Knowledge-Based Systems 277 (2023) 110798

Contents lists available at ScienceDirect

Knowledge-Based Systems

journal homepage: www.elsevier.com/locate/knosys

Detach and unite: A simple meta-transfer for few-shot learning

Yaoyue Zheng [a,c,d], Xuetao Zhang [a,c,d], Zhiqiang Tian [b,*], Wei Zeng [e], Shaoyi Du [a,c,d]

[a] Institute of Artificial Intelligence and Robotics, Xi'an Jiaotong University, Xi'an 710049, China
[b] School of Software Engineering, Xi'an Jiaotong University, Xi'an 710049, China
[c] National Key Laboratory of Human–Machine Hybrid Augmented Intelligence, Xi'an Jiaotong University, Xi'an 710049, China
[d] National Engineering Research Center for Visual Information and Applications, Xi'an Jiaotong University, Xi'an 710049, China
[e] School of Physics and Mechanical and Electrical Engineering, Longyan University, Longyan 364012, China

ARTICLE INFO

Article history:
Received 31 March 2023
Received in revised form 19 June 2023
Accepted 10 July 2023
Available online 14 July 2023

Keywords:
Few-shot learning
Meta-learning
Transfer learning
Image classification

ABSTRACT

Few-shot Learning (FSL) is a challenging problem that aims to learn and generalize from limited examples. Recent works have adopted a combination of meta-learning and transfer learning strategies for FSL tasks. These methods perform pre-training and transfer the learned knowledge to meta-learning. However, it remains unclear whether this transfer pattern is appropriate, and the objectives of the two learning strategies have not been explored. In addition, the inference of meta-learning in FSL relies on sample relations that require further consideration. In this paper, we uncover an overlooked discrepancy in learning objectives between pre-training and meta-learning strategies and propose a simple yet effective learning paradigm for the few-shot classification task. Specifically, the proposed method comprises two components: (i) Detach: We formulate an effective learning paradigm, Adaptive Meta-Transfer (A-MET), which adaptively eliminates undesired representations learned by pre-training to address the discrepancy. (ii) Unite: We propose a Global Similarity Compatibility Measure (GSCM) to jointly consider sample correlation at a global level for more consistent predictions. The proposed method is simple to implement without any complex components. Extensive experiments on four public benchmarks demonstrate that our method outperforms other state-of-the-art methods under more challenging scenarios with large domain differences between the base and novel classes and less support information available. Code is available at: https://github.com/yaoyz96/a-met.

1. Introduction

Humans can learn a concept from a few examples and generalize it to a new scenario. For example, a child can easily generalize the concept of "zebra" from a single picture. However, despite achieving impressive results in many visual recognition tasks, deep learning methods have failed to learn like human. Motivated by the learning ability of human, few-shot learning (FSL) [1] has been proposed to mimic this generalization ability and learn new concepts from very few labeled examples, which makes those deep learning methods more challenging and often leads to overfitting. Recently, there have been numerous works exploring the application of few-shot learning in various domains, such as image classification [2], object detection [3], and semantic segmentation [4]. These works aim to enhance the performance of deep learning models in more realistic scenarios.

With a few labeled examples, we can in principle train a classifier to assign a class label to each unlabeled example. But due to the scarcity of labeled data, the classifier usually tends to overfit. One practical approach is to apply transfer learning [5–7] to alleviate this problem. The basic idea is to pre-train a model on sufficient examples (*base classes* $\mathcal{D}_{base}$) and then fine-tune the model on a target task to learn unseen classes (*novel classes* $\mathcal{D}_{novel}$). But in practice, most works have to freeze the feature encoder and only train a new classifier on $\mathcal{D}_{novel}$ due to the data scarcity [8–10]. The transfer learning strategy in FSL is shown in the upper left side of Fig. 1. Some works [6,11] hold that this strategy can attain satisfactory performance in FSL. However, their base classes in pre-training usually have the same domain as novel classes. Recent work BiT [12] proposed that with large-scale labeled data (e.g. ImageNet-1k or ImageNet-22k [13]), the pre-trained model can get a more generalized representation for novel classes. But this means that it requires even much more computing resources.

Learning to learn [14] is a key idea that forms the basis of meta-learning, another prevalent learning strategy for FSL. Meta-learning defines a learning mechanism for FSL, often termed *episodic learning*. Different from standard deep learning, the training/test data of episodic learning is a collection of *tasks* $(\mathcal{S}, \mathcal{Q})$, instead of data itself. In each task, we can randomly select C unique

* Corresponding author.
E-mail addresses: z1037268262@stu.xjtu.edu.cn (Y. Zheng), xuetaozh@xjtu.edu.cn (X. Zhang), zhiqiangtian@xjtu.edu.cn (Z. Tian), zengwei@lyun.edu.cn (W. Zeng), dushaoyi@xjtu.edu.cn (S. Du).

https://doi.org/10.1016/j.knosys.2023.110798
0950-7051/

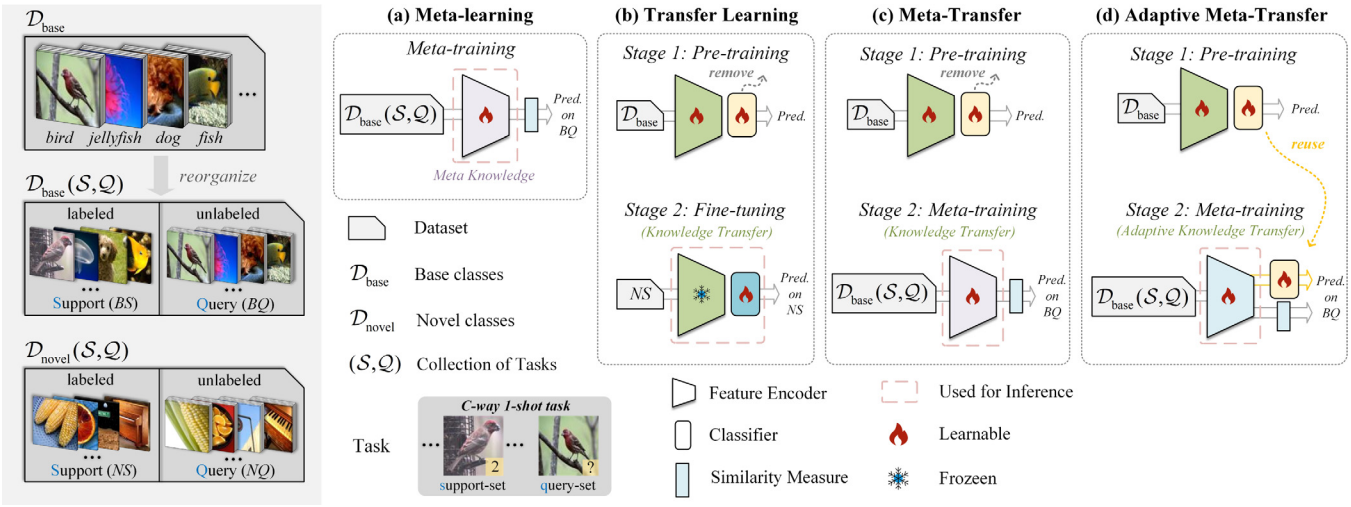

Fig. 1. Overview of the few-shot learning strategies: (a) Meta-learning, the training data is a collection of tasks ($\mathcal{S}$, $\mathcal{Q}$), instead of data itself. The model is optimized to perform well on $\mathcal{D}_{base}(\mathcal{S}, \mathcal{Q})$ to get generalization ability (meta-knowledge) across various task on $\mathcal{D}_{novel}(\mathcal{S}, \mathcal{Q})$. (b) Transfer learning involves pre-training a model on sufficient examples $\mathcal{D}_{base}$. Then, the feature encoder is frozen and a new classifier is fine-tuned using labeled novel data (*NS*) to enable predictions on unlabeled novel data (*NQ*). (c) Meta-Transfer combines meta-learning and transfer learning strategies. After the pre-training stage, the classifier is removed and the feature encoder is meta-trained to perform on various tasks to get meta-knowledge. (d) The proposed Adaptive Meta-Transfer differs from Meta-Transfer in that the classifier is used to eliminates domain bias during the meta-training stage. By employing an adversarial learning approach within a two-branch architecture, the feature encoder is able to adaptively eliminate bias, resulting in better transferability.

classes, K labeled samples (*support-set* $\mathcal{S}$), and N unlabeled samples (*query-set* $\mathcal{Q}$) within each class. The predictions of $\mathcal{Q}$ are based on the $\mathcal{S}$. Typically, for a task that is composed of C classes, K support samples for each class are usually formulated as a *C-way K-shot* task. The model is optimized to perform well on $\mathcal{D}_{base}(\mathcal{S}, \mathcal{Q})$ to get generalization ability (meta-knowledge) across various tasks on $\mathcal{D}_{novel}(\mathcal{S}, \mathcal{Q})$. The meta-learning strategy in FSL is shown in the upper right side of Fig. 1. Benefiting from this learning mechanism, meta-learning is a promising way for FSL [15,16], which enables the model to quickly adapt to different scenarios.

Inspired by the two learning strategies, some works [17–20] have proposed that meta-learning in cooperation with transfer learning becomes a stronger learning paradigm for FSL. We denote this two-stage learning paradigm as *Meta-Transfer*. Specifically, they first pre-train a standard classification network, i.e. feature encoder with a linear classifier, to get a converged feature encoder with transferability. Then they directly remove the linear classifier and meta-train the feature encoder to perform various tasks. However, with limited pre-training or support samples available, the performance of this paradigm still suffers. We perform episodic inference on the learned feature encoder and found two major reasons for misclassification. First, the model tends to place attention on small, incomplete features, which can negatively impact the similarity measurement between query and support samples in few-shot tasks. We attribute this phenomenon to an instance-specific bias arising from the presence of a classifier. Second, the less support information available, the more difficult for the model to distinguish commonalities between samples. We hold that different learning strategies will make a discrepancy between the two model's objective and naturally brings an *embedding gap*. The pre-trained feature encoder works together with the linear classifier to perform classification tasks, thus limiting its representation quality. Moreover, the episodic inference is based on the commonality between query and support samples, thus representing samples independently is sub-optimal.

