International Journal of Computer Vision (2023) 131:385–404
https://doi.org/10.1007/s11263-022-01700-x

Few-Shot Learning with Complex-Valued Neural Networks and Dependable Learning

Runqi Wang[1] · Zhen Liu[1] · Baochang Zhang[1] · Guodong Guo[2,3,4] · David Doermann[5]

Received: 20 August 2021 / Accepted: 30 September 2022 / Published online: 31 October 2022

Abstract

We present a flexible, general framework for few-shot learning where both inter-class differences and intra-class relationships are fully considered to improve recognition performance significantly. We introduce complex-valued convolutional neural networks (CNNs) to describe the subtle difference among inter-class samples and Dependable Learning to capture the intra-class relationship. Conventional CNNs use only real-valued CNNs and fail to extract more detailed information. Complex-valued CNNs, on the other hand, can provide amplitude and phase information to enhance the feature representation ability based on the proposed complex metric module (CMM). Building upon the recent episodic training mechanism, CMMs can improve the representation capacity by extracting robust complex-valued features to facilitate the modeling of subtle relationships among few-shot samples. Furthermore, we use Dependable Learning as a new learning paradigm, to promote a robust model against perturbation based on a new bilinear optimization to enhance the feature extraction capacity for very few available intra-class samples. Experiments on two benchmark datasets show that the proposed methods significantly improve the performance over other approaches and achieve state-of-the-art results.

Keywords Few-shot learning · Complex-valued CNNs · Dependable learning

1 Introduction

Great progress has been made on a variety of visual understanding tasks (He et al., 2016; Simonyan & Zisserman, 2014; Szegedy et al., 2015; Zeiler & Fergus, 2014) due to the advancement of deep learning models and the development of large amounts of labeled training data. Although deep learning has made widespread breakthroughs, its performance is significantly deteriorated in some scenarios due to limited amounts of labeled data. In contrast, humans can recognize new classes with a few labeled examples or samples of different but similar classes (Li & Fergus, 2006). One of the ultimate goals of CNNs is to match or outperform humans in any given task. It is imperative to have minimal dependency on large balanced labeled datasets with which current models can be successful. However, for other tasks where the labeled data is scarce (few samples only), the performance of the respective models drops significantly. Few-shot learning aims to solve this problem by using deep learning to recognize unlabeled samples with few labeled instances.

A variety of few-shot learning methods have been proposed, roughly divided into meta-learning, data augmentation and metric learning. The meta-learning methods optimize the (hyper) parameters of neural networks so that the models can quickly and efficiently adapt to the new task (Gidaris & Komodakis, 2018, 2019). Data augmentation is widely used to generate more samples from a small number

Communicated by Oisin Mac Aodha.

✉ Baochang Zhang
bczhang@buaa.edu.cn

Runqi Wang
runqiwang@buaa.edu.cn

Zhen Liu
liuzhenbuaa@buaa.edu.cn

Guodong Guo
guoguodong01@baidu.com

David Doermann
doermann@buffalo.edu

[1] Beihang University, Beijing, China

[2] Ant Group, Beijing, China

[3] Institute of Deep Learning, Baidu Research, Beijing, China

[4] National Engineering Laboratory for Deep Learning Technology and Application, Beijing, China

[5] University at Buffalo, Buffalo, NY, USA

of available instances (Liu et al., 2019; Zhang et al., 2018). With more training samples, data augmentation can significantly alleviate the over-fitting problem. Metric learning is a promising method for the few-shot classification problem. It learns a mapping from images to an embedding space, where samples from the same class become closer. To make full use of the limited data, metric learning methods (Snell et al., 2017; Sung et al., 2018; Vinyals et al., 2016) focus primarily on learning accurate relationships among samples using different metrics. Meta-learning based methods attempt to train a base learner, which can be quickly adapted in the presence of a few novel class examples. The drawbacks of existing methods are that they pay less attention to feature extraction, which is a deterministic element for the final performance (Szegedy et al., 2015). The traditional methods use only conventional real-valued convolution neural networks (CNNs) for feature representation. Due to the scarcity of samples in few-shot learning, feature extraction and learning should be more flexible and fully consider intra-class and inter-class information in the same framework.

As illustrated in Fig. 1, we provide a new framework to model inter-class and intra-class relationships. We introduce complex-valued CNNs to enhance feature representation by capturing subtle differences among samples to describe the inter-class relationships better. Complex-valued CNNs provide amplitude and phase information. The proposed complex metric module (CMM) (Liu et al., 2020) extracts phase information and can enlarge the inter-class distance (Trabelsi et al., 2018). Dependable Learning is further introduced to better describe the intra-class relationship based on a new attention method that facilitates the extraction of the correct features from a small number of samples. In particular, the attention can change the background distribution and highlight the feature information of the object, just like associating the representation of the object in different backgrounds, to enhance the ability of the model to express the intra-class relationship.

Unlike conventional few-shot learning methods based on real-valued features, we introduce complex-valued CNNs to enhance the discrimination ability of the feature representation by using richer amplitude and phase information. We also propose a unique metric learning method, which can measure the embedding distances among samples with amplitude and phase information. Using the distance metric, we utilize the entire query set for transductive inference to deal with the few-shot problem. Specifically, we develop a novel Complex Metric Module (CMM) by combining deep complex-valued CNNs and complex-valued distance metrics in the same framework. First, we map the input images to an embedding space using these complex-valued CNNs. Then, we measure the sample relations using the complex-valued metric and embedding. With a method of

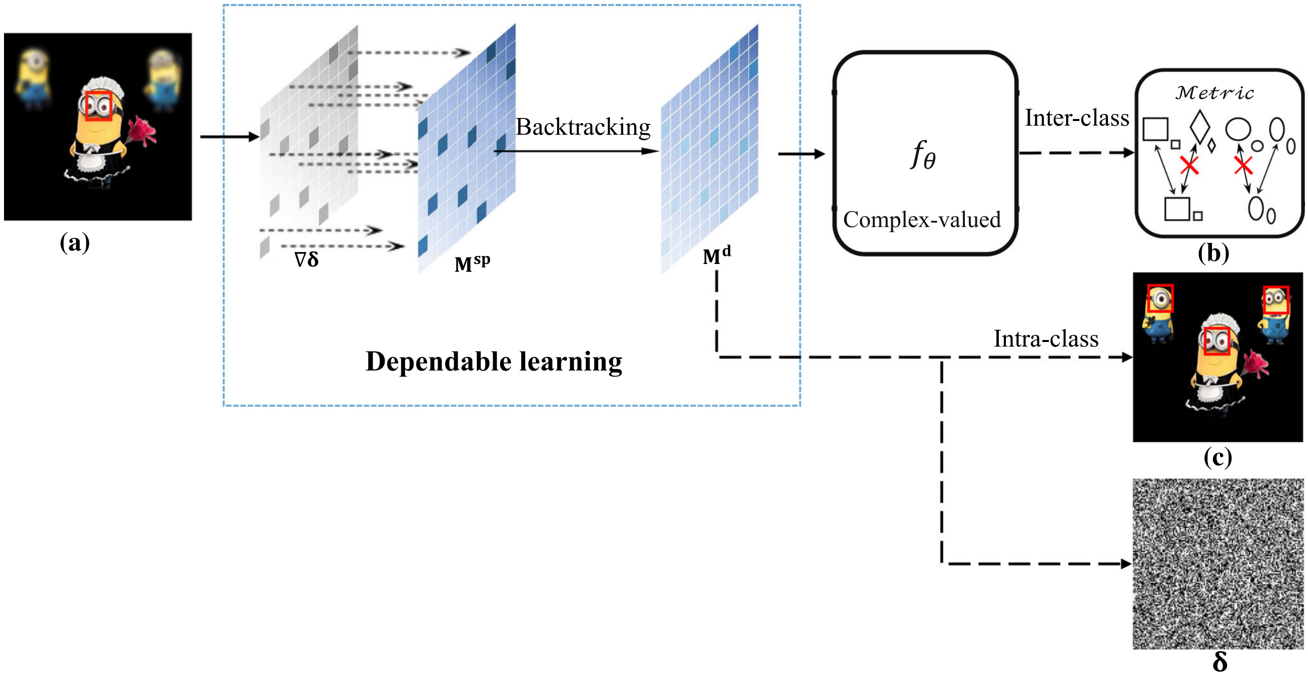


Fig. 1 An Illustration of few-shot learning based on complex-valued CNNs and Dependable Learning. **a** The perturbation, which causes attention to be lost from the two minions in the back. In (**b**), the complex-valued features are used to recognize the difference between squares and parallelograms and to correctly predict the label of the unlabeled inter-class samples. **c** Dependable Learning leads to attention that can be more effectively focused on information-rich areas and thus enhances the intra-class feature representation ability

transductive inference, we compute the cross-entropy loss using the support-query and query–query scores. Finally, we use the recent episodic training mechanism to update all parameters end-to-end during back-propagation.

Another bottleneck in few-shot learning is that the intra-class features clustering ability of the model is insufficient when there are few training samples. To address this issue, we introduce Dependable Learning based on adversarial training to directly incorporate adversarial examples into the training process (Mustafa et al., 2019; Shafahi et al., 2019), leading to a new learning paradigm and a generic approach to obtain intra-class information robustly. Due to the shortage of training samples in few-shot learning, the image background has more influence on the model (Wang et al., 2020). To address the issue, we do not want the perturbation to affect the model's learning of the intra-class relationship, which requires attention to resist the sharp perturbation. To this end, we backtrack the attention suffered from a sharp perturbation to achieve a dependable feature representation. Our method is easily implemented within the adversarial training framework. This is which extends our conference version (Liu et al., 2020) by including: (1) a Dependable Learning method is introduced to associate the intra-class features with different backgrounds; (2) more experiments conducted to validate the effectiveness of our method; (3) a theoretical investigation to show that our Dependable Learning can enhance causal learnability (Amar et al., 2021) and provide a robustness guarantee for the few-shot recognition. The main contributions of our work are as follows:

- We, for the first time, introduce complex-valued deep neural networks into few-shot learning, which explore the amplitude and phase information to enhance the discrimination ability of feature representation.
- We introduce a Dependable Learning method to formulate the attention and perturbation as a bilinear optimization problem, based on a backtracking method to obtain a robust model against perturbations.
- Experimental results on the miniImageNet and tieredImageNet public datasets show that the proposed method dramatically improves the 1-shot and 5-shot accuracy over the state-of-the-art methods.

The rest of our paper is organized as below. Section 2 introduces few-shot learning, complex-valued CNNs and adversarial training. In Sect. 3, a complex-valued CNN architecture for few-shot learning is proposed from the perspective of increasing the inter-class distance. In Sect. 4, Dependable Learning method is introduced to reduce the intra-class distance. In Sect. 5, metric learning and meta-learning tasks are used to verify the proposed method, and a series of ablation experiments are performed.

2 Related Work

2.1 Few-Shot Learning

In recent years, there has been a growing interest in few-shot learning. In Lake et al. (2015), a hierarchical Bayesian model was introduced to achieve human-level accuracy on alphabet recognition tasks in the setting of few-shot learning. Gregory Koch et al. (2015) first introduced the Siamese network, which computes the pair-wise distance between samples to classify unlabeled samples by the k-nearest neighbor algorithm for few-shot learning. In Snell et al. (2017), a prototype representation of each class was built to describe better few-shot samples based on the mean of sample embedding features. Flood Sung et al. (2018) considered that the measurement method is also an essential part of the network, which needs to be modeled and trained using a relation network (RelationNet) (such as CNN) to learn the measurement method distance. More recently, meta-learning-based approaches have been introduced. Sachin Ravi and Hugo Larochelle et al. (2017) designed a model updating the weights of a classifier using a Long Short-Term Memory (LSTM) approach. In Finn et al. (2017), a model agnostic meta-learning (MAML) algorithm was proposed to find parameters sensitive to changes in the task with a small number of samples. Another line of few-shot learning research directly solves the over-fitting problem by data augmentation (Wang et al., 2020).

Metric learning (Fehervari et al., 2019) is one of the most effective categories of few-shot learning approaches. The approach first learns a representation of a sample or class (*it depends on whether inter-class information is considered*) and then calculates the relation scores between query samples and support samples using a metric method. Siamese networks (Koch et al., 2015) can be used to extract features embedded in a supervised way. Calculating the distances of sample pairs estimates whether they belong to the same class and generates the corresponding probability distributions. Matching network (Vinyals et al., 2016) included different encoders for the support set and the query set, and the output of the final classifier is a weighted sum of the predicted relation values of the support and query samples. Prototype network (Snell et al., 2017) was based on the idea that there is a prototype representation for each class, and the prototype of the class is the mean of the support set in the embedding space. Then, the classification problem becomes finding the nearest neighbor in the embedding space. Flood Sung et al. (2018) believed that the metric is an essential part of the model, and a single fixed distance metric may not be optimal, so they trained a network to learn a better distance metric.

