are complementary learning systems in the hippocampus and neocortex: Insights from the successes and failures of connectionist models of learning and memory," *Psychol. Rev.*, vol. 102, pp. 419–457, 1995.

[30] C. Blundell, J. Cornebise, K. Kavukcuoglu, and D. Wierstra, "Weight uncertainty in neural network," in *Proc. 32nd Int. Conf. Mach. Learn.*, 2015, pp. 1613–1622.

[31] R. Hammell, "Data retrieved from Kaggle." Accessed: Feb. 1, 2019. [Online]. Available: https://www.kaggle.com/rhammell/ships-in-satellite-imagery

[32] C. P. Schwegmann, W. Kleynhans, B. Salmon, L. W. Mdakane, and R. G. V. Meyer, "A SAR Ship Dataset for Detection, Discrimination and Analysis," *Distributed by IEEE Dataport*, 2017, doi: 10.21227/H2RK82.

[33] J. R. Diemunsch and J. Wissinger, "Moving and stationary target acquisition and recognition (MSTAR) model-based automatic target recognition: Search technology for a robust ATR," in *Proc. Algo. Synth. Aperture Radar Imagery V*, 1998, vol. 3370, pp. 481–492.

[34] M. Chen *et al.*, "Diversity transfer network for few-shot learning," in *Proc. 34th AAAI Conf. Artif. Intell.*, 2020, pp. 10559–10566.

[35] Z. Huang, C. O. Dumitru, Z. Pan, B. Lei, and M. Datcu, "Classification of large-scale high-resolution SAR images with deep transfer learning," *IEEE Geosci. Remote Sens. Lett.*, vol. 18, no. 1, pp. 107–111, Jan. 2021.

[36] Z. Yu, L. Chen, Z. Cheng, and J. Luo, "TransMatch: A transfer-learning scheme for semi-supervised few-shot learning," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2020, pp. 12853–12861.

[37] S. Chen, H. Wang, F. Xu, and Y.-Q. Jin, "Target classification using the deep convolutional networks for SAR images," *IEEE Trans. Geosci. Remote Sens.*, vol. 54, no. 8, pp. 4806–4817, Aug. 2016.

[38] S. Zagoruyko and N. Komodakis, "Wide residual networks," in *Proc. Brit. Mach. Vis. Conf. (BMVC)*, 2016, pp. 87.1–87.12 Art. no. 87.

[39] C.-H. Chang, "Deep and shallow architecture of multilayer neural networks," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 26, no. 10, pp. 2477–2486, Oct. 2015.

[40] S. S. Mannelli, E. Vanden-Eijnden, and L. Zdeborová, "Optimization and generalization of shallow neural networks with quadratic activation functions," in *Advances in Neural Information Processing Systems*, H. Larochelle, M. Ranzato, R. Hadsell, M. F. Balcan, and H. Lin, Eds. Curran Associates, vol. 33, 2020, pp. 13445–13455.

[41] Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proc. IEEE*, vol. 86, no. 11, pp. 2278–2324, Nov. 1998.

[42] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," in *Proc. 3rd Int. Conf. Learn. Representations*, 2015.

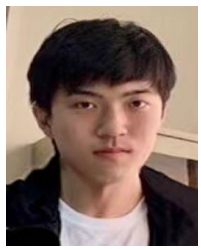
[43] J. Kaplan *et al.*, "Scaling laws for neural language models," 2020, *arXiv:2001.08361*.

[44] M. D. Zeiler and R. Fergus, "Visualizing and understanding convolutional networks," in *Proc. 13th Eur. Conf. Comput. Vis.*, 2014,pp. 818–833.

[45] Z. Huang, Z. Pan, and B. Lei, "What, where, and how to transfer in SAR target recognition based on deep CNNs," *IEEE Trans. Geosci. Remote Sens.*, vol. 58, no. 4, pp. 2324–2336, Apr. 2020.

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

[47] S. Ahn, S. X. Hu, A. Damianou, N. D. Lawrence, and Z. Dai, "Variational information distillation for knowledge transfer," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2019, pp. 9155–9163.



Yuan Tai received the B.E. degree in automation from the Huazhong University of Science and Technology, Wuhan, China, in 2017. He is currently working toward the Ph.D. degree with the National Key Laboratory of Science and Technology on Multi-Spectral Information Processing, School of Artificial Intelligence and Automation, Huazhong University of Science and Technology, Wuhan China.

Yihua Tan (Member, IEEE) received the Ph.D. degree in pattern recognition and intelligent systems from the Huazhong University of Science and Technology (HUST), Wuhan, China, in 2004.

Since 2005, he has been with the School of Artificial Intelligence and Automation, HUST, where he is currently a Professor. From 2005 to 2006, he was a Postdoctoral Staff with the Department of Electronics and Information, HUST. From 2010 to 2011, he was a Visiting Scholar with Purdue University, where he worked on remote sensing image analysis. He has authored more than 80 papers in journals and conferences. His research interests include digital image/video processing and analysis, object detection and classification, and machine learning.

Shengzhou Xiong received the master's degree in pattern recognition and intelligent systems in 2017 from the Huazhong University of Science and Technology, Wuhan, China, where he is currently working toward the Ph.D. degree with the National Key Laboratory of Science and Technology on Multi-Spectral Information Processing, School of Artificial Intelligence and Automation.

Zhaojin Sun is currently working toward the B.E degree in automation with the School of Artificial Intelligence and Automation, Huazhong University of Science and Technology, Wuhan, China.

Jinwen Tian received the Ph.D. degree in pattern classification and intelligent systems from the Huazhong University of Science and Technology (HUST), Wuhan, China, in 1998.

He was with the School of Artificial Intelligence and Automation, HUST, where he is currently a Professor. His research interests include remote sensing image analysis, image compression, computer vision, and fractal geometry.