In this work, we explore the insights into the cooperation between transfer learning and meta-learning strategies. Based on the few-shot classification (FSC) task, we uncover an overlooked discrepancy in learning objectives between the two strategies. Compared with the popularly used Meta-Transfer [19,21], we are concerned about the objective discrepancy and aim to detach undesired incomplete representations learned by pre-training. To this end, we propose a two-branch learning paradigm Adaptive Meta-Transfer (A-MET) to adaptively eliminate the incomplete representations for stronger feature embeddings with better transferability. The learning strategies Meta-Transfer and Adaptive Meta-Transfer are shown in Fig. 1. Moreover, we analyze the prediction mechanism and propose a new Global Similarity Compatibility Measure (GSCM) to jointly re-embed sample representations, which unite the query and support samples for more consistent predictions. Different from RelationNet [22] and CANet [23], GSCM considers sample correlation from a global level which reduces the probability of noise. GSCM does not introduce extra parameters, while efficiently facilitating metric-based meta-learning.

We conduct comprehensive experiments on the popular mini-ImageNet [1], tiered-ImageNet [24], Omniglot [25], and CUB [26] datasets. Experimental results demonstrate the effectiveness of the proposed method and show it has more advantages with large domain differences between the base and novel classes and less support information available.

Our contributions are summarized as follows.

- We are the first to explore the insights behind the cooperation between transfer learning and meta-learning strategies and uncover an overlooked but important discrepancy in learning objectives. Compared with the popularly used learning paradigm, we propose a simple but effective learning paradigm applicable for FSC with arbitrary network architectures.
- We formulate an effective learning paradigm A-MET for learning stronger feature embeddings with better transferability. A-MET adaptively eliminates undesired incomplete representations learned by the pre-training stage and solves the objective discrepancy between the two learning strategies, which has more advantages with large domain differences between the base and novel classes.

- We propose a simple representation measurement GSCM to represent samples by jointly re-embedding sample representations for more consistent prediction results within related samples. GSCM considers sample correlation from a global level, which reduces the probability of noise and has more advantages with less support information available.

2. Related work

2.1. Transfer learning for FSL

Transfer learning [27] has achieved great success in many tasks, especially when the available data is limited. Recent works apply transfer learning to perform FSL tasks [5–7]. They usually conduct pre-training on sufficient examples (*base classes* $\mathcal{D}_{base}$) and then fine-tune the model on the target task to learn unseen classes (*novel classes* $\mathcal{D}_{novel}$). Concretely, they pre-train a feature encoder f_θ and a classifier $C(\cdot|W_b)$ from scratch by minimizing a cross-entropy loss. Given the pre-trained feature encoder, they append a new classifier $C(\cdot|W_n)$ and fine-tuning the network using the few novel examples. During the fine-tuning stage, they have to freeze the feature encoder and only train a new classifier due to the data scarcity of novel classes [8–10]. However, their base classes in pre-training usually have the same domain as novel classes. This is seldom satisfied in real applications exists a large domain gap between the base and novel classes.

2.2. Meta-learning for FSL

Meta-learning is a de facto framework for FSL [1,28]. It entails training a meta-learner on a set of tasks (known as an "*episode*"), with the goal of extracting meta-knowledge that can be transferred to new tasks with scarce data. Various meta-learning architectures for FSL have been proposed, including memory-based methods [29–31], optimization-based methods [9,32,33], and metric-based methods [1].

Currently, the metric-based method has become a mainstream approach for FSL. The classic work Prototypical Networks [28] proposed to compute the average features of each class in the support-set as the *prototype* for each class. The classification of query samples is performed by calculating the similarity between each prototype and the query samples. Specifically, they compute an M-dimensional feature embedding for each support sample through an embedding function $f_\theta : \mathbb{R}^D \rightarrow \mathbb{R}^M$ with learnable parameters θ. The prototype is the mean feature vector of the K support that belongs to its class. However, in the more realistic 1-shot situation, it suffers a severe performance penalty.

An intuitive reason is that the prediction lacks sufficient support information. Another underlying reason that is overlooked by most works, that is the inference is fragile to irrelevant features. To be specific, neural networks cannot distinguish what is key information in the current task because each sample is considered independently. This problem has been concerned in some works [22,23,34], they worked on designing a complex network with greater capacity to focus on common features. RelationNet [22] presented a relation module as a learnable metric jointly trained with deep representations. CANet [23] proposed a cross attention network generate attention maps for each pair of prototype and query samples. ArL [34] proposed to learn object class concepts for relation learning. RENet [35] presented a relational embedding mechanism that enables learning from the relations between different classes. However, they consider sample correlation at a local level, which still suffers from the large variance across domains. Different from previous works, we propose a simple measurement GSCM to jointly represent samples from global level for more consistent predictions. GSCM does not introduce extra parameters, while efficiently facilitating metric-based meta-learning.

2.3. Meta-transfer

Many works [17,19] have shown that, rather than meta-learning from scratch, making a pre-training and transfer the learned knowledge to meta-learning can learn a stronger representation to generalize to new tasks. In these approaches, a standard classification network is initially pre-trained to obtain a converged feature encoder. Subsequently, the linear classifier is removed, and the feature encoder is meta-trained to perform various tasks. This learning paradigm can be denoted as Meta-Transfer (MET). However, whether this transfer pattern is appropriate remains unclear and requires further discussion.

Meta-Baseline [19] described and evaluated a conflict between the pre-training and meta-learning from the aspect of the generalization ability of base and novel classes. IFSL [36] explained the limitations of the pre-trained model in terms of structural causal. They argue the pre-trained knowledge is a confounder that limits the meta-learning performance. However, they are both not concerned about the objective discrepancy between the two learning strategies and solving it. CAE [37] claimed that the linear layer would constrain the ability of the feature encoder, thus bringing negative effects for downstream tasks. Their opinion is similar to ours, but it has not been verified in the FSL field.

Different from the above works, we specifically focus on the overlooked yet important problem of objective discrepancy between pre-training and meta-learning, aiming to bridge this gap for stronger feature embeddings with better transferability. LNL [38] proposed a regularization algorithm to constrain the model learning from biased data, where explicitly define the undesired attributes to not learn. Inspired by LNL, we formulate an effective learning paradigm that adaptively eliminates undesired representations to resolve the discrepancy in learning objective between the two training strategies. Different from LNL, the proposed learning paradigm does not require an explicit definition of the supervised attributes and can be applied to arbitrary network architectures.

3. Method

3.1. Task formulation

We consider the task of few-shot classification (FSC). Formally, we have a base class set $\mathcal{D}_{base}$ (training set) and a novel class set $\mathcal{D}_{novel}$ (test set), where the $\mathcal{D}_{base} \bigcap \mathcal{D}_{novel} = \varnothing$. The goal of FSC is to learn from $\mathcal{D}_{base}$ and generalize to $\mathcal{D}_{novel}$. Moreover, $\mathcal{D}_{novel}$ only has a few labeled data, which are termed as *support-set*, and the remaining unlabeled data termed as *query-set*. We can model the knowledge from $\mathcal{D}_{base}$ and test on $\mathcal{D}_{novel}$.

Following the two-stage learning paradigm, we first encode the transferable knowledge from $\mathcal{D}_{base}$ in a standard supervised manner (i.e. pre-training). We train a feature encoder f_θ with a linear classifier l_φ from scratch on the $\mathcal{D}_{base}$. Formally, we denote $P_\varphi(y|x;\theta)$ as the learned model, where x is the input data, y is the data label, φ and θ are the parameters for any linear classifier and feature encoder, respectively. By considering $L_y(x;\theta,\varphi)$ as the loss function and $\mathbb{E}(f_\theta, l_\varphi)$ as the objective function of $P_\varphi(y|x;\theta)$, the model can be optimized via:

$$\mathbb{E}(f_\theta, l_\varphi) = \frac{1}{N}\sum_{i=1\ldots N} L_y(x_i;\theta,\varphi), \tag{1}$$

where N denotes the number of samples in one batch. The loss function L_y for sample x can be defined as:

$$L_y = -\log\frac{\exp(p_k(x))}{\sum_j \exp(p_j(x))}, \tag{2}$$

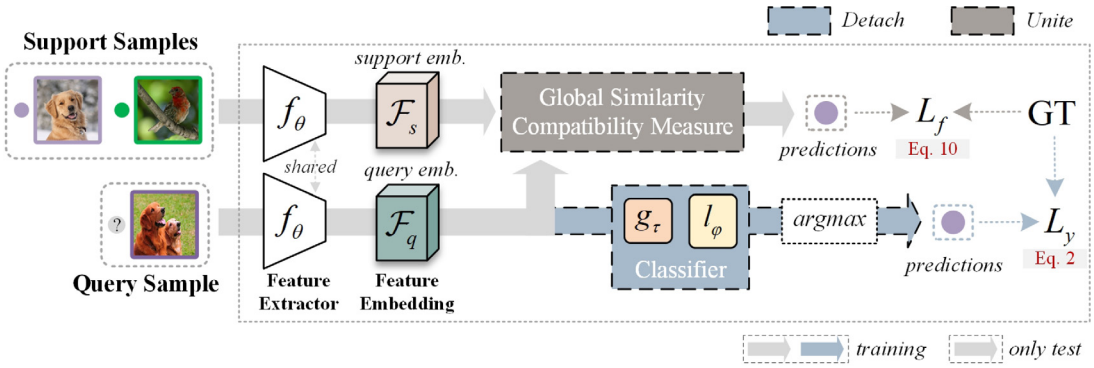

Fig. 2. Overview of the meta-learning stage of the proposed learning paradigm "Detach and Unite". The feature encoder f_θ and the linear layer l_φ are derived from the pre-training stage. The g_τ is a gradient reversal layer with parameter τ. "L" and "GT" represent loss function and ground truth, respectively. We do not explicitly designate a detailed architecture, since the proposed learning paradigm can be applied to arbitrary network architectures.