2.2 Complex-Valued Neural Networks

Complex-valued neural networks recently receive increasing attention (Trabelsi et al., 2018; Zhang et al., 2017), due to their potential to enable easier optimization (Nitta, 2002), better generalization (Hirose & Yoshida, 2012), and fast learning (Arjovsky et al., 2016). They are proved to have a richer representational capacity than real-valued models. These models can extract complex-valued features which consist of both amplitude and phase messages. The phase component is important from a biological point of view and a signal processing perspective. The amount of information present in The phase of an image is sufficient to recover the majority of the information encoded in its magnitude (Zhang et al., 2006). The phase describes objects by encoding shapes, edges, and orientations (Trabelsi et al., 2018). Using complex parameters has many advantages from computational and biological perspectives (Trabelsi et al., 2018; Zhang et al., 2017). In terms of computation, Ivo Danihelka et al. (2016) showed that holographic reduced representations store more information with complex-valued parameters and to efficient and stable retrieval from an associative memory. Unitary-RNN (Arjovsky et al., 2016) learns a unitary weight matrix, a complex generalization of orthogonal weight matrices with the absolute value of the eigenvalues equal to 1. Compared with other orthogonal counterparts, Unitary-RNN (Arjovsky et al., 2016) can be easier optimized, provide a richer representation, and show the potential in demanding tasks involving long-term dependencies. Using complex weights in neural networks is also biologically meaningful (Reichert & Serre, 2014) where a neural network formulation based on complex-valued neuronal units is introduced. These units are attributed with a fire rate and a phase, which can build richer and more versatile networks. The complex-valued formula allows one to express the output of neurons according to their firing rate and relative time of activity. The amplitude of a complex neuron represents the former, and its phase represents the latter. Moreover, input neurons with similar phases are viewed as synchronous since they add constructively, while asynchronous neurons increase destructively. Also, David P Reichert and Serre (2014) showed that this flexible mechanism of neuronal synchrony fulfills multiple functional roles in deep networks.

Existing works (Trabelsi et al., 2018; Mönning & Manandhar, 2018) perform over-fitting analysis on complex-valued CNNs. In Trabelsi et al. (2018), complex-valued CNNs are used in three different tasks and all the loss curves have a trend of regular decline like that of real-valued CNNs, which shows that the complex-valued CNNs are not easier to over-fitting compared to real-valued CNNs. Besides, Mönning and Manandhar (2018) conducts a detailed comparison of over-fitting based on the network structure of complex-valued neural networks and real-valued neural networks. Specifically, Mönning and Manandhar (2018) deepens complex-valued neural networks and their real-valued counterparts by adding the hidden layers continuously, and then compares the test performance. The results show that there is no obvious difference in over-fitting between complex-valued models and real-valued models.

2.3 Adversarial Training

The success of deep learning models has been demonstrated on various computer vision tasks, such as image classification, instance segmentation, and object detection. However, existing deep models are sensitive to adversarial attacks (Carlini & Wagner, 2017; Goodfellow et al., 2015; Szegedy et al., 2013), where adding an imperceptible perturbation to input images causes the models to perform incorrectly. Furthermore, Szegedy et al. (2013) observes that these adversarial examples are transferable across multiple models such that adversarial examples generated by one model might mislead other models. Therefore, models deployed in real-world scenarios are susceptible to adversarial attacks (Liu et al., 2016). After discovering adversarial examples by Szegedy et al. (2013), Goodfellow et al. (2015) proposes the Fast Gradient Sign Method (FGSM) to generate adversarial examples with a single gradient step. Later, in Kurakin et al. (2016), the researchers propose the Basic Iterative Method (BIM), which takes multiple and smaller FGSM steps to improve FGSM, but BIM renders the adversarial training very slow. This iterative adversarial attack is further strengthened by adding multiple random restarts into the adversarial training procedure. In addition, the projected gradient descent (PGD) (Madry et al., 2018) adversary attack, a variant of BIM with uniform random noise as initialization, is recognized as one of the most powerful first-order attacks (Athalye et al., 2018). Other popular attacks include the Carlini and Wagner (2017), Momentum Iterative Attack (Dong et al., 2018), and Diverse Input Iterative Attack (Xie et al., 2019). Among them, Carlini and Wagner (2017) devises state-of-the-art attacks under various pixel-space l_p norm-ball constraints by proposing multiple adversarial loss functions.

Many methods have been proposed to defend against these attacks (Szegedy et al., 2013; Cubuk et al., 2017). A category of defense methods improves the network's training regime to counter adversarial attacks. The most common way is adversarial training (Kurakin et al., 2016; Na et al., 2017; Tramèr et al., 2017) with adversarial examples added to the training data. In Madry et al. (2018), the researchers propose a Min-Max optimization defense method, which augments the training data with first-order attack samples. Wang and Zhang (2019) investigates the fast training of adversarially robust models to perturb both images and the labels during training. There are also some model defense methods (Athalye et al., 2018; Gupta & Rahtu, 2019; Liao et al., 2018; Wang &

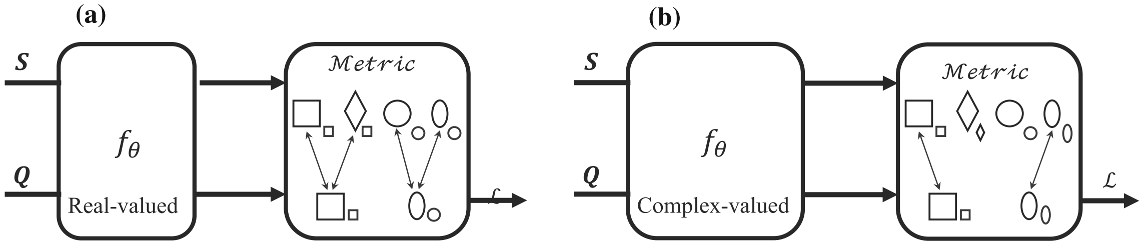


Fig. 2 Illustration of the motivation for complex-valued networks. S means support samples, and Q means query samples. Large geometry represents real features, and small geometry represents extracted features. **a** A real-valued feature extractor extracting only amplitude features cannot find the difference between squares and parallelograms and make a wrong prediction of the unlabeled sample. **b** A complex-valued feature extractor that extracts phase features can recognize the difference between squares and parallelograms and correctly predict the class of the unlabeled inter-class sample

Zhang, 2019; Ye et al., 2019). that target removing adversarial perturbation, by transforming the adversarial images into clean images before they are fed into the classifier. However, the adversarial samples can be associated with clean samples to improve the performance of classification (Wang et al., 2022). In Das et al. (2017), the researchers study the effect of JPEG compression on removing adversarial noise.

This paper introduces complex-valued CNNs to enhance the feature representation based on phase and amplitude information. We build upon the recent episodic training mechanism (Liu et al., 2019) and extract robust complex-valued features to model subtle relationships among few-shot samples. Moreover, we propose Dependable Learning to obtain a robust model against perturbation to enhance the extraction of the correct features for very few available samples.

3 Complex-Valued Architecture

We illustrate complex CNNs in Fig. 2 which shows that with the phase information, our complex-valued model can correctly find the subtle difference of inter-class samples. These samples include ellipses, circles, squares, and parallelograms, therefore facilitating the prediction of unlabeled samples. The proposed method is illustrated in Fig. 3, which utilizes both the amplitude and phase information of complex-valued CNNs to improve the performance for the few-shot classification problem.

3.1 Problem Definition

Typically, for few-shot classification tasks, there are two datasets: training set $\mathcal{D}_{train}$ and test set $\mathcal{D}_{test}$, which do not share the same categories. Generally speaking, $\mathcal{D}_{train}$ contains many classes, each of which has multiple samples. In the training stage, we randomly select C categories in $\mathcal{D}_{train}$, K samples of each category as the labeled data which form the support set (S_S), and then select Q samples from the remaining data of these C categories as unlabeled data, which forms the query set (S_Q). The model must learn to distinguish these $C * Q$ samples in the C categories. Such a task is called the C-way K-shot problem. In each task, the selected data are $(S_S, y_S, S_Q, y_Q) = (\mathcal{I}_1, \mathcal{I}_2, \ldots, \mathcal{I}_{C*(K+Q)}; y_1, y_2, \ldots, y_{C*(K+Q)})$, where $\mathcal{I}_i$ and y_i denote an image and its label respectively.

Each episode (Vinyals et al., 2016) samples different meta-tasks in the training process, including various combinations of classes. This mechanism enables the model to learn the common knowledge of different meta-tasks, such as extracting important features, comparing samples, and forgetting the task-related parts. Through this learning mechanism, samples can be classified for new meta-tasks.

3.2 Complex Metric Module

The proposed Complex Metric Module (CMM) features a complex-valued convolution neural network and a complex metric unit to measure relationships between samples.

3.2.1 Complex-Valued Feature Representation

To fully use limited samples, we employ complex convolutions and other corresponding components, including complex batch-normalization, complex pooling, and complex Relu strategies (Trabelsi et al., 2018) for complex-valued CNNs. The rule is different from traditional CNNs. We assume there is an input $\mathcal{I} = X + Yi$, and a complex filter matrix $W = A + iB$. X is an image matrix, Y is initialized to $\mathbf{0}$. A and B are real matrices since we simulate complex arithmetic using real-valued entities. Specifically, the rule of complex convolution is

$$\begin{bmatrix} \Re(\mathcal{I} * \mathrm{W}) \\ \Im(\mathcal{I} * \mathrm{W}) \end{bmatrix} = \begin{bmatrix} \mathrm{A} & -\mathrm{B} \\ \mathrm{B} & \mathrm{A} \end{bmatrix} * \begin{bmatrix} \mathrm{X} \\ \mathrm{Y} \end{bmatrix}. \tag{1}$$

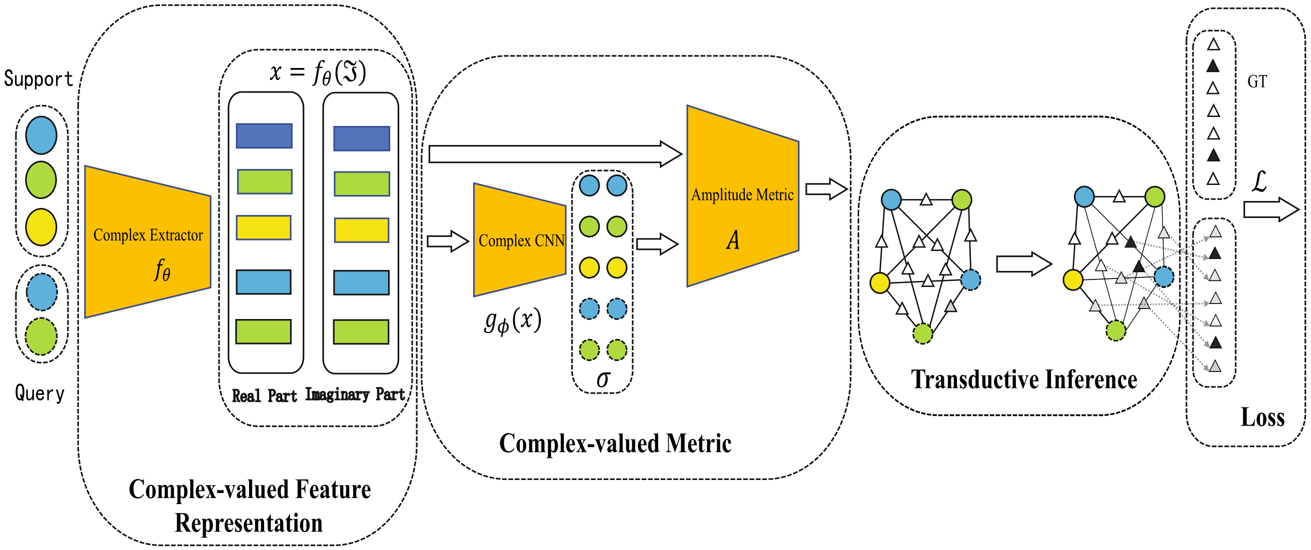


Fig. 3 The overall framework of our model is one in which a sample-wise parameter learns amplitude and phase information. It comprises four components: complex-valued feature representation, complex-valued metric, transductive inference, and loss generation

Complex Relu (ℂRelu) and complex Pooling (ℂPooling) both act on the real and imaginary parts of a neuron separately.

$$\mathbb{C}\text{Relu(z)} = \text{Relu}(\Re(\text{z})) + i\text{Relu}(\Im(\text{z})), \quad (2)$$

$$\mathbb{C}\text{Pooling(z)} = \text{Pooling}(\Re(\text{z})) + i\text{Pooling}(\Im(\text{z})). \quad (3)$$

We standardize the complex data to the standard normal complex distribution by scaling the data with the square root of their variances. Specifically, we multiply the 0-centered data $(x - E[x])$ by the inverse square root of the 2×2 covariance matrix V as

$$\hat{x} = (V)^{-\frac{1}{2}}(x - E[x]), \quad (4)$$

Similar to the real-valued batch normalization algorithm, β and γ are used in complex-valued batch normalization. The complex batch normalization is defined as

$$\mathbb{C}\text{BN}(\hat{\text{x}}) = \gamma\,\hat{\text{x}} + \beta. \quad (5)$$

The chain rule for complex-valued neural networks is also used in the back-propagation process. Let L be a real-valued loss function and z be a complex variable such that $z = a + ib$ where $a, b \in \mathbb{R}^D$. Then

$$\begin{aligned}\nabla_L(z) &= \frac{\partial L}{\partial z} = \frac{\partial L}{\partial a} + i\frac{\partial L}{\partial b} = \frac{\partial L}{\partial \Re(z)} + \frac{\partial L}{\partial \Im(z)} \\ &= \Re(\nabla_L(z)) + i\Im(\nabla_L(z)).\end{aligned} \quad (6)$$

To make a fair comparison in the experiments, the feature extractor f_θ follows the same architecture as in the latest works (Finn et al., 2017; Snell et al., 2017), which consists of four convolutional blocks (see Fig. 4). Each block begins with a 2D complex-valued convolutional layer with a 3×3 kernel and 64 filters and also includes a complex batch-normalization layer, a complex Relu nonlinearity, and a 2×2 average pooling layer.