where $p_j(x) = l_\varphi(f_\theta(x))[j]$ denotes the logits of the classifier for the jth category of input sample x. We aim to minimize the objective function and find the optimal parameters to ensure an overall good prediction result:

$$\hat{\varphi}, \hat{\theta} \leftarrow \arg\min_{\varphi,\theta} \mathbb{E}(f_\theta, l_\varphi). \tag{3}$$

In the meta-training stage, the episode is designed to simulate the settings in meta-testing. Specifically, $\mathcal{D}_{base}$ is re-organized as training episodes $\{(\mathcal{S}_i, \mathcal{Q}_i)\}$. Each episode is formed by randomly selecting C classes from $\mathcal{D}_{base}$ and its support-set $\mathcal{S}_i = \{(x_s^{(i)}, y_s^{(i)}) | y_s^{(i)} \in \mathcal{D}_{base}\}_{i=1}^{C\times K}$, which contains K labeled samples for each class. The rest Q samples of those C classes are served as query-set $\mathcal{Q}_i = \{(x_q^{(i)}, y_q^{(i)}) | y_q^{(i)} \in \mathcal{D}_{base}\}_{i=1}^{C\times Q}$. The learned feature encoder f_θ maps each support-set $\mathcal{S}_i$ to an embedding space, where each class is represented by a single prototype $c_k \in \mathbb{R}^M$ that is surrounded by the examples of that class. M is the dimension of the feature embedded by f_θ. The prototype c_k of class k is defined as the average of all embedded support samples in that class:

$$c_k = \frac{1}{|S_k|} \sum_{(x_i, y_i)\in S_k} f_\theta(x_i), \tag{4}$$

where S_k denotes support samples from class k. Given a similarity measurement $m : \mathbb{R}^M \to \mathbb{R}^C$, where C is the number of classes in one episode, the probability distribution over classes for the query samples can be defined as:

$$p(y = k|x) = \frac{\exp(-m(f_\theta(x), c_k))}{\sum_{k'=1}^{C} \exp(-m(f_\theta(x), c_{k'}))}. \tag{5}$$

3.2. Adaptive meta-transfer

Due to the limited availability of pre-training samples, the pre-trained model inevitably leans towards instance-specific bias, which hinders its ability to transfer to new classes. We aim to train a feature encoder f_θ that focuses on more general features, and thus performs well on unseen novel classes. The feature encoder can be optimized via Eq. (1) in the pre-training stage. In this section, we focus on the meta-learning stage. Fig. 2 illustrates the meta-learning stage of the proposed learning paradigm, we denote the two-branch architecture as Adaptive Meta-Transfer.

The training data $\mathcal{D}_{base}$ and the test data $\mathcal{D}_{novel}$ both have complex and unknown distributions, which are referred as $U(x)$ and $V(x)$, respectively. We define the instance-specific bias in dataset $\mathcal{D}$ as $\mathcal{B}$. Assume that $\mathcal{B}$ contains every possible target bias $b(\cdot)$ that $\mathcal{D}$ can possess, where $b : \mathcal{D} \to \mathcal{B}$. The pre-training stage ensures an overall good prediction performance of the combination of the feature encoder f_θ and the linear classifier l_φ on $U(x)$ with the learning objective:

$$\mathcal{O}\langle f_\theta, l_\varphi\rangle : l_\varphi(f_\theta(x)) \sim U(x), \tag{6}$$

where $\langle\cdot\rangle$ represents the integrated optimization objective. However, $U(x)$ includes both domain-invariant and domain-specific features (i.e., bias) simultaneously. As a result, the feature encoder f_θ is inevitably leaning towards bias, and the learning objective of f_θ can be denoted as:

$$\mathcal{O}\langle f_\theta\rangle : f_\theta(x) \sim \mathcal{I}(x) + \mathcal{B}_U(x), \tag{7}$$

where $\mathcal{I}(x)$ is the domain-invariant feature distribution and $\mathcal{B}_U(x)$ is the domain-specific feature distribution on $U(x)$. Our goal is to detach the $\mathcal{B}_U(x)$ learned by $\langle f_\theta, l_\varphi\rangle$ from f_θ and make:

$$\mathcal{O}\langle f_\theta\rangle \to \mathcal{O}'\langle f_\theta\rangle, \text{where } \mathcal{O}'\langle f_\theta\rangle : f_\theta(x) \sim \mathcal{I}(x). \tag{8}$$

Measuring $\mathcal{B}_U(x)$ explicitly is however non-trivial. The cooperation between the f_θ and l_φ in the pre-training stage leads to our idea. We devote to decrease the impact of l_φ in f_θ in the meta-training stage. Specifically, the combination of f_θ and l_φ has been learned from $\mathcal{I}(x)$ and $\mathcal{B}_U(x)$ during the pre-training stage by minimizing the loss function L_y defined in Eq. (2). In the meta-learning stage, f_θ is further trained to provide consistent predictions on $U(x)$. Consequently, we aim to find the parameters θ that maximize the loss L_y of the classifier l_φ, while simultaneously seeking the parameters θ that minimize the loss L_f. In practice, we jointly train the f_θ and l_φ with adversarial strategy, and the optimization target can be achieved by gradient reversal layer g_τ, parameterized by τ. Thus, the f_θ and l_φ can be optimized iteratively through:

$$\hat{f}_\theta \leftarrow \langle \nabla L_f, -\tau \nabla L_y\rangle, \tag{9a}$$

$$\hat{l}_\varphi \leftarrow \langle \nabla L_y\rangle, \tag{9b}$$

where L_y denotes the loss function which has been defined in Eq. (2) and L_f can be defined as the cross-entropy loss for the probability distribution $p(y|x)$ in Eq. (5):

$$L_f = -\log \frac{\exp(-m(f_\theta(x), c_k))}{\sum_{k'=1}^{C} \exp(-m(f_\theta(x), c_{k'}))} = -\log p(y = k|x). \tag{10}$$

The forward and backward propagation of the gradient reversal layer can be defined as:

$$g_\tau(x) = x, \quad \frac{dg_\tau}{dx} = -\tau I, \tag{11}$$

where τ is a constant hyper-parameter of the gradient reversal layer and I is an identity matrix. Thus, the final objective function

can be denoted as:

$$\begin{aligned}\tilde{\mathbb{E}}(f_\theta, l_\varphi) &= \frac{1}{N_e}\Big(\sum_{i=1..N_e} L_f(h(m(f_\theta(x_i;\theta), c_i)), y_i) \\ &+ \sum_{i=1..N_e} L_y(l_\varphi(g_\tau(f_\theta(x_i;\theta);-\tau);\varphi), y_i)\Big) \\ &= \frac{1}{N_e}\Big(\sum_{i=1..N_e} L_f^i(x;\theta) + \sum_{i=1..N_e} L_y^i(x;\theta,\varphi,\tau)\Big),\end{aligned} \tag{12}$$

where N_e is the number of episodes. $h(\cdot)$ is a mapping function that maps the similarity matrix to prediction. m is a similarity measurement has described in Eq. (3). c is the class prototype defined in Eq. (2). The f_θ and l_φ can be iteratively optimized through the following functions (with learning rate α):

$$\hat{\varphi}, \hat{\theta} \leftarrow \arg\min_{\varphi,\theta} \tilde{\mathbb{E}}(f_\theta, l_\varphi), \tag{13a}$$

$$\hat{\theta} \leftarrow \theta - \alpha\left(\frac{\partial L_f}{\partial \theta} - \tau\frac{\partial L_y}{\partial \theta}\right), \tag{13b}$$

$$\hat{\varphi} \leftarrow \varphi - \alpha\frac{\partial L_y}{\partial \varphi}. \tag{13c}$$

Algorithm 1 summarizes the proposed learning paradigm. In particular, l_φ is a fully connected layer and is specifically used for the joint training stage and does not affect the final performance. The critical requirement is to ensure the same l_φ is utilized in both learning stages.

Algorithm 1 The proposed learning paradigm. n: the number of examples in the $\mathcal{D}_{base}$. N_c: the number of classes. N_e: the number of episodes. C, K, and Q have been defined above. RandomSample(A, B) denotes chosen B elements randomly from A, without replacement. $(\cdot)_{ij}$ denotes the samples of $(\cdot)$ with i-th episode and j-th class.

Input: The training set $\mathcal{D}_{base} = \{(x_1, y_1), ..., (x_n, y_n)\}$, where each $y_i \in \{1, ..., N_c\}$.
Output: Feature extractor f_θ.

```
Randomly initialize θ and φ
for samples in D_base do
    Optimize θ and φ → l_φ(f_θ(x)) ~ U(x)   ▷ Eq. (1)
end for
for i in {1, ..., N_e} do
    D_i ← RandomSample({1, ..., N_c}, C)
    for j in {1, ..., C} do
        S_ij ← RandomSample(D_ij, K)
        Q_ij ← RandomSample(D_ij \ S_ij, Q)
    end for
    f_θ(S_i, Q_i) → (f_θ(S_i), f_θ(Q_i)) ∈ ℝ^{(K+Q)×M}
        update L_f(h(m(f_θ(Q_i; θ), c_i)), {Y_i})
    l_φ(f_θ(Q_i)) ∈ ℝ^Q
        update L_y(l_φ(g_τ(f_θ(Q_i; θ); −τ); φ), {Y_i})
    Optimize θ and φ:
        θ̂ ← θ − α(∂L_f/∂θ − τ ∂L_y/∂θ)        ▷ Eq. (13b)
        φ̂ ← φ − α ∂L_y/∂φ                     ▷ Eq. (13c)
end for
```

3.3. Global similarity compatibility measure

Intuitively, predictions on the query sample can be made by comparing its similarity with each support prototype. However, in the more realistic 1-shot situation, the model suffers a severe performance penalty. This can be attributed to the fragility of the inference process when dealing with irrelevant features. In other words, the model struggles to distinguish the key information relevant to the current task. Specifically, a generic feature encoder f_θ maps the input data $\mathcal{D}$ to feature embedding: $\mathcal{D} \rightarrow \mathbb{R}^M$. Each episode can be modeled as $f_\theta(\{x_i\}_{i=1}^N), N = (K+Q)\times C$, and the feature embedding is a row-stacked matrix $\mathcal{F} = \{f_\theta(x_i)\}_{i=1}^N \in \mathbb{R}^{N\times M}$. Therefore, the collection of feature embedding for an episode can be defined as:

$$\mathcal{F}_e = \begin{bmatrix}\mathcal{F}_s \\ \mathcal{F}_q\end{bmatrix} \in \mathbb{R}^{N\times M}, \tag{14}$$

where the support-set and query-set feature embedding can be represented as:

$$\mathcal{F}_s = [f_s^{(1)}, f_s^{(2)}, \ldots, f_s^{(K\times C)}]^T \in \mathbb{R}^{K\times C\times M}, \tag{15a}$$

$$\mathcal{F}_q = [f_q^{(1)}, f_q^{(2)}, \ldots, f_q^{(Q\times C)}]^T \in \mathbb{R}^{Q\times C\times M}, \tag{15b}$$

where $f_s \in \mathbb{R}^M$ and $f_q \in \mathbb{R}^M$ represent the support sample and query sample feature, respectively. To make a prediction on the query-set, a common practice is to calculate the similarity between c_k and f_q, where c_k is the support prototype defined in Eq. (4). However, $\mathcal{F}_e$ treats the query and support feature embeddings independently and the measurement is fragile to noise introduced by scale, occlusion, background, etc.

To tackle this issue, we propose establishing global dependencies between the query sample and each support sample. By incorporating global dependencies, the model can consider more comprehensive and informative feature representations, enabling predictions based on a holistic view of the input. In practice, we transform the feature embedding of each sample into a global similarity representation $\mathcal{T}$, from space $\mathbb{R}^M$ to $\mathbb{R}^N$, to construct the correlation with each sample. The transformation can be obtained by $\mathcal{F}_e \times \mathcal{F}_e' \rightarrow \mathcal{T}_e$. The $\mathcal{F}_e'$ can be defined as:

$$\mathcal{F}_e' = \begin{bmatrix}\mathcal{F}_s' & \mathcal{F}_q'\end{bmatrix} \in \mathbb{R}^{M\times N}, \tag{16}$$

where

$$\mathcal{F}_s' = [f_s^{(1)^T}, f_s^{(2)^T}, \ldots, f_s^{(K\times C)^T}] \in \mathbb{R}^{M\times K\times C}, \tag{17a}$$

$$\mathcal{F}_q' = [f_q^{(1)^T}, f_q^{(2)^T}, \ldots, f_q^{(Q\times C)^T}] \in \mathbb{R}^{M\times Q\times C}. \tag{17b}$$

The transformed feature embedding collection $\mathcal{T}_e$ can be denoted as:

$$\mathcal{T}_e = \begin{bmatrix}\mathcal{T}_s \\ \mathcal{T}_q\end{bmatrix} \in \mathbb{R}^{N\times N}, \tag{18}$$

where

$$\mathcal{T}_s = [t_s^{(1)}, t_s^{(2)}, \ldots, t_s^{(K\times C)}]^T \in \mathbb{R}^{K\times C\times N}, \tag{19a}$$

$$\mathcal{T}_q = [t_q^{(1)}, t_q^{(2)}, \ldots, t_q^{(Q\times C)}]^T \in \mathbb{R}^{Q\times C\times N}. \tag{19b}$$

In practice, the transformation can be computed efficiently by the inner product operation. Each transformed weight $w_{(i,j)}$ represents the similarity between the sample x_i and x_j. The feature embedding $t^{(i)}$ of x_i can be denoted as:

$$t^{(i)} = [w_{(i,1)}, \ldots, w_{(i,j)}, \ldots, w_{(i,N)}] \in \mathbb{R}^{N\times 1}, \tag{20}$$

where $w_{(i,j)}$ equals to 1 when $i = j$, which represents the similarity between the sample and itself.

The $\mathcal{T}_e$ considers that two samples satisfying similarity consistency are compatible. In other words, each sample is represented based on its relationship with other samples. When we say that two samples are similar, it means that they share the same relationship with other samples, rather than simply measuring the similarity between the two samples themselves. We define the global similarity compatibility measure between the sample

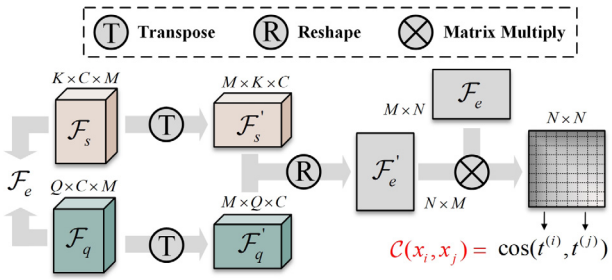

Fig. 3. The calculation process of the proposed global similarity compatibility measure.

x_i and x_j as follows:

$$\begin{aligned} \mathcal{C}(x_i, x_j) &= cos(t^{(i)}, t^{(j)}) \\ &= \frac{t^{(i)} \cdot t^{(j)}}{\|t^{(i)}\| \|t^{(j)}\|} \\ &= \frac{\sum_{k=1}^{N} w_{(i,k)} \times w_{(k,j)}}{\sqrt{\sum_{k=1}^{N} (w_{(i,k)})^2} \times \sqrt{\sum_{k=1}^{N} (w_{(k,j)})^2}}. \end{aligned} \tag{21}$$

If sample x_i and x_j is compatible, $\mathcal{C}(x_i, x_j) \rightarrow 1$, and the relationship between the two samples and other support samples is consistent. The GSCM reduces the probability of noise being involved in the compatible set and enhances the robustness of predictions. Meanwhile, GSCM is decoupled from a specific network structure and can be applied to any features derived from a feature encoder. An illustration of the GSCM is shown in Fig. 3.

4. Results on standard benchmarks

4.1. Datasets

To verify the proposed method, we perform the few-shot classification task under three scenarios:

- Handwritten character recognition task, with one handwritten dataset Omniglot [25].
- Generic object recognition task, with two general object datasets based on ImageNet [39], i.e. mini-ImageNet [1] and tiered-ImageNet [24].
- Fine-grained image classification task, with one fine-grained classification dataset CUB-200-2011 [26] (CUB for short).

Omniglot [25] is a dataset of 1623 handwritten characters (classes) collected from 50 different alphabets. Each character contains 20 examples, which are written by a different human subject. Omniglot has a two-level hierarchy, i.e. alphabets and characters. A large number of classes and a few examples makes Omniglot a suitable benchmark for FSC. A common data setting [1,28] flattens and ignores its two-level hierarchy of alphabets and characters. They use 1200 characters and augment the character with rotations (4800 classes in total) for training, and the remaining characters for testing. We follow this setting for the evaluation of the proposed method. Furthermore, We also follow the settings of original splits [25] which is more challenging since the split is on the alphabets level, where 30 alphabets are for training and 20 for testing. We reserve the 5 smallest alphabets (i.e. with the least number of character classes) from the training set for validation [40].

mini-ImageNet [1] is a general object recognition benchmark for FSC, which contains 100 classes selected from ILSVRC-2012 [39], and each class contains 600 samples. The mini-ImageNet is randomly split into 64/16/20 classes for training, validation, and testing. Following previous works [6,20,41], we use the data split proposed by Ravi and Larochelle [41] in the experiments.

tiered-ImageNet [24] is another common ImageNet-based benchmark with a much larger scale than mini-ImageNet. It contains 608 classes with 779,165 images from 34 super-categories. The super-category are split into 20/6/8 disjoint categories, resulting in 351, 97, and 160 classes for training, validation, and testing. Each class contains 1300 samples. It needs to note that, mini-ImageNet has a smaller data capacity, it did not concern with the similarity between base classes and novel classes. While the settings of tiered-ImageNet are more challenging since base classes and novel classes come from different super-categories, thus the training data and test data of tiered-ImageNet have much larger differences in appearance and semantics.

Caltech-UCSD Birds-200-2011 (CUB) [26] is a dataset for fine-grained classification with 200 different bird species and 11,788 images in total. In practice, We did not use the provided bounding boxes to crop the images, instead, the original raw images are used, which provides a harder challenge. Note that CUB is only used for cross-domain evaluation in the experiment.

Fig. 4 gives some examples for these datasets above.

4.2. Preprocessing and data-augmentation

To perform the few-shot learning task, it is necessary to organize each dataset into collections of episodes (i.e., tasks). Specifically, each task is created by randomly selecting C classes and K support samples for each class. For Omniglot, we construct a task with either 5 or 20 classes, where one support sample for each class. These tasks are referred to as 5-way 1-shot and 20-way 1-shot tasks, respectively. As for the mini- and tiered-ImageNet, we follow the previous works and create tasks with 5-way 1-shot and 5-way 5-shot configurations. For CUB, we assess the performance of the model on both 5-way and 20-way tasks, considering 1-shot and 5-shot scenarios.

In the experiments, standard data augmentation is applied, including random crop, horizontal flip, and color jittering. We resize the samples to 28×28 for Omniglot, 84×84 for mini-ImageNet and tiered-ImageNet following previous works.

4.3. Architectures

We adopt two commonly used embedding architectures ConvNet-4 and ResNet-12 that follow the most recent works [19,28]. The **ConvNet-4** is composed of four convolutional blocks, and each block comprises a 64-filter 3×3 convolution, batch normalization layer, and a ReLU nonlinearity. After each convolutional block, a 2×2 max-pooling layer is used to downsample the feature maps. ConvNet-4 results in 64-dimensional output space. The widths of the four convolutional blocks both are 64. The **ResNet-12** follows the architecture of ResNet with four basic blocks, each having three convolutional operations. Each basic block contains one residual operation. The widths of the basic block of the four stages are [64, 128, 256, 512]. Each basic block has three 3×3 convolution layers with batch normalization and a 0.1 leaky ReLU. A max-pooling layer with stride 2 after the three convolution layers at each block. ResNet-12 results in 512-dimensional output space. Fig. 5 shows the architecture of ConvNet-4 and ResNet-12.

Fig. 4. Example images sampled from Omniglot, mini-ImageNet, tiered-ImageNet, and CUB, respectively.