3.2.2 Complex-Valued Metric

Different from other methods for few-shot learning, the extracted complex-valued features $x_i = f_\theta(\mathcal{I}_i) = \Re(x_i) + i\Im(x_i)$, $\Re(x_i), \Im(x_i) \in \mathbb{R}^D$, in CMM have both amplitude and phase information, where D is the number of feature dimensions. We design a unique metric learning method to measure relationships between samples. Our complex metric module contains two parts: complex-valued parameter generation and sample relation metric.

To use the amplitude and phase information in the feature embedding, we choose a commonly used Gaussian similarity function based on a learnable complex-valued network g_ϕ to produce a sample-wise length-scale parameter σ_i,

$$\sigma_i = g_\phi(f_\theta(\mathcal{I}_i)) = g_\phi(\Re(x_i) + i\Im(x_i)), \quad (7)$$

where $\sigma_i = \Re(\sigma_i) + i\Im(\sigma_i)$ and σ_i is generated by the amplitude and phase information of feature embedding. The detailed architecture is illustrated in Fig. 4. Then, our relationship matrix is defined below as

$$A_{i,j} = \exp\left(-\frac{1}{2}d\left(\mathcal{M}\left(\frac{x_i}{\sigma_i}\right), \mathcal{M}\left(\frac{x_j}{\sigma_j}\right)\right)\right), \quad (8)$$

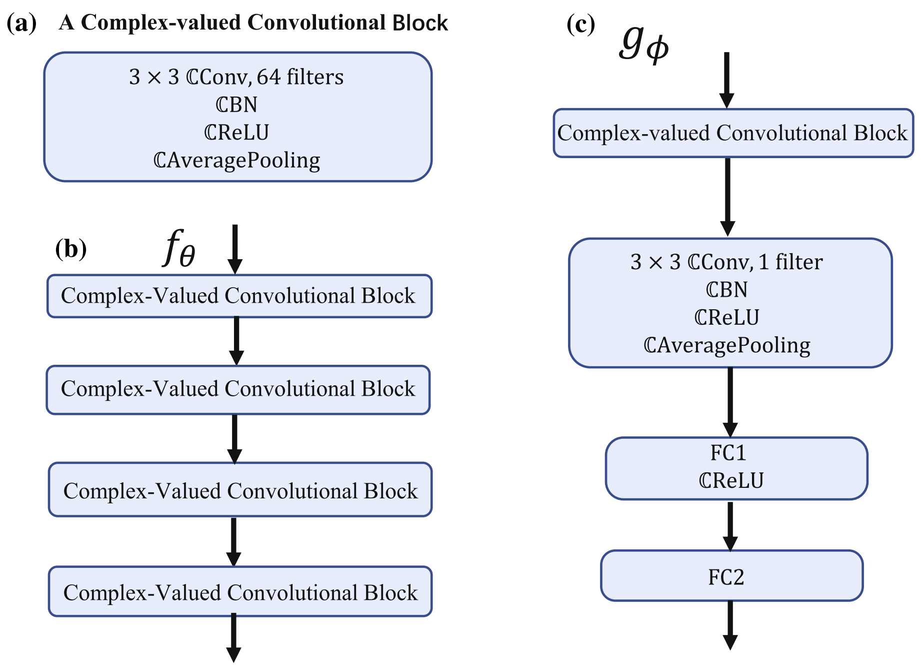

(a) A Complex-valued Convolutional Block
3 × 3 ℂConv, 64 filters
ℂBN
ℂReLU
ℂAveragePooling
(b) f_θ
Complex-Valued Convolutional Block
Complex-Valued Convolutional Block
Complex-Valued Convolutional Block
Complex-Valued Convolutional Block
(c) g_ϕ
Complex-valued Convolutional Block
3 × 3 ℂConv, 1 filter
ℂBN
ℂReLU
ℂAveragePooling
FC1
ℂReLU
FC2

Fig. 4 The detailed architecture of CMM. **a** The detailed architecture of a complex-valued convolutional block. **b** The detailed architecture of the feature extractor f_θ. **c** The detailed architecture of the network g_ϕ

where d denotes the distance function, and $\mathcal{M}$ means the $\mathcal{L}_2$ norm. We can also use the real and imaginary parts of x_i and σ_i to define the operation as

$$\frac{x_i}{\sigma_i} = \frac{\Re(x_i) + i\Im(x_i)}{\Re(\sigma_i) + i\Im(\sigma_i)} = \frac{\Re(x_i)\Re(\sigma_i) + \Im(x_i)\Im(\sigma_i) + i(\Re(\sigma_i)\Im(x_i) - \Re(x_i)\Im(\sigma_i))}{\Re^2(\sigma_i) + \Im^2(\sigma_i)}. \tag{9}$$

Then, we normalize A, and the overall sample relationship is defined as

$$R = D_A^{-1/2} A D_A^{-1/2}, \tag{10}$$

where D_A is the diagonal matrices with the (i, i)-value to be the sum of the i-th row of A. We test our method based on the transductive inference (Liu et al., 2019). $A, R \in \mathbb{R}^{(C\times(K+Q))\times(C\times(K+Q))}$ denote all support and query samples. We only keep the k-max values in each row of R to reduce the noise. We empirically set $k = K + Q + C$ that guarantees that the model can learn label information of K support samples and Q query samples of the same class.

3.3 Transductive Method

In the transductive inference process, we learn the query sample labels from the support sample labels. Different from the transductive inference of Liu et al. (2019), we learn the labels with both support and query samples. In this way, we can learn more accurate inter- and intra-class relationships. Let $R \in \mathbb{R}^{(C\times(K+Q))\times(C\times(K+Q))}$ denote the learned relation matrix whose (i, j)-value is the relationship between the ith sample and the jth sample. Define an initial relation matrix $\mathfrak{I}$ as

$$\mathfrak{I}_{i,j} = \begin{cases} \mathbb{I}(y_i == y_j), & \text{if } x_i, x_j \in S_S, \\ 1/C, & \text{if } x_i, x_j \in S_Q, \\ 0, & \text{otherwise,} \end{cases} \tag{11}$$

where $\mathbb{I}$ is the indicator function. Starting from the initial matrix $\mathfrak{I}$ defined in (11), we iteratively learn the query samples labels from the union set $S_S \cup S_Q$ as

$$Y_{t+1} = (1 - \alpha)RY_t + \alpha\mathfrak{I}, \tag{12}$$

where Y_t denotes the predicted labels at t, R denotes the normalized relation matrix, and α controls the amount of the learned information. Also, it is well known that the sequence $\{Y_t\}$ has a closed-form solution as

$$Y^* = (I - \alpha R)^{-1}\mathfrak{I}, \tag{13}$$

where Y^* denotes the last predicted relationship between samples.

3.4 Label Prediction and Loss Generation

After computing the last learned relation matrix Y^*, we can directly convert the relation matrix Y^* to label scores using softmax as

$$P_{QS}(\tilde{y}_i = j|\mathcal{I}_i) = \frac{\exp(\sum_{z=1}^{K} Y^*_{i,z+K(j-1)})}{\sum_{l=1}^{C}\exp(\sum_{z=1}^{K} Y^*_{i,z+K(l-1)})}, \tag{14}$$

where $\tilde{y}_i$ denotes the final predicted label for the ith sample of the query set. And then, we compute classification loss between the predictions of the query set and the ground-truth labels of the support and query sets union to update all parameters end-to-end. First, we split the classification loss into two parts: query-support (QS) and query-query (QQ) classification losses. Experiments in Sect. 4 show that considering the relation among query samples can make the model learn a better relationship and perform better. The QS classification loss is defined as

$$\mathcal{L}_{QS} = \sum_{i=CK+1}^{C(K+Q)} \sum_{j=1}^{C} -\mathbb{I}(y_i == j) log(P_{QS}(\tilde{y}_i = j|\mathcal{I}_i)), \tag{15}$$

where y_i is the ground-truth label of $\mathcal{I}_i$. Similar to the QS classification loss, the QQ loss is defined as

$$P_{QQ}(\tilde{y}_i = j|\mathcal{I}_i) = \frac{\exp(\sum_{z=1}^{Q} Y^*_{i,NK+z+Q(j-1)})}{\sum_{l=1}^{C} \exp(\sum_{z=1}^{Q} Y^*_{i,NK+z+Q(l-1)})}, \tag{16}$$

$$\mathcal{L}_{QQ} = \sum_{i=CK+1}^{C(K+Q)} \sum_{j=1}^{C} -\mathbb{I}(y_i == j) log(P_{QQ}(\tilde{y}_i = j|\mathcal{I}_i)). \tag{17}$$

Then, the overall loss is the sum of the QS and QQ losses as

$$\mathcal{L}_{CMM} = \mathcal{L}_{QS} + \mathcal{L}_{QQ}. \tag{18}$$

Note that P_{QQ} is only used during the training process to learn a better inter- or intra-class relationship, while P_{QS} will be used both in the test and training process.

4 Dependable Learning

4.1 Background

Traditional adversarial training algorithms indiscriminately add the adversarial perturbation on the whole input image, which deteriorates the feature representation capacity for clean or natural images (Dong et al., 2020). However, we find that adding the adversarial perturbation to the background can strengthen the intra-class feature extraction capability, especially for few-shot learning. We lead a new adversarial training method to improve the intra-class modeling ability by segmenting the background part from the input image and then attacking it, in which the model associates the clean samples with the adversarial samples to diversify the limited samples. The technique can achieve a better feature representation for few-shot learning by introducing a new attention method to locate the foreground part and background part robustly. In particular, attention can be interpreted as a means of biasing the allocation of available computational resources towards the most informative components of a signal (Olshausen et al., 1993; Itti et al., 1998; Itti & Koch, 2001; Larochelle & Hinton, 2010; Mnih et al., 2014; Vaswani et al., 2017). Attention mechanisms have demonstrated their utility across many tasks, including localization, understanding in images (Cao et al., 2015; Jaderberg et al., 2015) and image segmentation (Rassadin, 2020). This work introduces a dependable attention method to effectively segment the foreground and background by considering the linear (bilinear) relationship between the attention map and the perturbation. We fully consider their interaction, and lead a highly robust attention map for few-shot learning. The process is shown in Algorithm 1 and we explain its optimization and the working pipeline in detail.

```
Algorithm 1: Dependable Learning
Input: Training data, validation data, hyper-parameters $\xi_1 = 0.1$,
    $\xi_2 = 0.1$, K=0 and S=65;
Create model weights $\mathbf{W}$ and spatial attention $\mathbf{M^{sp}}$
Output: The network model with dependable attention $\mathbf{M^d}$;
Training an architecture for epochs;
while ($K \leq S$) do
    Inference,
    According to Fig. 5, update the spatial attention $\mathbf{M^{sp}}$,
    According to Eq. 29, backtracking $\mathbf{M^{sp}}$,
    Return $\mathbf{M^d}$,
    Add perturbation to the data via Eq. 19,
    Use Eq. 20 to calculate the object function,
    Back propagation,
    Update weights $\mathbf{W}$,
    $K \leftarrow K + 1$.
end
```

4.2 Selective Attack

We start by explaining the existence of adversarial examples. The typical adversarial input is $\mathbf{x}' = \mathbf{x} + \boldsymbol{\eta}$, where $\boldsymbol{\eta}$ is the perturbation or attack of the input data. We focus on $\boldsymbol{\eta}$ that made up of two parts as

$$\boldsymbol{\eta} = k(\mathbf{M}) \circ \boldsymbol{\delta}, \tag{19}$$

where $\mathbf{M} \in \mathbb{R}^{M \times M}$ is an attention map and $\circ$ denotes the Hadamard production. The kernel function $k(\cdot)$ is the selective mask, which is designed for a selected attack, e.g., selecting background $k(\mathbf{M})$= $(1 - \mathbf{M} \circ \mathbf{M})$. Noted that different from random-erasing (Zhong et al., 2020), our approach is to attack the background part of the image guided by the attention map. The images of the objects in different backgrounds are simulated by associating clean images and

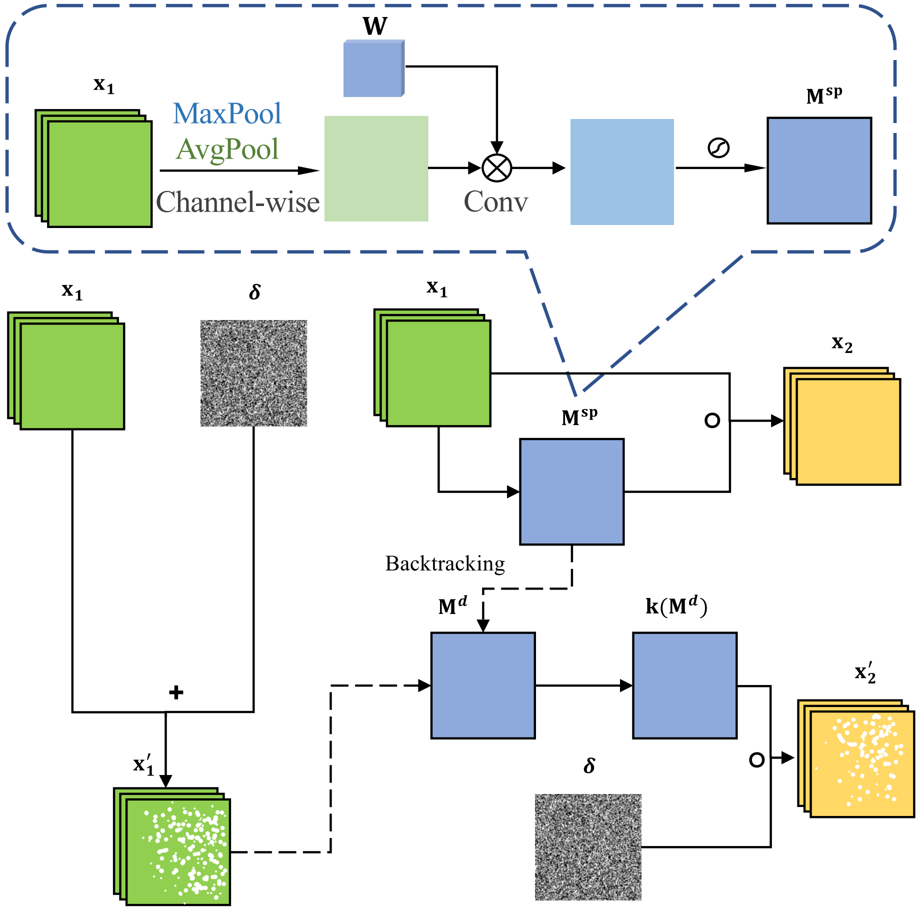


Fig. 5 The flow chart of Dependable Learning. $\mathbf{M^s p}$ denotes the spatial attention obtained from the input $\mathbf{x}_1$. $\mathbf{x}'_1$ denotes the adversarial input for local attack via $\mathbf{M^s p}$, $\mathbf{M^d}$ denotes the dependable attention that $\mathbf{M^s p}$ is backtracked by perturbation $\boldsymbol{\delta}$. $\mathbf{x}_2$, $\mathbf{x}'_2$ denote the output with and without perturbation, respectively. $k(\mathbf{M^d}) = (1 - \mathbf{M^d} \circ \mathbf{M^d})$,which guides the background attack

adversarial images, which help the model to better extract features.