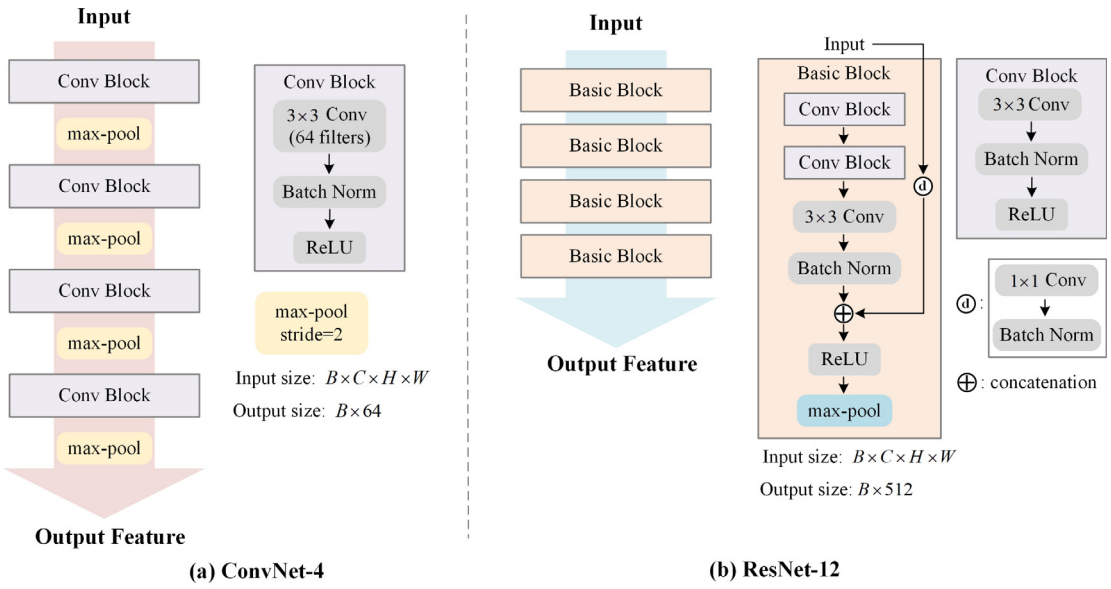

Fig. 5. Architecture of the feature encoder used in the experiments.

4.4. Implementation details

Training details. All of our models were trained via SGD optimizer with momentum 0.9. For the pre-training stage, the initial learning rate is 0.1 and the decay factor is 0.1. The max training epochs is 100 and the learning rate decays at epoch 90. The batch size for Omniglot and mini-ImageNet is 128 and for tiered-ImageNet is 256. For the meta-training stage, the initial learning rate is 1e−4 and the decay factor is 0.1. We also train the model with 100 epochs and each epoch contains 1000 episodes (randomly sampled), the learning rate decays at epochs 40 and 80. Specifically, the meta-training is performed on a 5-way 1-shot task, where we randomly select 5 classes each with 1 support sample and 15 query samples. The proposed A-MET only introduces one extra hyper-parameters, i.e. the gradient weight τ of gradient reversal layer g_τ in Eq. (11). We tried three settings of τ in the experiment (0.1, 0.3, 0.5), and find different settings will affect the convergence difficulty of the model. For the trade-off between training speed and accuracy, the gradient weight τ is set to 0.1 in all experiments. For other hyper-parameters, the weight decay is 5e−4. For tiered-ImageNet, we freeze the batch normalization layer in the meta-training stage.

Episodic evaluation. Both the two training stage (pre-training and meta-training) is evaluated in an episodic manner, and the sampling strategy is consistent with meta-training. Specifically, the validation interval is 20 epochs and each validation stage comprises 600 episodes. To get a fair comparison, we perform model selection according to the validation set. The training loss curve and the validation result of mini-ImageNet are shown in Fig. 6. We can observe that A-MET achieves higher validation

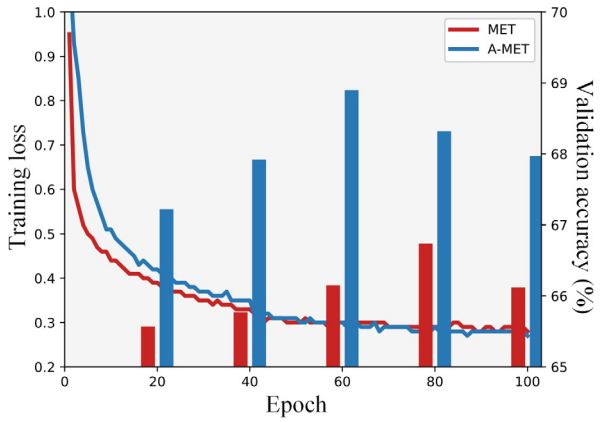

Fig. 6. Training loss curves and validation results for the proposed paradigm A-MET and the baseline paradigm MET. The training is performed on mini-ImageNet dataset with ResNet-12 architecture.

accuracy at 60 epochs early than MET. It suggests that A-MET is able to learn more effectively from limited labeled data. At the test stage, we perform 2000 episodes and repeat this procedure 5 times to get the average accuracy. We report the average accuracy and the corresponding 95% confidence interval for all experiments.

Experimental environment. The experiment was performed on Ubuntu 16.04 with 1 NVIDIA RTX 3090 GPU. The implementations are based on PyTorch 1.8 and Python 3.7. For Omniglot and mini-ImageNet, the pre-training stage takes about 5 GB of GPU memory. For tiered-ImageNet, the pre-training stage takes about 11 GB of GPU memory. In the meta-training stage, both the above datasets take about 5 GB of GPU memory.

4.5. Main results

We compare the proposed method with classic and state-of-the-art works on the FSC task. Matching Networks [1] uses non-standard train/test splits in the experiment and ProtoNet [28] did not perform on the ResNet-12, so we use the re-implemented results from previous work [19,42]. It should be noted that the comparison methods introduce more well-designed architectures, while our method achieves competitive or even better performance based on a plain network without any extra modules. In the following experiments, we denote **Linear** as the transfer learning strategy, **MET** (MEta-Transfer) as the previous two-stage learning paradigm, and **A-MET** (Adaptive MEta-Transfer) as the proposed learning paradigm.

Results on ImageNet-based Benchmarks. Table 1 present the comparison results on two ImageNet-based datasets with different degrees of domain overlap, utilizing two distinct feature encoder structures. It is evident that MET offers a limited improvement on both datasets and even exhibits negative effects on tiered-ImageNet. This finding highlights the impact of instance-specific biases on the encoder, resulting from different learning objectives. Consequently, by eliminating this bias and making the learning objective of the encoder towards general features, a significant improvement is observed on A-MET. Moreover, we observed that the proposed method consistently improves the model performance on the two common architectures. Typically, the improvement from A-MET on tiered-ImageNet is more noticeable. This phenomenon supports our analysis that MET is more prone to undesired incomplete features, which limits the transferability of the model for generalizing to novel classes. We also notice that GSCM is always helpful to the average accuracy, and has more advantages in the challenging 1-shot setting. This demonstrates that GSCM helps to distinguish commonalities when with fewer support samples available. Overall, the improvement on tiered-ImageNet is larger than that on mini-ImageNet. This proves that the proposed method benefits the scenarios with large domain differences between the base and novel classes. Compared with other methods, the proposed method obtains comparable or even better performance with a plain network.

Results on Omniglot. The comparison results of Omniglot are shown in Table 2. We first evaluate the model on the common data setting ("Common") proposed by Vinyals et al. [1]. This an easy setting and all works achieve high accuracy on the 5-way 1-shot task. Furthermore, we evaluate the model on a harder setting ("Hard") [40], which is more challenging to solve. Under this setting, the proposed method obtains better results compared to other methods. This demonstrates that the proposed method has more advantages in the scenario with large domain differences between the base and novel classes.

Cross-domain Evaluation. We conducted further evaluation of the cross-domain transferability of the proposed method in two settings: (i) from the generic object dataset mini-ImageNet to the handwritten character dataset Omniglot, and (ii) from the generic object dataset mini-ImageNet to the fine-grained dataset CUB. Specifically, the model was trained on the training set of mini-ImageNet, which consists of 64 classes and follows the standard setting. For the Omniglot dataset, we employed the challenging data split based on the alphabets level, and the test set for cross-domain evaluation contained 20 alphabets. For CUB, we used the entire dataset for evaluation, which consists of 200 classes. The evaluation results are shown in Tables 3 and 4, respectively. From the table, we observe that A-MET effectively promotes the transferability of the model. For Omniglot, the improvement is larger on the 1-shot than on the 5-shot for both 5-way and 20-way settings. For more challenging CUB, the overall improvement is smaller than it was on Omniglot. A possible reason is that fine-grained classification task requires more attention to detail, but A-MET focuses more on general characteristics. The results also support our analysis that MET is prone to undesired representations and the proposed method is more effective in challenging settings. Moreover, GSCM achieves more progress on 1-shot than on 5-shot, which proves that giving suitable embeddings for the sample always helps.

Robustness Analysis. Tables 5, 6, and Fig. 7 present the quantitative and graphical results demonstrating the model's robustness in generalizing to a smaller number of shots or a larger number of ways. First, we test the model trained on 5-way 5-shot tasks to smaller K-shot settings ($K < 5$). We can observe that the performance of the model decreases sharply when 2-shot $\rightarrow$ 1-shot. Intuitively, a possible reason is that the 1-shot setting has much uncertainty, otherwise this uncertainty will be diminished. Moreover, we test the model trained on 5-way 1-shot to larger C-way settings ($C > 5$). We can observe that A-MET shows an advantage in the gradually increasing C-way. Both the two curves show that GSCM benefits the 1-shot situation and A-MET improves the overall performance by learning a general feature representation.

4.6. Ablation analysis

We perform the ablation analysis to verify the effectiveness of each component in the proposed method. The ablation study is performed on the mini- and tiered-ImageNet for 5-way 1-shot and 5-shot tasks, respectively. We use the plain backbone ResNet-12 as the feature extractor. The ablation study results are shown in Tables 8, 7, Figs. 8, and 9, respectively. The first row of Table 8 represents the pre-training stage. From Table 8, we get an average improvement of 5.62% of the proposed method.

Table 1
Comparison results on the 5-way 1-shot and 5-way 5-shot classification task. Average accuracy (%) with a 95% confidence interval. The best results are in **bold black**. The red font denotes improvement based on the baseline MET [1,15,19,21–23,28,33–35,42–44].

Method	Backbone	mini-ImageNet		tiered-ImageNet	
		1-shot	5-shot	1-shot	5-shot
Matching Networks[a] [1]	ConvNet-4	43.56 ± 0.84	55.31 ± 0.73	–	–
MAML [15]	ConvNet-4	48.70 ± 1.84	63.11 ± 0.92	51.67 ± 1.81	70.30 ± 1.75
ProtoNet [28]	ConvNet-4	49.42 ± 0.78	68.20 ± 0.66	53.31 ± 0.89	**72.69 ± 0.74**
RelationNet [22]	ConvNet-4	50.44 ± 0.82	65.32 ± 0.70	54.48 ± 0.93	71.32 ± 0.78
Linear	ConvNet-4	50.44 ± 0.47	65.37 ± 0.37	51.68 ± 0.50	70.30 ± 0.36
MET	ConvNet-4	51.16 ± 0.46	66.64 ± 0.37	51.81 ± 0.49	70.21 ± 0.36
A-MET (Ours)	ConvNet-4	54.10 ± 0.47 +2.94	68.77 ± 0.38 +2.13	54.92 ± 0.52 +3.11	71.77 ± 0.42 +1.56
A-MET w/ GSCM (Ours)	ConvNet-4	**56.64 ± 0.51** +5.48	**69.88 ± 0.38** +3.24	**59.65 ± 0.54** +7.84	72.67 ± 0.43 +2.46
Matching Networks[a] [1]	ResNet-12	63.08 ± 0.80	75.99 ± 0.60	68.50 ± 0.92	80.60 ± 0.71
ProtoNet[a] [28]	ResNet-12	60.37 ± 0.83	78.02 ± 0.57	61.74 ± 0.77	80.00 ± 0.55
CANet [23]	ResNet-12	63.85 ± 0.48	79.44 ± 0.34	69.89 ± 0.51	84.23 ± 0.37
ProtoNets+TRAML [43]	ResNet-12	60.31 ± 0.48	77.94 ± 0.57	–	–
Meta-Baseline [19]	ResNet-12	63.17 ± 0.23	79.26 ± 0.17	68.62 ± 0.27	83.74 ± 0.18
ConstellationNet [44]	ResNet-12	64.89 ± 0.23	79.95 ± 0.17	–	–
RENet [35]	ResNet-12	67.60 ± 0.44	82.58 ± 0.30	71.61 ± 0.51	85.28 ± 0.35
MetaOptNet [33]	ResNet-12	62.64 ± 0.61	78.63 ± 0.46	65.99 ± 0.72	81.56 ± 0.53
MetaOptNet+ArL [34]	ResNet-12	65.21 ± 0.58	80.41 ± 0.49	–	–
APP2S [42]	ResNet-12	66.25 ± 0.20	83.42 ± 0.15	72.00 ± 0.22	86.23 ± 0.15
DeepBDC [21]	ResNet-12	67.34 ± 0.43	**84.46 ± 0.28**	72.34 ± 0.49	**87.31 ± 0.32**
Linear	ResNet-12	62.07 ± 0.46	78.90 ± 0.33	65.76 ± 0.55	79.15 ± 0.41
MET	ResNet-12	63.16 ± 0.47	78.86 ± 0.33	66.25 ± 0.54	79.30 ± 0.40
A-MET (Ours)	ResNet-12	64.61 ± 0.47 +1.45	80.06 ± 0.32 +1.20	69.39 ± 0.57 +3.14	81.11 ± 0.39 +1.81
A-MET w/ GSCM (Ours)	ResNet-12	**68.47 ± 0.51** +5.31	80.89 ± 0.33 +2.03	**74.08 ± 0.54** +7.83	84.91 ± 0.35 +5.61

[a]Denotes the result obtained by re-implemented.

Table 2
Comparison results of 5-way and 20-way 1-shot classification tasks on Omniglot. The feature encoder for all methods is ConvNet-4. **Common** and **Hard** represent data split setting.

Setting	Method	5-way	20-way
Common	Matching Networks [1]	97.90	93.50
	ProtoNet [28]	98.80	96.00
	RENet [35]	99.32	96.73
	MET	98.50	96.00
	A-MET w/ GSCM (Ours)	99.50	98.53
Hard	Matching Networks [1]	97.44	93.91
	ProtoNet [28]	97.95	94.46
	RENet [35]	98.18	95.70
	MET	98.31	95.31
	A-MET w/ GSCM (Ours)	98.74	96.12

Table 3
Evaluation of cross-domain transferability: mini-ImageNet → Omniglot, with ResNet-12.

Method	5-way		20-way	
	1-shot	5-shot	1-shot	5-shot
MET	76.16	90.36	52.36	75.35
A-MET	81.84	93.29	60.94	82.12
A-MET w/ GSCM	86.89	94.17	64.52	83.35

Table 4
Evaluation of cross-domain transferability: mini-ImageNet → CUB, with ResNet-12.

Method	5-way		20-way	
	1-shot	5-shot	1-shot	5-shot
MET	45.41	62.01	14.69	23.91
A-MET	47.28	64.64	18.48	28.21
A-MET w/ GSCM	49.46	64.95	22.01	31.43

Table 5
Quantitative results of the robustness of the model in generalizing to a smaller number of shots.

shots	MET	A-MET	A-MET w/ GSCM
5[a]	78.37 ± 0.34	80.20 ± 0.32	80.82 ± 0.33
4	77.18 ± 0.35	78.70 ± 0.34	79.55 ± 0.35
3	74.88 ± 0.37	76.53 ± 0.36	77.83 ± 0.37
2	71.05 ± 0.41	72.70 ± 0.40	74.75 ± 0.42
1	63.13 ± 0.46	64.61 ± 0.47	68.47 ± 0.51

[a]Represents the model is trained under this setting.

Table 6
Quantitative results of the robustness of the model in generalizing to a larger number of ways.

ways	MET	A-MET	A-MET w/ GSCM
5[a]	63.16 ± 0.47	64.41 ± 0.47	68.47 ± 0.51
6	59.00 ± 0.41	60.36 ± 0.42	64.16 ± 0.46
7	55.16 ± 0.38	56.72 ± 0.38	60.71 ± 0.41
8	52.11 ± 0.34	53.86 ± 0.35	57.20 ± 0.38
9	47.16 ± 0.31	51.07 ± 0.31	54.76 ± 0.30
10	42.99 ± 0.26	48.85 ± 0.29	52.31 ± 0.33

[a]Represents the model is trained under this setting.

Table 7
Average 5-way accuracy on mini-ImageNet. Base gen. and Novel gen. denote the generalization ability for the base and novel classes, respectively.

	Base gen.		Novel gen.	
	1-shot	5-shot	1-shot	5-shot
MET	87.26	93.53	65.07	80.26
A-MET	87.37	93.29	69.76	81.98

Effects of A-MET. Table 8 shows the prediction results on the mini- and tiered-ImageNet. As mentioned earlier, mini-ImageNet has a large domain overlap as it does not consider the similarity between base and novel classes, whereas the tiered-ImageNet does the opposite. We can observe that the proposed A-MET paradigm is effective in both situations and performs better on the tiered-ImageNet dataset. This demonstrates that A-MET can detach the instance-specific bias and enhance the transferability

Table 8
Ablation study on mini- and tiered-ImageNet for 5-way 1-shot and 5-shot tasks. The first row represents the feature encoder pre-trained in a standard supervised manner. The proposed A-MET is represented as MET with ADAPTIVE.

MET	ADAPTIVE	GSCM	mini-ImageNet				tiered-ImageNet			
			1-shot	Δ	5-shot	Δ	1-shot	Δ	5-shot	Δ
			62.07	–	78.90	–	65.76	–	79.15	–
✓			63.16	+1.09	78.86	−0.04	66.25	+0.49	79.30	+0.15
✓	✓		64.61	+2.54	80.06	+1.16	69.39	+3.63	81.11	+1.96
✓		✓	66.75	+4.68	79.08	+0.18	72.81	+7.05	80.13	+0.98
✓	✓	✓	68.47	+6.40	80.89	+1.99	74.08	+8.32	84.91	+5.76

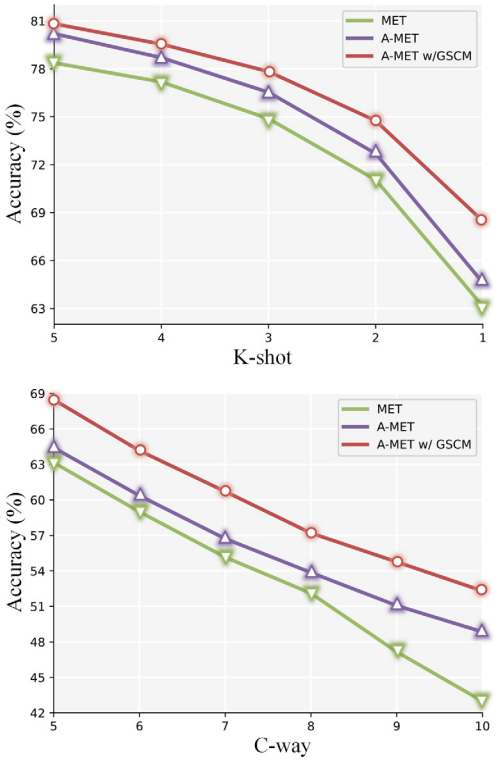

Fig. 7. Performance of generalizing to smaller shot and larger way. When the shot changes, the way is fixed, and vice versa.

of the model to novel classes. The results also highlight the importance of A-MET in adapting to different datasets with varying domain overlaps.