Since the attention inherits from the input data, if there is a perturbation in data, attention's role in locating the critical areas of the image will be affected. In other words, the attention and perturbation are coupled. As a result, we formulate attention and perturbation as a bilinear model better against perturbation. Bilinear optimization models are cornerstones of many computer vision algorithms. The optimized objectives or models are often influenced by two or more hidden factors that interact to produce the observations (Heide et al., 2015; Yang et al., 2017). A fundamental bilinear optimization problem (Mairal et al., 2010) attempts to optimize the following objective function as

$$\arg\min_{\mathbf{M}} G(f_{\{W,\mathbf{M},\boldsymbol{\delta}\}}(\mathbf{x}), y), \tag{20}$$

where $f_{\{W,\mathbf{M},\boldsymbol{\delta}\}}$ is the CNN model with three sets of parameters $\{W, \mathbf{M}, \boldsymbol{\delta}\}$, $\mathbf{M} \in \mathbb{R}^{M\times M}$ and $\boldsymbol{\delta} \in \mathbb{R}^{M\times M}$ are bilinear variables to be optimized. G is the cross-entropy loss function, and $\mathbf{x}$ and y represent the input image and its label respectively. $\boldsymbol{\delta}$ is an improved attack method based on Goodfellow et al. (2015) as

$$\boldsymbol{\delta} = \boldsymbol{\epsilon} \cdot \mathbf{sign}(-\nabla_x \mathcal{L}(f_W(\mathbf{x}_i), y_i)), \tag{21}$$

where $\boldsymbol{\epsilon} \in \mathbb{R}^{M\times M}$, the elements $\epsilon_{i,j} \in [-\epsilon, +\epsilon]$, ϵ is a constant which will be described in the experimental section, $\mathbf{x}_i$ denotes the i-th input and y_i denotes the i-th label. By adding an imperceptibly small vector whose elements are equal to the sign of the gradient of the negative cost function concerning the input, we can change the image's classification and add more examples to sufficiently train the network. Adversarial learning is implemented to simply and effectively improve the robustness and generalization of the model. In adversarial training, we need to add perturbation to increase the robustness of the training, but we do not want perturbation to affect the model's learning of critical areas in the data. To this end, we introduce the dependable attention, which is robust to the perturbation and elaborated below.

4.3 Dependable Attention

We propose the dependable attention $\mathbf{M^d}$ which is an improvement of spatial attention $\mathbf{M^s p}$ (Woo et al., 2018). Figure 5 shows that dependable attention is calculated in two steps. First, traditional spatial attention is generated from noiseless inputs. The dependable attention is then achieved by backtracking the attention suffering from a strong perturbation. The detailed derivation is shown below.

We calculate the dependable attention from a new perspective such that $\mathbf{M}^{\mathbf{s}}\mathbf{p}$ and $\boldsymbol{\delta}$ are coupled. Based on the optimization objective defined in Eq. 20, the chain rule (Petersen et al., 2008) and its notations, we have

$$\mathbf{M}^{\mathbf{d}} = \mathbf{M}^{\mathbf{s}}\mathbf{p} - \xi_1 \frac{\partial G(\mathbf{M}^{\mathbf{sp}})}{\partial \mathbf{M}^{\mathbf{sp}}} - \xi_2 Tr\left(\left(\frac{\partial G(\boldsymbol{\delta})}{\partial \boldsymbol{\delta}}\right)^T \frac{\partial \boldsymbol{\delta}}{\partial \mathbf{M}^{\mathbf{sp}}}\right), \tag{22}$$

where ξ_1 denotes the learning rate of $\mathbf{M}^{\mathbf{d}}$, ξ_2 denotes the backtracking rate of $\mathbf{M}^{\mathbf{d}}$, $Tr(\cdot)$ represents the trace of the matrix, which means that each element in the matrix $\frac{\partial G}{\partial \mathbf{M}^{\mathbf{sp}}}$ adds the trace of the corresponding matrix related to $\mathbf{M}^{\mathbf{sp}}$. We further define

$$\hat{G}(\boldsymbol{\delta}, \mathbf{M}^{\mathbf{sp}}) = \left(\frac{\partial G(\boldsymbol{\delta})}{\partial \boldsymbol{\delta}}\right)^T / \mathbf{M}^{\mathbf{sp}}, \tag{23}$$

where $\hat{G}$ is defined by considering the bilinear optimization problem (Zhuo et al., 2020) as in Eq. 20. Then we have

$$\frac{\partial G(\boldsymbol{\delta})}{\partial \mathbf{M}^{\mathbf{sp}}} = Tr\left[\mathbf{M}^{\mathbf{sp}} \hat{G} \frac{\partial \boldsymbol{\delta}}{\partial \mathbf{M}^{\mathbf{sp}}}\right]. \tag{24}$$

We denote $\hat{G} = [\hat{g}_1, \ldots, \hat{g}_M]$. Assuming that $\boldsymbol{\delta}_m$ and $\mathbf{M}^{\mathbf{s}}\mathbf{p}_n$ are independent when $m \neq n$, where $\boldsymbol{\delta}_m$ and $\mathbf{M}^{\mathbf{s}}\mathbf{p}_n$ are column vectors, we have

$$\frac{\partial \boldsymbol{\delta}_m}{\partial \mathbf{M}^{\mathbf{s}}\mathbf{p}} = \begin{bmatrix} 0 \ldots & \frac{\partial \boldsymbol{\delta}_m}{\partial \mathbf{M}^{\mathbf{s}}\mathbf{p}_{1,m}} & \ldots 0 \\ \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot \\ 0 \ldots & \frac{\partial \boldsymbol{\delta}_m}{\partial \mathbf{M}^{\mathbf{s}}\mathbf{p}_{M,m}} & \ldots 0 \end{bmatrix}, \tag{25}$$

and

$$\mathbf{M}^{\mathbf{s}}\mathbf{p}\hat{G} = \begin{bmatrix} \mathbf{M}^{\mathbf{s}}\mathbf{p}_1 \hat{g}_1 & \ldots & \mathbf{M}^{\mathbf{s}}\mathbf{p}_1 \hat{g}_n & \ldots & \mathbf{M}^{\mathbf{s}}\mathbf{p}_1 \hat{g}_M \\ \cdot & & \cdot & & \cdot \\ \cdot & & \cdot & & \cdot \\ \cdot & & \cdot & & \cdot \\ \mathbf{M}^{\mathbf{s}}\mathbf{p}_M \hat{g}_1 & \ldots & \mathbf{M}^{\mathbf{s}}\mathbf{p}_M \hat{g}_n & \ldots & \mathbf{M}^{\mathbf{s}}\mathbf{p}_M \hat{g}_M \end{bmatrix}. \tag{26}$$

We combine Eq. 25 and Eq. 26, and get

$$\mathbf{M}^{\mathbf{s}}\mathbf{p}\hat{G}\frac{\partial \boldsymbol{\delta}_m}{\partial \mathbf{M}^{\mathbf{s}}\mathbf{p}} = \begin{bmatrix} 0 \ldots & \mathbf{M}^{\mathbf{s}}\mathbf{p}_1 \sum_{n=1}^{M} \hat{g}_n \frac{\partial \boldsymbol{\delta}_m}{\partial \mathbf{M}^{\mathbf{s}}\mathbf{p}_{n,m}} & \ldots 0 \\ \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot \\ 0 \ldots & \mathbf{M}^{\mathbf{s}}\mathbf{p}_M \sum_{n=1}^{M} \hat{g}_n \frac{\partial \boldsymbol{\delta}_m}{\partial \mathbf{M}^{\mathbf{s}}\mathbf{p}_{n,m}} & \ldots 0 \end{bmatrix} \tag{27}$$

After that, the trace of Eq. 22 is then calculated by

$$Tr[\mathbf{M}^{\mathbf{s}}\mathbf{p}\hat{G}\frac{\partial \boldsymbol{\delta}}{\partial \mathbf{M}^{\mathbf{s}}\mathbf{p}_m}] = \mathbf{M}^{\mathbf{s}}\mathbf{p}_m \sum_{l=1}^{M} \hat{g}_l \frac{\partial \boldsymbol{\delta}_m}{\partial \mathbf{M}^{\mathbf{s}}\mathbf{p}_{l,m}}. \tag{28}$$

Defining $\hat{\mathbf{M}}^{\mathbf{sp}} = \mathbf{M}^{\mathbf{s}}\mathbf{p} - \xi_1 \frac{\partial G(\boldsymbol{\delta}, \mathbf{M}^{\mathbf{s}}\mathbf{p})}{\partial \mathbf{M}^{\mathbf{s}}\mathbf{p}}$, dependable attention is calculated by combining Eq. 22 and Eq. 28 as

$$\begin{aligned} \mathbf{M}^{\mathbf{d}} &= \hat{\mathbf{M}}^{\mathbf{sp}} - \xi_2 \begin{bmatrix} \sum_{n=1}^{M} \hat{g}_n \frac{\partial \boldsymbol{\delta}_1}{\partial \mathbf{M}^{\mathbf{s}}\mathbf{p}_{n,1}} \\ \cdot \\ \cdot \\ \cdot \\ \sum_{n=1}^{M} \hat{g}_n \frac{\partial \boldsymbol{\delta}_M}{\partial \mathbf{M}^{\mathbf{s}}\mathbf{p}_{n,M}} \end{bmatrix} \odot \begin{bmatrix} \mathbf{M}^{\mathbf{s}}\mathbf{p}_1 \\ \cdot \\ \cdot \\ \cdot \\ \mathbf{M}^{\mathbf{s}}\mathbf{p}_M \end{bmatrix} \\ &= \hat{\mathbf{M}}^{\mathbf{sp}} - \xi_2 \begin{bmatrix} < \hat{G}, \frac{\partial \boldsymbol{\delta}_1}{\partial \mathbf{M}^{\mathbf{s}}\mathbf{p}_1} > \\ \cdot \\ \cdot \\ \cdot \\ < \hat{G}, \frac{\partial \boldsymbol{\delta}_M}{\partial \mathbf{M}^{\mathbf{s}}\mathbf{p}_M} > \end{bmatrix} \odot \begin{bmatrix} \mathbf{M}^{\mathbf{s}}\mathbf{p}_1 \\ \cdot \\ \cdot \\ \cdot \\ \mathbf{M}^{\mathbf{s}}\mathbf{p}_M \end{bmatrix} \\ &= \hat{\mathbf{M}}^{\mathbf{sp}} - \xi_2 \odot \mathbf{M}^{\mathbf{s}}\mathbf{p} \\ &= P(\hat{\mathbf{M}}^{\mathbf{s}}\mathbf{p}, \mathbf{M}^{\mathbf{s}}\mathbf{p}), \end{aligned} \tag{29}$$

where $\odot$ represents the Hadamard product, $\xi_2\gamma$ denotes the step size of backtracking. To simplify the calculation, $\frac{\partial \boldsymbol{\delta}}{\partial \mathbf{M}^{\mathbf{sp}}}$ can be approximated by $\frac{\Delta \boldsymbol{\delta}}{\Delta \mathbf{M}^{\mathbf{sp}}}$. Equation 29 shows our method is actually based on a projection function for gradient update. In this method, we build our dependable attention $\mathbf{M}^{\mathbf{d}}$ based on $\mathbf{M}^{\mathbf{s}}\mathbf{p}$ and $\hat{\mathbf{M}}^{sp}$. The procedure is finally summarized in Algorithm 1

In this method, we consider the coupling information in $\mathbf{M}^{\mathbf{sp}}$ to backtrack $\hat{\mathbf{M}}^{\mathbf{s}}\mathbf{p}$ and finally obtain the dependable attention $\mathbf{M}^{d}$. To better resist the perturbation, $\mathbf{M}^{d}$ needs backtracking for the attention on the severely perturbed areas in the updating process. To this end, we introduce the triggering condition for backtracking, defined as

$$\mathbf{M}^{\mathbf{d}}_{m,n} = \begin{cases} P(\hat{\mathbf{M}}^{\mathbf{sp}}_{m,n}, \mathbf{M}^{\mathbf{s}}\mathbf{p}_{m,n}) & if\ R(\frac{\partial G}{\partial \boldsymbol{\delta}_{m,n}}) > \zeta, \\ \hat{\mathbf{M}}^{\mathbf{s}}\mathbf{p}_{m,n} & otherwise, \end{cases} \tag{30}$$

where $\{\cdot\}_{m,n}$ denotes the element at row m column n, $P(\cdot)$ denotes the projection function as shown in Eq. 29, $R(\frac{\partial G}{\partial \boldsymbol{\delta}_{m,n}})$ denotes the ranking of $\frac{\partial G}{\partial \boldsymbol{\delta}_{m,n}}$ and ζ represents the threshold.