Fig. 8 presents the visualized Class Activation Mapping (CAM) of query samples for three different methods. We observe that the feature encoder, which is pre-trained using the standard supervised method, tends to focus on small and incomplete features ("Linear"). Despite the fine-tuning of the encoder through the meta-learning strategy to adapt it to few-shot tasks, there are still features that are not suitable for the similarity measurement between two samples ("MET"). Such a phenomenon might result in suboptimal performance in downstream tasks or operations. In contrast, the model trained with the proposed A-MET paradigm can capture more general and robust features. The visualized CAM results demonstrate the effectiveness of A-MET in learning feature representations that are generalizable and more robust to downstream tasks.

Table 7 presents the generalization ability of the model for both base and novel classes. The generalization performance of base classes is evaluated by sampling episodes from unseen samples in the base classes, while the generalization performance of novel classes refers to the performance of episodes sampled from novel classes. Specifically, we define the base classes as the training set and the unseen samples are selected from ILSVRC-2012 that are not included in mini-ImageNet. For novel classes, we use a combination of the validation set and test set, which includes a total of 36 classes. From the table, we can see that the model trained with A-MET achieves higher generalization performance for novel classes while maintaining competitive performance for base classes compared to MET.

Effects of GSCM. The quantitative results are shown in Table 8. We can see that GSCM has a significant improvement in the evaluation metric. It improves the average accuracy by 3.22% compared with MET. Benefiting from GSCM, the predictions of query samples can refer to the corresponding support samples in an episode to obtain more consistent prediction results. Compared with 1-shot, the improvement on 5-shot is limited. This can be attributed to the fact that the 5-shot setting provides more support information, which in turn reduces the impact of irrelevant features on prediction results.

The visualizations of the focal regions on query samples with different supports are shown in Fig. 9. The visualization results demonstrate that the focal regions of the query samples are more consistent with the corresponding support samples, which confirms the effectiveness of GSCM in reducing the impact of irrelevant features on the predictions of the model.

5. Conclusion

In this paper, we present a simple method that enables effective few-shot classification task. First, we explore the insights into the cooperation between transfer learning and meta-learning strategies based on the Few-shot Classification (FSC) task. Our experiments reveal an overlooked discrepancy between the two learning strategies. Second, we propose a new learning paradigm for the FSC task that aims to detach undesired representations in the meta-training stage. Additionally, we analyze the prediction mechanism and propose a new measurement for more consistent predictions. Finally, we evaluate the proposed method on four public FSC benchmarks. Experimental results demonstrate that the proposed method can consistently improve performance in common settings and achieve better generalization in novel classes. The proposed method is decoupled from the specific network structure and can be applied to arbitrary architectures. Meanwhile, the proposed method has more advantages with large domain differences between the base and novel classes and less support information available. Despite its promising performance, the proposed A-MET paradigm does have certain limitations. It is specifically designed for a two-stage few-shot learning strategy, as the key classifier of the adversarial training is derived from the first stage. Additionally, the effectiveness of GSCM relies to some extent on the feature embedding. Compared to employing GSCM on the MET directly, A-MET can further demonstrate its ability by ensuring a robust feature embedding. A potential solution to address these limitations is to design a debiasing network that can debias spontaneously. This can be achieved by integrating a bias indicator into the network and training the model end-to-end. This approach is similar to the work proposed in [45] and has shown feasibility. In future work, we plan to investigate an end-to-end solution for the few-shot

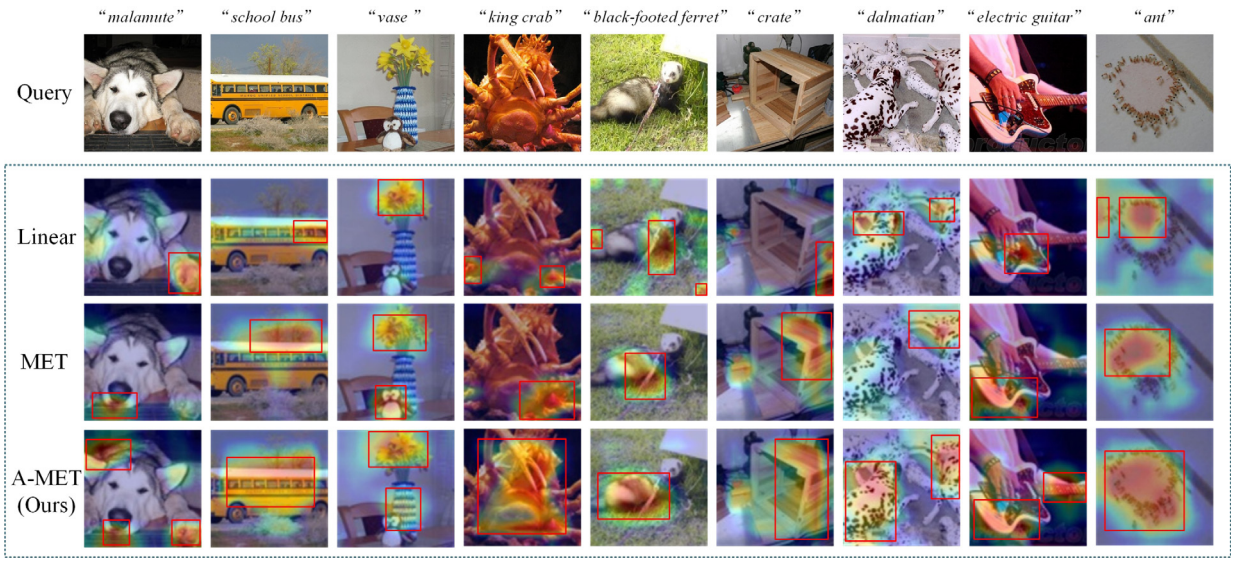

Fig. 8. Visualizations of class activation mapping for query samples. **Linear** represents the feature encoder pre-trained in a standard supervised manner.

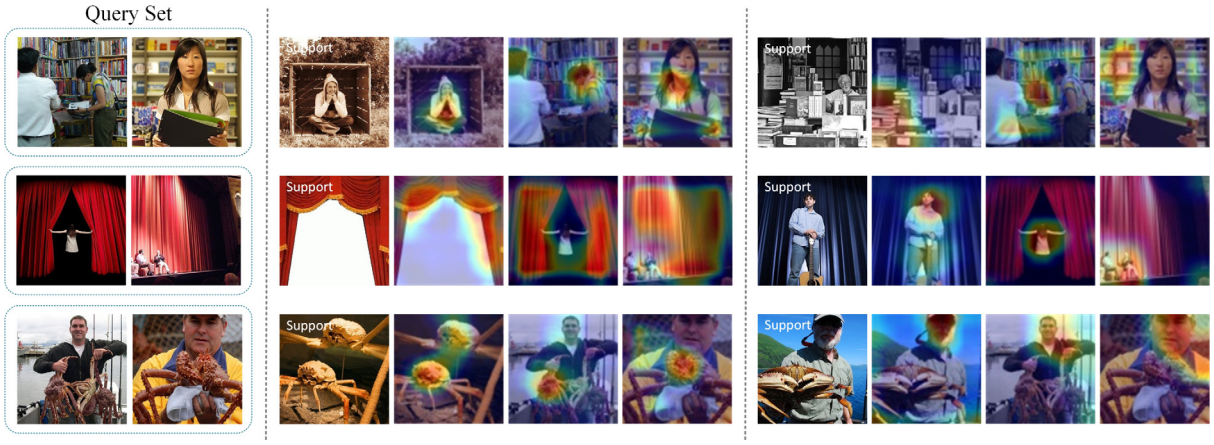

Fig. 9. Class activation mapping (CAM) of query samples based on different support samples. It reveals that the prediction of query samples should consider support samples to get more consistent focal regions. Note that the CAM is derived from the proposed method.

learning task that incorporates the advantages of the proposed method. Additionally, we aim to explore the applicability of this approach to various other few-shot learning tasks.

CRediT authorship contribution statement

Yaoyue Zheng: Conceptualization, Data curation, Investigation, Methodology, Software, Writing – original draft. **Xuetao Zhang:** Funding acquisition, Investigation, Supervision, Writing – review & editing. **Zhiqiang Tian:** Funding acquisition, Investigation, Supervision, Writing – review & editing. **Wei Zeng:** Methodology, Writing – review & editing. **Shaoyi Du:** Methodology, Writing – review & editing.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

The data used in this study are publicly available.

Acknowledgments

This work was supported by NSFC, China under Grant No. 62173269, the Natural Science Basic Research Plan in Shaanxi Province of China under Grant No. 2022JM-324, and the Social Science Foundation of Shaanxi Province of China under Grant No. 2021K014.

References

[1] O. Vinyals, C. Blundell, T. Lillicrap, D. Wierstra, et al., Matching networks for one shot learning, in: Advances in Neural Information Processing Systems, Vol. 29, 2016, http://dx.doi.org/10.48550/arXiv.1606.04080.
[2] L. Zhang, S. Zhang, B. Zou, H. Dong, Unsupervised deep representation learning and few-shot classification of PolSAR images, IEEE Trans. Geosci. Remote Sens. 60 (2020) 1–16, http://dx.doi.org/10.1109/TGRS.2020.3043191.
[3] X. Li, J. Deng, Y. Fang, Few-shot object detection on remote sensing images, IEEE Trans. Geosci. Remote Sens. 60 (2021) 1–14, http://dx.doi.org/10.1109/TGRS.2021.3051383.
[4] L. Lai, J. Chen, C. Zhang, Z. Zhang, G. Lin, Q. Wu, Tackling background ambiguities in multi-class few-shot point cloud semantic segmentation, Knowl.-Based Syst. 253 (2022) 109508, http://dx.doi.org/10.1016/j.knosys.2022.109508.
[5] G.S. Dhillon, P. Chaudhari, A. Ravichandran, S. Soatto, A baseline for few-shot image classification, 2019, http://dx.doi.org/10.48550/arXiv.1909.02729, arXiv preprint arXiv:1909.02729.

[6] W.-Y. Chen, Y.-C. Liu, Z. Kira, Y.-C.F. Wang, J.-B. Huang, A closer look at few-shot classification, 2019, http://dx.doi.org/10.48550/arXiv.1904.04232, arXiv preprint arXiv:1904.04232.
[7] Y. Tian, Y. Wang, D. Krishnan, J.B. Tenenbaum, P. Isola, Rethinking few-shot image classification: a good embedding is all you need? in: European Conference on Computer Vision, Springer, 2020, pp. 266–282, http://dx.doi.org/10.1007/978-3-030-58568-6_16.
[8] S. Qiao, C. Liu, W. Shen, A.L. Yuille, Few-shot image recognition by predicting parameters from activations, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2018, pp. 7229–7238, http://dx.doi.org/10.48550/arXiv.1706.03466.
[9] A.A. Rusu, D. Rao, J. Sygnowski, O. Vinyals, R. Pascanu, S. Osindero, R. Hadsell, Meta-learning with latent embedding optimization, in: International Conference on Learning Representations, 2018, http://dx.doi.org/10.48550/arXiv.1807.05960.
[10] T. Scott, K. Ridgeway, M.C. Mozer, Adapted deep embeddings: A synthesis of methods for k-shot inductive transfer learning, Adv. Neural Inf. Process. Syst. 31 (2018) https://dl.acm.org/doi/abs/10.5555/3326943.3326951.
[11] A. Nakamura, T. Harada, Revisiting fine-tuning for few-shot learning, 2019, http://dx.doi.org/10.48550/arXiv.1910.00216, arXiv preprint arXiv:1910.00216.
[12] A. Kolesnikov, L. Beyer, X. Zhai, J. Puigcerver, J. Yung, S. Gelly, N. Houlsby, Big transfer (bit): General visual representation learning, in: European Conference on Computer Vision, Springer, 2020, pp. 491–507, http://dx.doi.org/10.1007/978-3-030-58558-7_29.
[13] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, L. Fei-Fei, Imagenet: A large-scale hierarchical image database, in: 2009 IEEE Conference on Computer Vision and Pattern Recognition, Ieee, 2009, pp. 248–255, http://dx.doi.org/10.1109/CVPR.2009.5206848.
[14] D.K. Naik, R.J. Mammone, Meta-neural networks that learn by learning, in: [Proceedings 1992] IJCNN International Joint Conference on Neural Networks, Vol. 1, IEEE, 1992, pp. 437–442, http://dx.doi.org/10.1109/IJCNN.1992.287172.
[15] C. Finn, P. Abbeel, S. Levine, Model-agnostic meta-learning for fast adaptation of deep networks, in: International Conference on Machine Learning, PMLR, 2017, pp. 1126–1135, https://dl.acm.org/doi/abs/10.5555/3305381.3305498.
[16] S. Gidaris, N. Komodakis, Dynamic few-shot visual learning without forgetting, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2018, pp. 4367–4375, https://doi.ieeecomputersociety.org/10.1109/CVPR.2018.00459.
[17] Q. Sun, Y. Liu, T.-S. Chua, B. Schiele, Meta-transfer learning for few-shot learning, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2019, pp. 403–412, http://dx.doi.org/10.1109/CVPR.2019.00049.
[18] C. Zhang, Y. Cai, G. Lin, C. Shen, Deepemd: Few-shot image classification with differentiable earth mover's distance and structured classifiers, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2020, pp. 12203–12213, http://dx.doi.org/10.1109/CVPR42600.2020.01222.
[19] Y. Chen, Z. Liu, H. Xu, T. Darrell, X. Wang, Meta-baseline: Exploring simple meta-learning for few-shot learning, in: Proceedings of the IEEE/CVF International Conference on Computer Vision, 2021, pp. 9062–9071, http://dx.doi.org/10.1109/ICCV48922.2021.00893.
[20] D. Wertheimer, L. Tang, B. Hariharan, Few-shot classification with feature map reconstruction networks, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2021, pp. 8012–8021, http://dx.doi.org/10.1109/CVPR46437.2021.00792.
[21] J. Xie, F. Long, J. Lv, Q. Wang, P. Li, Joint distribution matters: Deep brownian distance covariance for few-shot classification, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2022, pp. 7972–7981, http://dx.doi.org/10.1109/CVPR52688.2022.00781.
[22] F. Sung, Y. Yang, L. Zhang, T. Xiang, P.H. Torr, T.M. Hospedales, Learning to compare: Relation network for few-shot learning, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2018, pp. 1199–1208, http://dx.doi.org/10.1109/CVPR.2018.00131.
[23] R. Hou, H. Chang, B. Ma, S. Shan, X. Chen, Cross attention network for few-shot classification, Adv. Neural Inf. Process. Syst. 32 (2019) http://dx.doi.org/10.48550/arXiv.1910.07677.
[24] M. Ren, E. Triantafillou, S. Ravi, J. Snell, K. Swersky, J.B. Tenenbaum, H. Larochelle, R.S. Zemel, Meta-learning for semi-supervised few-shot classification, 2018, http://dx.doi.org/10.48550/arXiv.1803.00676, arXiv preprint arXiv:1803.00676.
[25] B.M. Lake, R. Salakhutdinov, J.B. Tenenbaum, Human-level concept learning through probabilistic program induction, Science 350 (6266) (2015) 1332–1338, http://dx.doi.org/10.1126/science.aab3050.
[26] C. Wah, S. Branson, P. Welinder, P. Perona, S. Belongie, The caltech-ucsd birds-200-2011 dataset, 2011, https://authors.library.caltech.edu/27452/.
[27] S.J. Pan, Q. Yang, A survey on transfer learning, IEEE Trans. Knowl. Data Eng. 22 (10) (2009) 1345–1359, http://dx.doi.org/10.1109/TKDE.2009.191.
[28] J. Snell, K. Swersky, R. Zemel, Prototypical networks for few-shot learning, in: Advances in Neural Information Processing Systems, Vol. 30, 2017, https://dl.acm.org/doi/abs/10.5555/3294996.3295163.
[29] A. Santoro, S. Bartunov, M. Botvinick, D. Wierstra, T. Lillicrap, Meta-learning with memory-augmented neural networks, in: International Conference on Machine Learning, PMLR, 2016, pp. 1842–1850, https://proceedings.mlr.press/v48/santoro16.html.
[30] N. Mishra, M. Rohaninejad, X. Chen, P. Abbeel, A simple neural attentive meta-learner, in: International Conference on Learning Representations, 2018, http://dx.doi.org/10.48550/arXiv.1707.03141.
[31] T. Munkhdalai, X. Yuan, S. Mehri, A. Trischler, Rapid adaptation with conditionally shifted neurons, in: International Conference on Machine Learning, PMLR, 2018, pp. 3664–3673, http://dx.doi.org/10.48550/arXiv.1712.09926.
[32] E. Grant, C. Finn, S. Levine, T. Darrell, T. Griffiths, Recasting gradient-based meta-learning as hierarchical bayes, in: International Conference on Learning Representations, 2018, http://dx.doi.org/10.48550/arXiv.1801.08930.
[33] K. Lee, S. Maji, A. Ravichandran, S. Soatto, Meta-learning with differentiable convex optimization, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2019, pp. 10657–10665, http://dx.doi.org/10.1109/CVPR.2019.01091.
[34] H. Zhang, P. Koniusz, S. Jian, H. Li, P.H. Torr, Rethinking class relations: Absolute-relative supervised and unsupervised few-shot learning, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2021, pp. 9432–9441, http://dx.doi.org/10.48550/arXiv.2001.03919.
[35] D. Kang, H. Kwon, J. Min, M. Cho, Relational embedding for few-shot classification, in: Proceedings of the IEEE/CVF International Conference on Computer Vision, 2021, pp. 8822–8833, http://dx.doi.org/10.1109/ICCV48922.2021.00870.
[36] Z. Yue, H. Zhang, Q. Sun, X.-S. Hua, Interventional few-shot learning, in: Advances in Neural Information Processing Systems, Vol. 33, 2020, pp. 2734–2746, https://dl.acm.org/doi/abs/10.5555/3495724.3495954.
[37] X. Chen, M. Ding, X. Wang, Y. Xin, S. Mo, Y. Wang, S. Han, P. Luo, G. Zeng, J. Wang, Context autoencoder for self-supervised representation learning, 2022, http://dx.doi.org/10.48550/arXiv.2202.03026, arXiv preprint arXiv:2202.03026.
[38] B. Kim, H. Kim, K. Kim, S. Kim, J. Kim, Learning not to learn: Training deep neural networks with biased data, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2019, pp. 9012–9020, http://dx.doi.org/10.48550/arXiv.1812.10352.
[39] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, et al., Imagenet large scale visual recognition challenge, Int. J. Comput. Vis. 115 (3) (2015) 211–252, http://dx.doi.org/10.1007/s11263-015-0816-y.
[40] E. Triantafillou, T. Zhu, V. Dumoulin, P. Lamblin, U. Evci, K. Xu, R. Goroshin, C. Gelada, K. Swersky, P.-A. Manzagol, et al., Meta-dataset: A dataset of datasets for learning to learn from few examples, 2019, http://dx.doi.org/10.48550/arXiv.1903.03096, arXiv preprint arXiv:1903.03096.
[41] S. Ravi, H. Larochelle, Optimization as a model for few-shot learning, 2016, https://openreview.net/forum?id=rJY0-Kcll.
[42] R. Ma, P. Fang, T. Drummond, M. Harandi, Adaptive Poincaré point to set distance for few-shot classification, in: Proceedings of the AAAI Conference on Artificial Intelligence, 2022, pp. 1926–1934, http://dx.doi.org/10.1609/aaai.v36i2.20087.
[43] A. Li, W. Huang, X. Lan, J. Feng, Z. Li, L. Wang, Boosting few-shot learning with adaptive margin loss, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2020, pp. 12576–12584, http://dx.doi.org/10.1109/CVPR42600.2020.01259.
[44] W. Xu, Y. Xu, H. Wang, Z. Tu, Attentional constellation nets for few-shot learning, in: International Conference on Learning Representations, 2021, https://openreview.net/forum?id=vujTf_I8Kmc.
[45] S. Qu, Y. Pan, G. Chen, T. Yao, C. Jiang, T. Mei, Modality-agnostic debiasing for single domain generalization, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2023, pp. 24142–24151, http://dx.doi.org/10.48550/arXiv.2303.07123.