We define $\gamma = c(\hat{G}, k(\boldsymbol{\delta}_M, \mathbf{M}^{\mathbf{s}}\mathbf{p}_M))$, which implies the coupling relationship between $\boldsymbol{\delta}_M$ and $\mathbf{M}^{\mathbf{s}}\mathbf{p}_M$. Assuming that the model convergence can be seen as a bounded stability, where $\mathbf{M}^{\mathbf{d}}$ is stable, and $c(k, \hat{G}) = \gamma \approx 0$. The coupling information in γ becomes independent from the training loss. That means that the coupling relationship between attention and perturbation is independent of the training process, according to Amar et al. (2021), which can enhance the causal learnability and improve the system robustness based on our bilinear optimization method.

5 Experiments

We evaluate the effectiveness of our proposed CMM and Dependable Learning by comparing other state-of-the-art approaches on two datasets, miniImageNet, and tieredImageNet.

5.1 Datasets

5.1.1 MiniImageNet

The miniImageNet (Krizhevsky et al., 2012) is a subset of ImageNet which has 100 classes selected randomly from ImageNet, and each class has 600 images. Following the split proposed by Ravi and Larochelle (2017), the dataset is divided into training, validation, and test sets, with 64, 16, and 20 classes, respectively.

5.1.2 TieredImageNet

The tieredImageNet (Krizhevsky et al., 2012) dataset is a larger subset of ImageNet with 608 classes. Unlike miniImageNet, it has a hierarchical structure of broader categories of high-level nodes in ImageNet. This set of nodes is partitioned into 20, 6, and 8 disjoint sets of training, validation, and testing nodes, and the corresponding classes form the respective meta-sets. Therefore, the training classes have distinct semantical samples from the test classes, making them more challenging and realistic for few-shot learning.

5.2 Results for Metric Learning

5.2.1 Experimental Setting

Following the recent work (Vinyals et al., 2016), we use the same episodic training procedure to update our model parameters. To be specific, during the training process, we randomly select C classes in $\mathcal{D}_{train}$ and K samples in each class as the supporting data, and then select 15 samples from the remaining data of these C classes as the query data. In all experiments, we set α to 0.01, ξ_2 to 0.1, ζ to 0.05, i.e., the top 5% of the maximum perturbation gradient is backtracked, and use a weight decay of 5×10^{-4}. We take Adam (Kingma & Ba, 2014) as the optimizer with an initial learning rate of 10^{-3} which is halved for every 25,000 episodes on both miniImageNet and tieredImageNet. All experiments are done without data augmentation.

5.2.2 Results and Analysis

We compare our model with several state-of-the-art approaches in various settings. As the proposed CMM belongs to the metric learning type, we mainly compare our model with other state-of-the-art metric learning models, including Matching Nets (Vinyals et al., 2016), Prototypical Nets (Snell et al., 2017), Relation Nets (Sung et al., 2018), and Reptile (Nichol et al., 2018). Moreover, we also choose TPN (Liu et al., 2019) and use the simple transductive method named MAML+Transduction designed by Liu et al. (2019), which explicitly utilizes the query set. Experimental results, including the combinations of 5 and 10 ways and 1 and 5 shots, are shown in Tables 1 and 2. Each accuracy is the average of 600 randomly generated episodes from the test set $\mathcal{D}_{test}$ and top results are highlighted. The methods are divided into three groups with three different inference methods. The first is "N" for inference methods without transduction. The second is "Y" for transductive inference methods, where all query samples are simultaneously predicted. And third is "BN" for query batch statistics used to share information among test samples. The attention $\mathbf{M^d}$ is selected as $\mathbf{M}$ in Eq. 19. FGSM (Goodfellow et al., 2015) is selected as the adversarial training method in Dependable Learning, in which the perturbation bound $\epsilon = \frac{8}{255}$.

The experiments show that the proposed CMM achieves state-of-the-art results and outperforms other methods by a large margin. For a 5-way 1-shot on miniImageNet, our model can achieve high accuracy of 56.26% with a significant improvement of 2.51% over the best-compared method TPN. Even in a more realistic scenario of 10-ways, the absolute improvement of CMM can also achieve 2.19% and 2.53% for 1-shot and 3.24% and 2.68% for 5-shot on miniImageNet and tieredImageNet, respectively. Dependable Learning further improves the model performance. In Tables 1 and 2, the performance of various scenarios is improved by 0.16%–3.47%. The results show that CMM and Dependable Learning effectively improve the performance of few-shot learning.

Another observation is that $\mathcal{L}_{QQ}$ can slightly improve the accuracy of our model. In the process of transductive inference, the loss of CMM consists of two parts, $\mathcal{L}_{QS}$ and $\mathcal{L}_{QQ}$. The first part can make our model learn the relationships between support and query samples and predict the labels of query samples in the test. The second part aims to make our model learn better relationships among query samples which can improve the performance of transductive inference. Clearly, our model can perform better with more accurate relationships among samples.

Table 1 Few-shot classification accuracies on miniImageNet

Model	Trans.	Mini 5-way		Mini 10-way	
		1-shot	5-shot	1-shot	5-shot
MAML (Finn et al., 2017)	B	48.70	63.11	31.27	46.92
MAML+Trans. (Finn et al., 2017)	Y	50.83	66.19	31.83	48.23
Prototypical Net (Snell et al., 2017)	N	46.14	65.77	32.88	49.29
Maching Net (Vinyals et al., 2016)	N	43.56	55.31	–	–
Relation Net (Sung et al., 2018)	B	51.38	67.07	34.86	47.94
Reptile (Nichol et al., 2018)	N	47.07	62.74	31.10	44.66
Reptile+BN (Nichol et al., 2018)	B	49.97	65.99	32.00	47.60
TPN (Liu et al., 2019)	Y	53.75	69.43	36.63	52.32
CMM (QS)	Y	56.21	70.53	37.68	55.39
CMM (QS+QQ)	Y	56.26	70.98	38.82	55.56
CMM + dependable learning (QS+QQ)	Y	**59.73**	**72.07**	**39.14**	**55.78**

Each result is an average of 600 test episodes
Bold values indicate the best results in each experiments

Table 2 Few-shot classification accuracies on tieredImageNet

Model	Trans.	Tiered 5-way		Tiered 10-way	
		1-shot	5-shot	1-shot	5-shot
MAML (Finn et al., 2017)	B	51.67	70.30	34.44	53.32
MAML+Trans. (Finn et al., 2017)	Y	53.23	70.83	34.78	54.67
Prototypical Net (Snell et al., 2017)	N	48.58	69.57	37.35	57.83
Maching Net (Vinyals et al., 2016)	N	54.02	70.11	-	-
Relation Net (Sung et al., 2018)	B	54.48	71.31	36.32	58.05
Reptile (Nichol et al., 2018)	N	48.97	66.47	33.67	48.04
Reptile+BN (Nichol et al., 2018)	B	52.36	71.03	35.32	51.98
TPN (Liu et al., 2019)	Y	57.53	72.85	40.93	59.17
CMM (QS)	Y	57.12	72.74	43.30	61.71
CMM (QS+QQ)	Y	58.12	73.46	43.46	61.85
CMM + dependable learning (QS+QQ)	Y	**58.53**	**73.85**	**43.70**	**62.01**

Each result is an average of 600 test episodes
Bold values indicate the best results in each experiments

5.3 Results for Meta-learning

5.3.1 Experimental Setting

We use a ResNet-12 (He et al., 2016) network as our base learner to conduct experiments on miniImageNet, tieredImageNet datasets. We use an SGD optimizer with a momentum of 0.9 in all experiments. For miniImageNet and tieredImageNet datasets, we set the initial learning rate ξ_1 to 0.1, ξ_2 to 0.1, ζ to 0.1, i.e., the top 10% of the maximum perturbation gradient is backtracked, and use a weight decay of 5×10^{-4}. For experiments on miniImageNet datasets, we train for 65 epochs. The learning rate is decayed by a factor of 0.1 after the first 60 epochs. We train for 60 epochs for experiments on the tieredImageNet dataset. The learning rate is decayed by a factor of 0.1 when the epoch is 30, 40, and 50. In this part, we only adopt the complex-valued feature representation module of CMM. The attention $\mathbf{M}$ in Eq. 19 is selected as $\mathbf{M}^{\mathbf{d}}$. We use the FGSM (Goodfellow et al., 2015) with the perturbation bound $\epsilon = \frac{2}{255}$ as the adversarial training method in Dependable Learning.

5.3.2 Results and Analysis

We present our results on two popular benchmark datasets in Tables 3 and 4, demonstrating that our method consistently outperforms state-of-the-art (SOTA) meta-learning methods on both 5-way 1-shot and 5-way 5-shot tasks. Our method outperforms the baseline IER (Rizve et al., 2021) method across all datasets for both 1-shot and 5-shot tasks. We show results on miniImageNet and tieredImageNet (Tables 3, 4), where we consistently improve the SOTA methods by 0.41%–2.44%.

Table 3 Average 5-way few-shot classification accuracy with 95% confidence intervals on miniImageNet dataset

Methods	Backbone	1-shot	5-shot
MAML (Finn et al., 2017)	32-32-32-32	48.70	63.11
Matching Net (Vinyals et al., 2016)	64-64-64-64	43.56	55.31
Proto-Net (Snell et al., 2017)	64-64-64-65	49.42	68.20
Relation Net (Sung et al., 2018)	54-96-128-256	50.44	65.32
R2D2 (Bertinetto et al., 2019)	96-192-384-512	51.20	68.80
SNAIL (Mishra et al., 2018)	ResNet-12	55.71	68.88
AdaResNet (Munkhdalai et al., 2018)	ResNet-12	56.88	71.94
TADAM (Oreshkin et al., 2018)	ResNet-12	58.50	76.70
Shot-Free (Ravichandran et al., 2019)	ResNet-12	59.04	77.64
TEWAV (Qiao et al., 2019)	ResNet-12	60.07	75.90
MTL (Sun et al., 2019)	ResNet-12	61.20	75.50
MetaOptNet (Lee et al., 2019)	ResNet-12	62.64	78.63
Bossting (Gidaris et al., 2019)	WRN-28-10	63.77	80.70
Fine-tuneing (Dhillon et al., 2020)	WRN-28-10	57.73	78.17
LEO-trainval (Rusu et al., 2019)	WRN-28-10	61.76	77.59
Deep DTN (Chen et al., 2020)	ResNet-12	63.45	77.91
AFHN (Li et al., 2020)	ResNet-18	62.38	78.16
AWGIM (Guo & Cheung, 2020)	WRN-28-10	63.12	78.40
DSN-MR (Simon et al., 2020)	ResNet-12	64.60	79.51
MABAS (Kim et al., 2020)	ResNet-12	65.08	82.70
RFS-Simple (Tian et al., 2020)	ResNet-12	62.02	79.64
RFS-Distill (Tian et al., 2020)	ResNet-12	64.82	82.14
IER (Rizve et al., 2021)	ResNet-12	66.82	84.35
CMM	ResNet-12	67.37	84.58
CMM + dependable learning	ResNet-12	**68.45**	**85.02**

Bold values indicate the best results in each experiments

Table 4 Average 5-way few-shot classification accuracy with 95% confidence intervals on tieredImageNet dataset

Methods	Backbone	1-shot	5-shot
MAML (Finn et al., 2017)	32-32-32-32	51.67	70.30
Proto-Net (Snell et al., 2017)	64-64-64-64	53.31	72.69
Relation Net (Sung et al., 2018)	54-96-128-256	54.48	71.32
Shot-Free (Ravichandran et al., 2019)	ResNet-12	63.52	82.59
MetaOptNet (Lee et al., 2019)	ResNet-12	65.99	81.56
Boosting (Gidaris et al., 2019)	WRN-28-10	70.53	84.98
Fine-tuneing (Dhillon et al., 2020)	WRN-28-10	66.58	85.55
LEO-trainval (Rusu et al., 2019)	WRN-28-10	66.33	81.44
AWGIM (Guo & Cheung, 2020)	WRN-28-10	67.69	72.82
DSN-MR (Simon et al., 2020)	ResNet-12	67.39	82.85
RFS-Simple (Tian et al., 2020)	ResNet-12	62.02	79.64
IER (Rizve et al., 2021)	ResNet-12	71.87	86.82
CMM	ResNet-12	72.15	87.01
CMM + dependable learning	ResNet-12	**72.65**	**87.33**

Bold values indicate the best results in each experiments

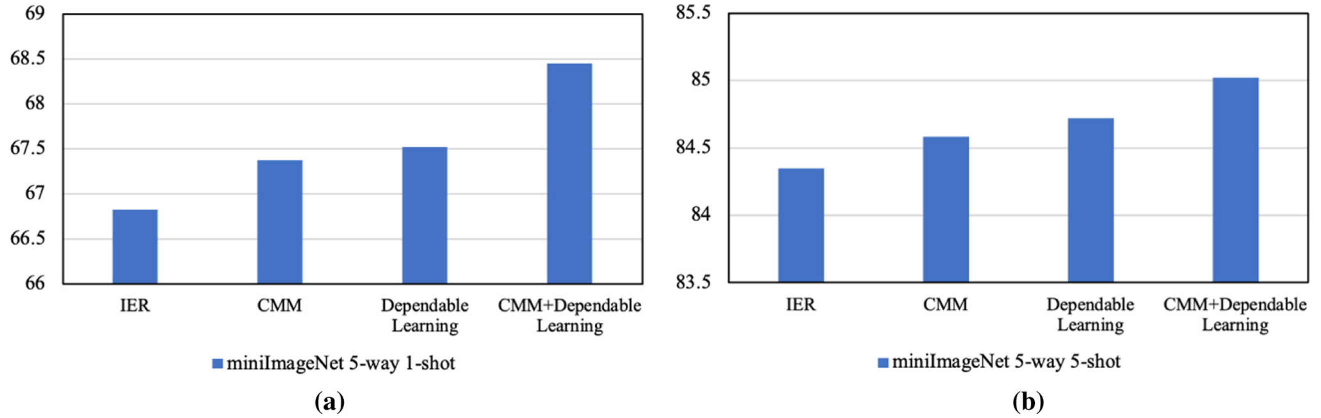


Fig. 6 The comparison of IER, baseline with CMM, baseline mounted with Dependable Learning and baseline mounted with both on miniImageNet dataset for **a** 5-way 1-shot and **b** 5-way 5-shot

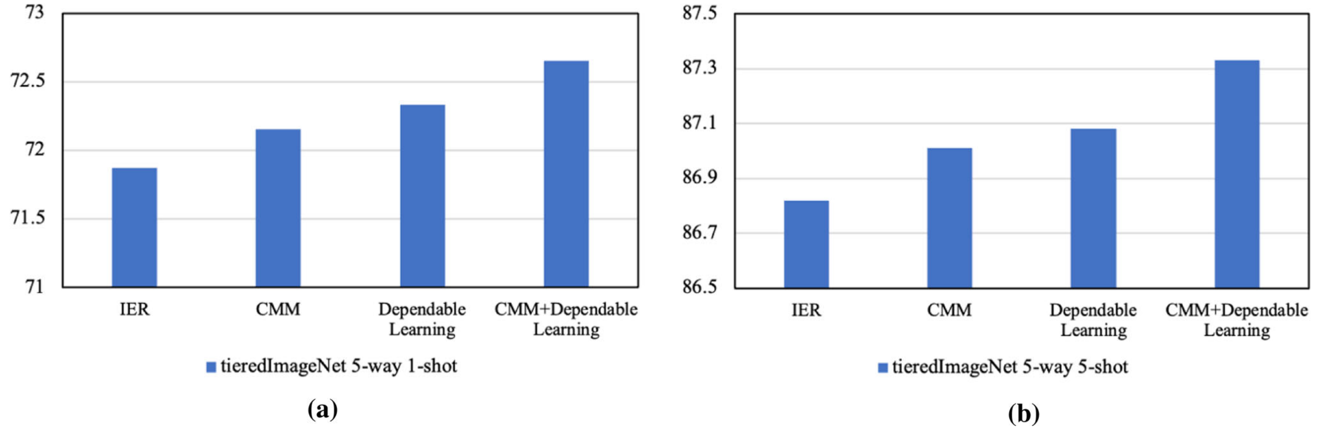


Fig. 7 The comparison of IER, baseline with CMM, baseline mounted with Dependable Learning and baseline mounted with both on tieredImageNet dataset for **a** 5-way 1-shot and **b** 5-way 5-shot

Table 5 Few-shot classification accuracies on miniImageNet with different metric methods

Methods	5-way Acc		10-way Acc	
	1-shot	5-shot	1-shot	5-shot
TPN-64	53.75	69.43	36.62	52.32
TPN-128	54.75	69.79	37.08	53.53
$\mathcal{RI}$	54.99	70.02	37.03	54.43
$\mathcal{AP}$	**56.26**	**70.98**	**38.82**	**55.56**

Each result is an average of 600 test episodes. $\mathcal{RI}$ measures sample distances using the real and imaginary parts of features. $\mathcal{AP}$ measures sample distances using the amplitude part of features
Bold values indicate the best results in each experiments

In Fig. 6, we compare the following: (1) baseline (IER (Rizve et al., 2021)), (2) baseline with CMM, (3) baseline mounted with Dependable Learning, and (4) baseline mounted with CMM and Dependable Learning on the miniImageNet dataset. We can see that both CMM and Dependable Learning can improve the performance with respect to the baseline respectively. Furthermore, the combination of CMM and Dependable Learning leads to better performance. In the case of 5-way 1-shot, the results of the four methods are 66.82%, 67.37%, 67.52%, and 68.45%. In the case of 5-way 5-shot, the results of the four methods are 84.35%, 84.58%, 84.72%, and 85.02%. On tieredImageNet dataset, we get the same conclusion as shown in Fig. 7. In the case of 5-way 1-shot, the results of the four methods are 71.87%, 72.15%, 72.33%, and 72.65%. In the case of 5-way 5-shot, the results of the four methods are 86.82%, 87.01%, 87.08%, and 87.33%.

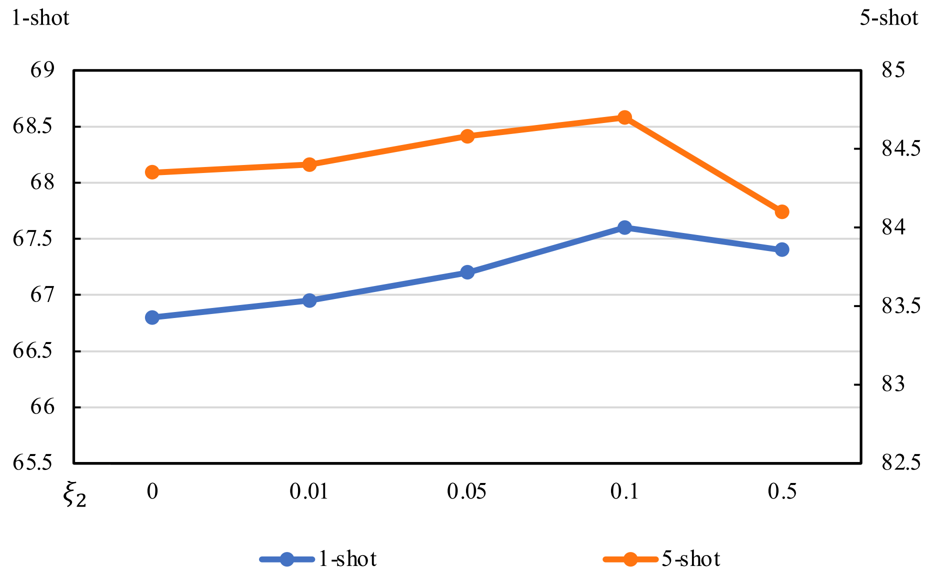


Fig. 8 Dependable attention versus spatial attention ($\xi_2 = 0$). The results of few-shot learning on miniImageNet show that the system performance is improved by backtracking, especially when $\xi_2 = 0.1$

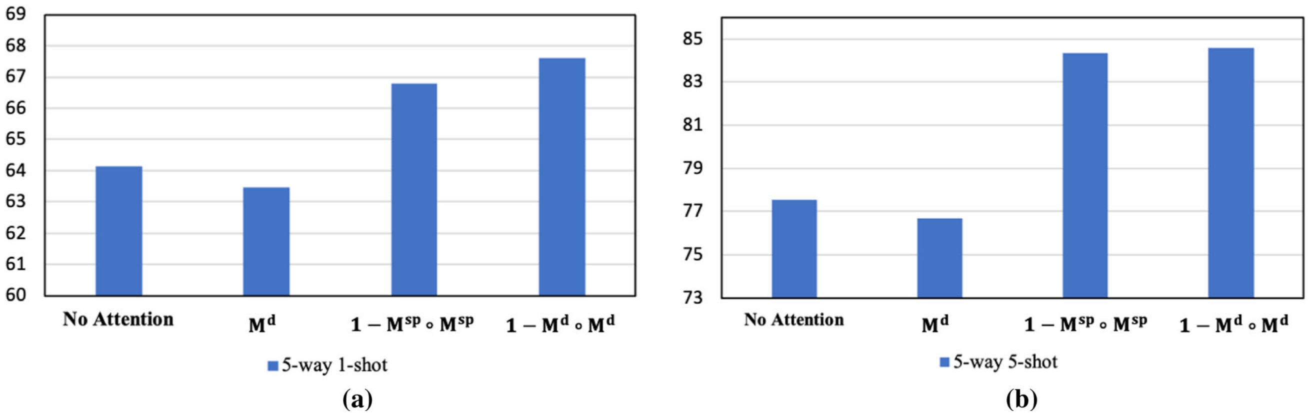


Fig. 9 The impacts of various types of $k(\mathbf{M})$ on the few-shot learning process on miniImageNet for **a** 5-way 1-shot and **b** 5-way 5-shot

5.4 Ablation Experiments

5.4.1 The Complex Metric Unit

$\mathcal{AP}$. To validate the effectiveness of the proposed complex metric unit named $\mathcal{AP}$, we design a commonly used metric method named $\mathcal{RI}$, which measures the real and imaginary parts of sample features separately. Especially, our complex metric unit generates a complex value, which learns from amplitude and phase information of the feature embedding. Different from the complex metric unit $\mathcal{AP}$, $\mathcal{RI}$ measures the Gaussian distance of the real and imaginary parts between samples and then directly sums them up. This method is commonly used to measure sample distances in Sung et al. (2018), Vinyals et al. (2016) and Liu et al. (2019). Furthermore, to eliminate the influence of the number of parameters on the comparison, we design a TPN-128 with the same number of parameters as CMM. The experimental results are listed in Table 5, which shows that the proposed complex metric unit $\mathcal{AP}$ has a better performance than commonly used metric methods. Even with the same number of parameters and fewer filters, CMM still has a higher classification accuracy than TPN.

5.4.2 The Backtracking Rate

ξ_2**.** Besides, we verify the effectiveness of the backtracking operation in Dependable Learning. We perform an ablation study on the meta-learning task with miniImageNet by varying the value of ξ_2, which is based on CMM model, and $k(\mathbf{M}) = (1 - \mathbf{M} \circ \mathbf{M})$. We present the experimental results in Fig. 8, which demonstrates that the performance of our method improves with the increasing ξ_2, but reaches its maximum when $\xi_2 = 0.1$. It should be noted that when ξ_2 is dropped to zero, dependable attention degrades to general spatial attention. Based on results in Fig. 8, we set the value of ξ_2 to 0.1.

5.4.3 The Effects of Different

$k(\mathbf{M})$**.** During the adversarial training process, we apply different ways to make selective attacks. We set the selective mask $k(\mathbf{M})$ as 1, $\mathbf{M^d}$, $(1 - \mathbf{M^{sp}} \circ \mathbf{M^{sp}})$ and $(1 - \mathbf{M^d} \circ \mathbf{M^d})$, and perform experiments respectively. $k(\mathbf{M}) = 1$ (baseline) represents that there is no attention information used to guide the attack. When $k(\mathbf{M}) = \mathbf{M^d}$, the perturbation will be added to the foreground pae. $k(\mathbf{M}) = (1 - \mathbf{M^{sp}} \circ \mathbf{M^{sp}})$ and $k(\mathbf{M}) = (1 - \mathbf{M^d} \circ \mathbf{M^d})$ are background masks. We only use Dependable Learning to conduct 5-way 1-shot task on miniImageNet. The experiment setting is the same as Sect. 5.3. As shown in Fig. 9, the performance varies according to the guided mask $k(\mathbf{M})$. It is worth mentioning that Dependable Learning degrades to the general FGSM attack method when $k(\mathbf{M}) = 1$. Compared with IER in Fig. 6, the model accuracy decreased after FGSM attack. Besides, the effectiveness of foreground attack ($k(\mathbf{M}) = \mathbf{M^d}$) is lower than baseline, which indicates that attack on foreground part is not helpful to few-learning learning. The results achieved by background attacks, i.e., $k(\mathbf{M}) = (1 - \mathbf{M^d} \circ \mathbf{M^d})$ are better than other methods. In our experiments, the attack using $(1 - \mathbf{M^d} \circ \mathbf{M^d})$ performs the best, meaning that $\mathbf{M^d}$ is more accurate and dependable for adversarial samples. It can guide background attacks more effectively.

To verify the impact of different attack methods on Dependable Learning, we use FGSM (Goodfellow et al., 2015), BIM (Kurakin et al., 2016) and PGD (Madry et al., 2018) with $\epsilon = \frac{2}{255}$, step size of $\frac{2}{255}$ and 10 iterative steps on miniImageNet. We adopt an SGD optimizer with a momentum of 0.9 and a weight decay of 5×10^{-4}. In the experiment, we use $k(\mathbf{M}) = (1 - \mathbf{M^d} \circ \mathbf{M^d})$ to carry out a background attack. In Fig. 10, all these attacks are based on Dependable Learning. The extensive experiments prove that Dependable Learning is universal to attack methods, which shows that the performance of few-shot learning can be improved as long as the background part is considered for attack methods, such as FGSM, BIM and PGD.

To further prove the effectiveness of Dependable Learning, we compare Dependable Learning with other available attack methods based on the CMM model. We test the 5-way 1-shot task on the miniImageNet dataset. The perturbation degree $\epsilon = \frac{2}{255}$ and step size are $\frac{2}{255}$ for all attack methods, and 10 iterative steps are for BIM and PGD. As shown in Fig. 11, the performance of CMM+Dependable Learning is better than CMM with other attack methods.

In order to verify the influence of different degrees of perturbations on the model, we use $k(\mathbf{M}) = (1 - \mathbf{M^d} \circ \mathbf{M^d})$ to guide FGSM to attack the background part on miniImageNet dataset. The degrees of perturbations $\epsilon = 0, \frac{1}{255}, \frac{2}{255}, \frac{8}{255}$ and $\frac{16}{255}$. As shown in Fig. 12, with the increase of perturbation, the accuracy of the 5-way 1-shot task continued to increase until $\epsilon = \frac{8}{255}$. The model achieves the best performance at $\epsilon = \frac{2}{255}$ on 5-way 5-shot tasks. The results show that the perturbation helps to disrupt the background distribution and can affect the few-shot learning performance.

Table 6 The comparison of real-valued CNN and complex-valued CNN after training 200 epochs on CIFAR-10-LT dataset

Dataset	CIFAR-10-LT
Imbalance factor	50
Baseline (real-valued ResNet-32)	82.07
Complex-valued counterpart	82.75

5.5 The Discussion on Over-Fitting of Complex-Valued CNN

The Complex-valued CNN has additional degrees of freedom which help in learning a better representation of the metric latent space. However, the increased model capacity also puts forward higher requirements for optimization. To test whether Complex-valued CNN causes over-fitting of the model on an imbalanced dataset, we compared Complex-valued CNN with its real-valued counterpart on the long-tailed versions of the CIFAR-10 dataset. We use an imbalance factor β to describe the severity of the long-tailed problem with the number of training samples for the most frequent class and the least frequent class, e.g., $\beta = \frac{N_{max}}{N_{min}}$. The imbalance factor we use in the experiment is 50. We train the ResNet-32 (He et al., 2016) as our backbone network by SGD with the momentum of 0.9, weight decay of 2×10^{-4}. We train all the models on a single NVIDIA 1080Ti GPU for 200 epochs with batch size of 128. The initial learning rate is set to 0.1 and the first five epochs are trained with the linear warm-up learning rate schedule. The learning rate is decayed at the 120th and 160th epoch by 0.1. The pipeline is the same as those in Zhou et al. (2020). The results of both models on CIFAR-10-LT dataset are shown in Table 6, in which, the accuracy of complex-valued CNN is higher than its real-valued counterpart. The loss curves of the two models in the validation set are shown in Fig. 13. There is no obvious difference between the two curves, and the loss value of validation set in the first ten epochs of complex-valued CNN is higher than the real-valued counterpart, and then the two models tend to be converged. We compared the average value and variance of the loss on validation dataset, and the results are shown in Table 7. The average value of complex-valued CNN loss is slightly higher than its real-valued counterpart. Despite this, the loss of Complex-valued CNN does not show an increasing trend, nor does performance deteriorate. Therefore, complex-valued CNN does not appear more obvious over-fitting compared to its real-valued counterpart.

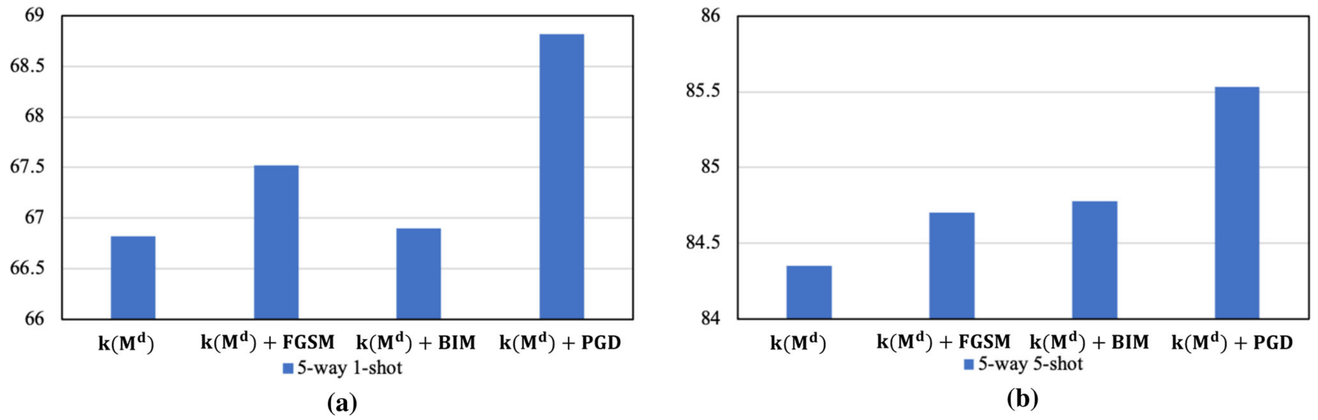


Fig. 10 The comparison of background attack results with different attack methods guided by dependable attention for **a** 5-way 1-shot and **b** 5-way 5-shot

Fig. 11 The comparison of 5-way 1-shot between Dependable Learning based on CMM model and other attack methods on miniImageNet

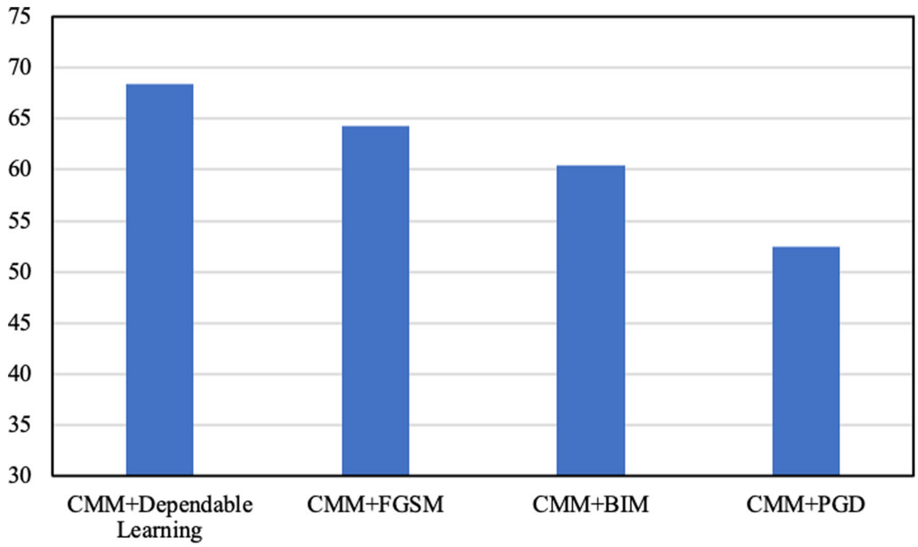
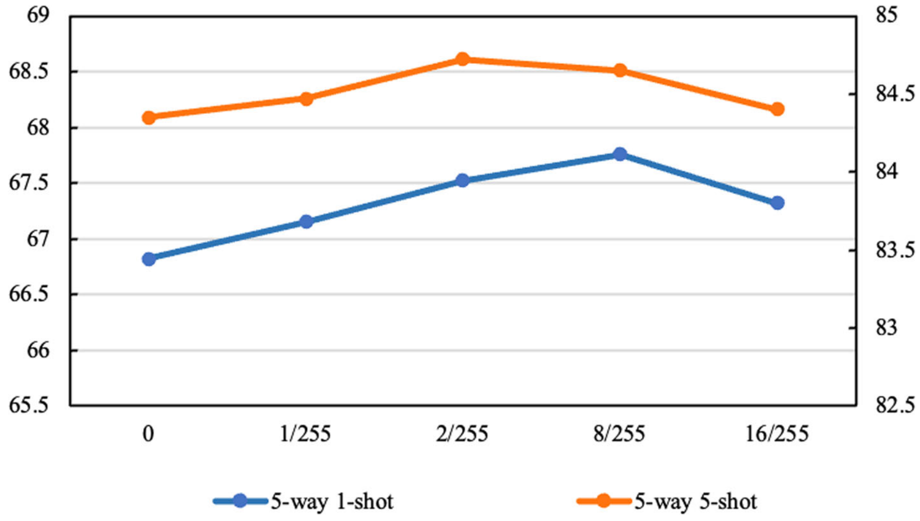


Fig. 12 The impacts of various degrees of perturbations on miniImageNet



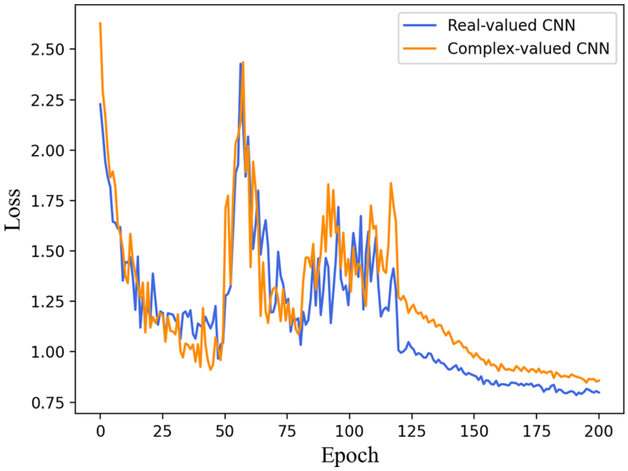


Fig. 13 The comparison of loss curves between complex-valued CNN and real-valued CNN on CIFAR-10-LT validation set

Table 7 The loss average and loss variance of the two models

	Loss average	Loss variance
Real-valued CNN	1.13	0.11
CMM	1.22	0.11

6 Conclusions

We have introduced complex-valued CNNs and Dependable Learning for the few-shot learning in this work. Our Complex Metric Module (CMM) adopts complex-valued CNNs to learn samples' amplitude and phase information, improving the system performance. A Dependable Learning method is further introduced to enhance feature extraction. We add a dependable attention mechanism based on a new parameter update method, "backtracking", which decouples perturbation and attention to calculate the attention robustly. Overall, we improve the robustness of the few-shot learning model and achieve state-of-the-art results on miniImageNet and tieredImageNet. In our future work, we will explore the potential of our method on more applications, such as object detection and segmentation.

Acknowledgements This work was supported by "the Fundamental Research Funds for the Central Universities", and the National Natural Science Foundation of China under Grant 62076016, Beijing Natural Science Foundation-Xiaomi Innovation Joint Fund L223024

References

Amar, D., Sinnott-Armstrong, N., Ashley, E. A., & Rivas, M. A. (2021). Graphical analysis for phenome-wide causal discovery in genotyped population-scale biobanks. *Nature Communications, 12*(1), 1–11.

Arjovsky, M., Shah, A., & Bengio, Y. (2016) Unitary evolution recurrent neural networks. In *ICML* (pp. 1120–1128).

Athalye, A., Carlini, N., & Wagner, D. (2018). Obfuscated gradients give a false sense of security: Circumventing defenses to adversarial examples. In *ICML* (pp. 274–283).

Bertinetto, L., Henriques, J. F., Torr, P. H. S., & Vedaldi, A. (2019). Meta-learning with differentiable closed-form solvers. In *ICLR* (pp. 1–11).

Cao, C., Liu, X., Yang, Y., Yu, Y., Wang, J., Wang, Z., Huang, Y., Wang, L., Huang, C., Xu, W., Ranmanan, D., Huang, T.(2015). Look and think twice: Capturing top-down visual attention with feedback convolutional neural networks. In *ICCV* (pp. 2956–2964).

Carlini, N., & Wagner, D. (2017). Towards evaluating the robustness of neural networks. In *2017 IEEE symposium on security and privacy* (pp. 39–57).

Chen, M., Fang, Y., Wang, X., Luo, H., Geng, Y., Zhang, X., Huang, C., Liu, W., & Wang, B. (2020). Diversity transfer network for few-shot learning. In *AAAI* (pp. 10559–10566).

Cubuk, E. D., Zoph, B., Schoenholz, S. S., & Le, Q. V. (2017). Intriguing properties of adversarial examples. In *ICLR* (pp. 1–17).

Danihelka, I., Wayne, G., Uria, B., Kalchbrenner, N., & Graves, A. (2016). Associative long short-term memory. In *ICML* (pp. 1986–1994).

Das, N., Shanbhogue, M., Chen, S., Hohman, F., Chen, L., Kounavis, M. E., & Chau, D. H. (2017). Keeping the bad guys out: Protecting and vaccinating deep learning with jpeg compression. *arXiv* (pp. 1–15).

Dhillon, G. S., Chaudhari, P., Ravichandran, A., & Soatto, S. (2020). A baseline for few-shot image classification. In *ICLR* (pp. 1–20).

Dong, X., Han, J., Chen, D., Liu, J., Bian, H., Ma, Z., Li, H., Wang, X., Zhang, W., & Yu, N. (2020). Robust superpixel-guided attentional adversarial attack. In *CVPR* (pp. 12895–12904).

Dong, Y., Liao, F., Pang, T., Su, H., Zhu, J., Hu, X., & Li, J. (2018). Boosting adversarial attacks with momentum. In *CVPR* (pp. 9185–9193).

Fehervari, I., Ravichandran, A., & Appalaraju, S. (2019). Unbiased evaluation of deep metric learning algorithms. *arXiv* (pp. 1–9).

Finn, C., Abbeel, P., & Levine, S. (2017). Model-agnostic meta-learning for fast adaptation of deep networks. In *ICML* (pp. 1126–1135).

Gidaris, S., Bursuc, A., Komodakis, N., Pérez, P., & Cord, M. (2019). Boosting few-shot visual learning with self-supervision. In *ICCV* (pp. 8059–8068).

Gidaris, S., & Komodakis, N. (2018). Dynamic few-shot visual learning without forgetting. In *CVPR* (pp. 4367–4375).

Gidaris, S., & Komodakis, N. (2019). Generating classification weights with gnn denoising autoencoders for few-shot learning. In *CVPR* (pp. 21–30).

Goodfellow, I. J., Shlens, J., & Szegedy, C. (2015). Explaining and harnessing adversarial examples. In *ICLR* (pp. 1–11).

Guo, Y., & Cheung, N.-M. (2020). Attentive weights generation for few shot learning via information maximization. In *CVPR* (pp. 13499–13508).

Gupta, P., & Rahtu, E. (2019). Ciidefence: Defeating adversarial attacks by fusing class-specific image inpainting and image denoising. In *ICCV* (pp. 6708–6717).

He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep residual learning for image recognition. In *CVPR* (pp. 770–778).

Heide, F., Heidrich, W., & Wetzstein, G. (2015). Fast and flexible convolutional sparse coding. In *CVPR* (pp. 5135–5143).

Hirose, A., & Yoshida, S. (2012). Generalization characteristics of complex-valued feedforward neural networks in relation to signal coherence. *IEEE Transactions on Neural Networks, 23*(4), 541–551.

Itti, L., & Koch, C. (2001). Computational modelling of visual attention. *Nature Reviews Neuroscience, 2*(3), 194–203.

Itti, L., Koch, C., & Niebur, E. (1998). A model of saliency-based visual attention for rapid scene analysis. *IEEE Transactions on pattern analysis and machine intelligence, 20*(11), 1254–1259.

Jaderberg, M., Simonyan, K., Zisserman, A., Kavukcuoglu, K. (2015). Spatial transformer networks. In *NeurIPS* (pp. 2017–2025).

Kim, J., Kim, H., & Kim, G. (2020). Model-agnostic boundary-adversarial sampling for test-time generalization in few-shot learning. In *ECCV* (pp. 599–617).

Kingma, D. P., & Adam, J. B. (2014). A method for stochastic optimization. In *ICLR*.

Koch, G., Zemel, R., & Salakhutdinov, R. (2015). Siamese neural networks for one-shot image recognition. In *ICML deep learning workshop* (pp. 1–8).

Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2012). Imagenet classification with deep convolutional neural networks. In *NeurIPS* (pp. 1097–1105).

Kurakin, A., Goodfellow, I., & Bengio, S. (2016). Adversarial examples in the physical world. In *ICLR* (pp. 1–13).

Lake, B. M., Salakhutdinov, R., & Tenenbaum, J. B. (2015). Human-level concept learning through probabilistic program induction. *Science, 350*(6266), 1332–1338.

Larochelle, H., & Hinton, G. E. (2010). Learning to combine foveal glimpses with a third-order Boltzmann machine. In *NeurIPS* (pp. 1243–1251).

Lee, K., Maji, S., Ravichandran, A., & Soatto, S. (2019). Meta-learning with differentiable convex optimization. In *CVPR* (pp. 10657–10665).

Li, K., Zhang, Y., Li, K., & Fu, Y. (2020). Adversarial feature hallucination networks for few-shot learning. In *CVPR* (pp. 13470–13479).

Liao, F., Liang, M., Dong, Y., Pang, T., Hu, X., & Zhu, J. (2018). Defense against adversarial attacks using high-level representation guided denoiser. In *CVPR* (pp. 1778–1787).

Li, F. F., & Fergus, R. (2006). One-shot learning of object categories. *IEEE Transactions on Pattern Analysis and Machine Intelligence, 25*(4), 594–611.

Liu, Y., Lee, J., Park, M., Kim, S., Yang, E., Hwang, S., & Yang, Y. (2019). Learning to propagate labels: Transductive propagation network for few-shot learning. In *ICLR* (pp. 1–14).

Liu, Y., Chen, X., Liu, C., & Song, D. (2016). Delving into transferable adversarial examples and black-box attacks. In *ICLR* (pp. 1–24).

Liu, Z., Zhang, B., & Guo, G. (2020). Few-shot learning with complex-valued neural networks. In *BMVC* (pp. 541–552).

Madry, A., Makelov, A., Schmidt, L., Tsipras, D., & Vladu, A. (2018). Towards deep learning models resistant to adversarial attacks. *ICLR* (pp. 1–28).

Mairal, J., Bach, F., Ponce, J., & Sapiro, G. (2010). Online learning for matrix factorization and sparse coding. *JMLR, 11*(Jan), 19–60.

Mishra, N., Rohaninejad, M., Chen, X., & Abbeel, P. (2018). A simple neural attentive meta-learner. In *ICLR*.

Mnih, V., Heess, N., Graves, A., Kavukcuoglu, K. (2014). Recurrent models of visual attention. In *NeurIPS* (pp. 2204–2212).

Mönning, N., & Manandhar, S. (2018). Evaluation of complex-valued neural networks on real-valued classification tasks. In *arXiv* (pp. 1–18).

Munkhdalai, T., Yuan, X., Mehri, S., & Trischler, A. (2018). Rapid adaptation with conditionally shifted neurons. In *International conference on machine learning* (pp. 3664–3673).

Mustafa, A., Khan, S., Hayat, M., Goecke, R., Shen, J., & Shao, L. (2019). Adversarial defense by restricting the hidden space of deep neural networks. In *ICCV* (pp. 3385–3394).

Na, T., Ko, J. H., & Mukhopadhyay, S. (2017). Cascade adversarial machine learning regularized with a unified embedding. In *ICLR* (pp. 1–15).

Nichol, A., Achiam, J., & Schulman, J. (2018). On first-order meta-learning algorithms. In *ICML* (pp. 324–330).

Nitta, T. (2002). On the critical points of the complex-valued neural network. *ICNIP, 3,* 1099–1103.

Olshausen, B. A., Anderson, C. H., Essen, V., & David, C. (1993). A neurobiological model of visual attention and invariant pattern recognition based on dynamic routing of information. *Journal of Neuroscience, 13*(11), 4700–4719.

Oreshkin, B. N., Rodriguez, P., & Lacoste, A. (2018). Tadam: Task dependent adaptive metric for improved few-shot learning. In *NeurIPS* (pp. 719–729).

Petersen, K. B., Pedersen, M. S. (2008). The matrix cookbook. *Technical University of Denmark, 7*(15), 510.

Qiao, L., Shi, Y., Li, J., Wang, Y., Huang, T., & Tian, Y. (2019). Transductive episodic-wise adaptive metric for few-shot learning. In *ICCV* (pp. 3603–3612).

Rassadin, A. (2020) Deep residual 3d u-net for joint segmentation and texture classification of nodules in lung. In *ICIAR* (pp. 419–427).

Ravi, S., & Larochelle, H. (2017). Optimization as a model for few-shot learning. In *ICLR* (pp. 1–11).

Ravichandran, A., Bhotika, R., & Soatto, S. (2019). Few-shot learning with embedded class models and shot-free meta training. In *ICCV* (pp. 331–339).

Reichert, D. P., & Serre, T. (2014). Neuronal synchrony in complex-valued deep networks. In *ICLR* (pp. 1–14).

Rizve, M. N., Khan, S., Khan, F. S., & Shah, M. (2021). Exploring complementary strengths of invariant and equivariant representations for few-shot learning. In *CVPR* (pp. 10836–10846).

Rusu, A. A., Rao, D., Sygnowski, J., Vinyals, O., Pascanu, R., Osindero, S., & Hadsell, R. (2019). Meta-learning with latent embedding optimization. In *ICLR*.

Shafahi, A., Najibi, M., Ghiasi, A., Xu, Z., Dickerson, J., Studer, C., Davis, L. S., Taylor, G., & Goldstein, T. (2019). Adversarial training for free! In *NeurIPS* (pp. 3358–3369).

Simon, C., Koniusz, P., Nock, R., & Harandi, M. (2020). Adaptive subspaces for few-shot learning. In *CVPR* (pp. 4136–4145).

Simonyan, K., & Zisserman, A. (2014). Very deep convolutional networks for large-scale image recognition. In *ICLR* (pp. 1–14).

Snell, J., Swersky, K., & Zemel, R. S. (2017). Prototypical networks for few-shot learning. In *NeurIPS* (pp. 4080–4090).

Sun, Q., Liu, Y., Chua, T.-S., & Schiele, B. (2019). Meta-transfer learning for few-shot learning. In *CVPR* (pp. 403–412).

Sung, F., Yang, Yo., Zhang, L., Xiang, T., Torr, P. H. S., & Hospedales, T. M. (2018). Learning to compare: Relation network for few-shot learning. In *CVPR* (pp. 1199–1208).

Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., Erhan, D., Vanhoucke, V., & Rabinovich, A. (2015). Going deeper with convolutions. In *CVPR* (pp. 1–9).

Szegedy, C., Zaremba, W., Sutskever, I., Bruna, J., Erhan, D., Goodfellow, I., & Fergus, R. (2013). Intriguing properties of neural networks. In *ICLR*.

Tian, Y., Wang, Y., Krishnan, D., Tenenbaum, J. B., & Isola, P. (2020). Rethinking few-shot image classification: a good embedding is all you need? In *ECCV* (pp. 266–282).

Trabelsi, C., Bilaniuk, O., Zhang, Y., Serdyuk, D., Subramanian, S., Santos, J. F., Mehri, S., Rostamzadeh, N., Bengio, Y., & Pal, C. (2018). Deep complex networks. In *ICLR* (pp. 1–19).

Tramèr, F., Kurakin, A., Papernot, N., Goodfellow, I., Boneh, D., & McDaniel, P. (2017). Ensemble adversarial training: Attacks and defenses. In *ICLR* (pp. 1–22).

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., & Polosukhin, I. (2017). Attention is all you need. In *NeurIPS* (pp. 5998–6008).

Vinyals, O., Blundell, C., Lillicrap, T., Kavukcuoglu, K., & Wierstra, D. (2016). Matching networks for one shot learning. In *NeurIPS* (pp. 3630–3638).

Wang, J., & Zhang, H. (2019). Bilateral adversarial training: Towards fast training of more robust models against adversarial attacks. In *ICCV* (pp. 6629–6638).

Wang, R., Bao, Y., Zhang, B., Liu, J., Zhu, W., & Guo, G. (2022). Anti-retroactive interference for lifelong learning. In *arXiv preprint*arXiv:2208.12967

Wang, Y., Yao, Q., Kwok, J. T., & Ni, L. M. (2020). Generalizing from a few examples: A survey on few-shot learning. In *ACM computing surveys* (pp. 1–34).

Woo, S., Park, J., Lee, J.-Y., & Kweon, I. S. (2018). Cbam: Convolutional block attention module. In *ECCV* (pp. 3–19).

Xie, C., Zhang, Z., Zhou, Y., Bai, S., Wang, J., Ren, Z., & Yuille, A. L. (2019). Improving transferability of adversarial examples with input diversity. In *CVPR* (pp. 2730–2739).

Yang, L., Li, C., Han, J., Chen, C., Ye, Q., Zhang, B., Cao, X., Liu, W. (2017). Image reconstruction via manifold constrained convolutional sparse coding for image sets. *JSTSP, 11*(7), 1072–1081.

Ye, S., Xu, K., Liu, S., Cheng, H., Lambrechts, J., Zhang, H., Zhou, A., Ma, K., Wang, Y., & Lin, X. (2019). Adversarial robustness vs. model compression, or both? In *ICCV* (pp. 111–120).

Zeiler, M. D., & Fergus, R. (2014). Visualizing and understanding convolutional networks. In *ECCV* (pp. 818–833).

Zhang, B., Shan, S., Chen, X., & Gao, W. (2006). Histogram of Gabor phase patterns (HGPP): A novel object representation approach for face recognition. *IEEE Transactions on Image Processing, 16*(1), 57–68.

Zhang, R., Che, T., Ghahramani, Z., Bengio, Y., & Song, Y. (2018). Metagan: An adversarial approach to few-shot learning. In *NeurIPS* (pp. 1–8).

Zhang, Z., Wang, H., Xu, F., & Jin, Y. (2017). Complex-valued convolutional neural network and its application in polarimetric SAR image classification. *IEEE Transactions on Geoscience and Remote Sensing, 55*(12), 7177–7188.

Zhong, Z., Zheng, L., Kang, G., Li, S., & Yang, Y. (2020). Random erasing data augmentation. In *AAAI* (pp. 1–8).

Zhou, B., Cui, Q., Wei, X.-S., & Chen, Z.-M. (2020). BBN: Bilateral-branch network with cumulative learning for long-tailed visual recognition. In *CVPR* (pp. 9719–9728).

Zhuo, L., Zhang, B., Yang, L., Chen, H., Ye, Q., Doermann, D., Ji, R., & Guo, G. (2020). Cogradient descent for bilinear optimization. In *CVPR* (pp. 7959–7967).

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.