Consistent Meta-Regularization for Better Meta-Knowledge in Few-Shot Learning

Pinzhuo Tian, Wenbin Li, and Yang Gao, *Member, IEEE*

***Abstract*— Recently, meta-learning provides a powerful paradigm to deal with the few-shot learning problem. However, existing meta-learning approaches ignore the prior fact that good meta-knowledge should alleviate the data inconsistency between training and test data, caused by the extremely limited data, in each few-shot learning task. Moreover, legitimately utilizing the prior understanding of meta-knowledge can lead us to design an efficient method to improve the meta-learning model. Under this circumstance, we consider the data inconsistency from the distribution perspective, making it convenient to bring in the prior fact, and propose a new consistent meta-regularization (Con-MetaReg) to help the meta-learning model learn how to reduce the data-distribution discrepancy between the training and test data. In this way, the ability of meta-knowledge on keeping the training and test data consistent is enhanced, and the performance of the meta-learning model can be further improved. The extensive analyses and experiments demonstrate that our method can indeed improve the performances of different meta-learning models in few-shot regression, classification, and fine-grained classification.**

***Index Terms*— Deep learning, few-shot learning, meta-learning, meta-regularization.**

I. Introduction

LEARNING quickly is a kind of ability of human intelligence, e.g., children can recognize objects only from a few examples. However, this poses a great challenge to the existing deep learning models, which requires large-scale annotated training data to achieve promising performance. Moreover, collecting plenty of labeled training data is laborious and time-consuming. In some real environments, it is even impossible because of the intrinsic lack of data [1]. Hence, equipping a deep model with the ability to learn new concepts from a few labeled data is meaningful for practical use. Recently, meta-learning (or learning to learn) has drawn increasing interest in the machine learning community [2]–[4]. Casting few-shot learning as a meta-learning problem provides a promising paradigm to tackle this problem [5]–[7].

Manuscript received December 4, 2020; revised April 12, 2021; accepted May 24, 2021. This work was supported in part by the Science and Technology Innovation 2030—"New Generation Artificial Intelligence" Major Project under Grant 2018AAA0100905 and in part by the Nanjing Science and Technology Innovation Project for Overseas-Educated Scholars under Grant 13006002. *(Corresponding author: Yang Gao.)*

The authors are with the State Key Laboratory for Novel Software Technology, Nanjing University, Nanjing 210023, China (e-mail: tianpinzhuo@smail.nju.edu.cn; liwenbin@nju.edu.cn; gaoy@nju.edu.cn).

Color versions of one or more figures in this article are available at https://doi.org/10.1109/TNNLS.2021.3084733.

Digital Object Identifier 10.1109/TNNLS.2021.3084733

The goal of meta-learning is to gain experience over multiple learning episodes by learning how learning algorithms perform on these tasks [8]. With this experience, meta-learning can help the model learn fast or adapt quickly to new tasks, that is to say, effective meta-knowledge is critical for meta-learning to achieve this goal. However, in few-shot learning, training data in each training episode (or task) is limited, making it difficult to comprehensively describe the real data distribution. In this sense, the training data in each few-shot learning task can be considered as a biased representation of the whole data. Thus, there exists data inconsistency between training and test data in the few-shot learning task, as shown in Fig. 1. The similar problem caused by the limited data is also discussed in semisupervised learning [9]. Regarding this fact, the oracle meta-knowledge for few-shot learning should know how to eliminate the inconsistency and help the model trained by a few training data work well on the test data.

However, few existing meta-learning approaches for few-shot learning take notice of the above prior fact that can be utilized to improve the quality of meta-knowledge. To this end, we propose a new consistent meta-regularization (Con-MetaReg) method to reinforce the ability of the meta-learning model on alleviating the data inconsistency for better general meta-knowledge. In order to achieve this purpose, we use the data-distribution discrepancy between training and test data to encapsulate the data inconsistency. From this perspective, it is convenient to bring in the prior understanding of good meta-knowledge. In particular, we suppose that if the data distributions of two datasets are similar, the learning models trained by these two datasets should be close to each other in the hypothesis space. In other words, there exists a one-to-one mapping between the dataset and the learned model. Eliminating the gap between models could implicitly make the data distributions similar. Hence, Con-MetaReg is designed to align the models trained by the training and test data in each task for eliminating the data-distribution discrepancy between them. In this way, the ability of meta-knowledge on keeping the data consistent can be improved.

To comprehensively explain our method, we analyze the effect of Con-MetaReg on learning the parameter of the meta-learner in the linear few-shot regression. The result shows that Con-MetaReg can adapt the learned parameter based on the discrepancies of the training and test data and their corresponding labels in training tasks. In experiments, we adopt three few-shot scenarios to evaluate the proposed Con-MetaReg, i.e., few-shot linear regression,

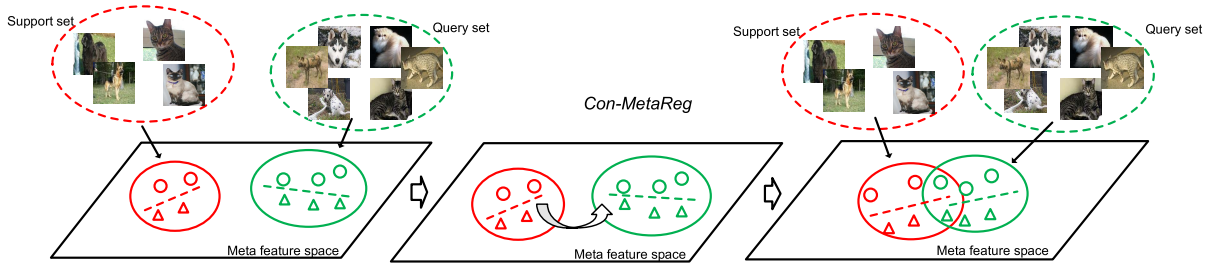

Fig. 1. Illustration of the motivation of our method. Because of the lack of annotated examples in few-shot learning, there exists data inconsistency between support and query set. For example, in this figure, although the images in support and query set belong to dog and cat, they are from different subcategories. Good meta-knowledge should alleviate this inconsistency. Thus, we propose Con-MetaReg to align the models trained by support and query set for implicitly eliminating the data discrepancy. In this way, the ability of the learned meta-knowledge on alleviating the data inconsistency can be improved.

few-shot classification, and few-shot fine-grained classification. In these scenarios, we compare the performance of some state-of-the-art meta-learning approaches integrated with Con-MetaReg or not to verify the superiority of the proposed method. The experimental results show that our method can help these meta-learning algorithms achieve better performance.

In summary, our contributions in this article are listed as follows.

1) We consider a new problem of meta-learning model when applied to few-shot learning. The learned meta-knowledge should help alleviate the data inconsistency between training and test data in each few-shot learning task. How can we bring in this prior knowledge to improve the meta-learning model?
2) We consider this problem from the data-distribution perspective and develop a novel Con-MetaReg to help the meta-learning method learn better meta-knowledge.
3) Extensive experiments highlight that with Con-MetaReg, the conventional meta-learning methods can indeed achieve better performance. We also provide an explanation to further analyze the inner mechanism of the proposed method.

II. Related Work

This article is mainly related to meta-learning and few-shot learning. Few-shot learning has been studied for decades. Recently, many approaches for few-shot learning are developed on the meta-learning framework and achieve a significant breakthrough. Hence, in this section, we briefly introduce some categories of approaches for few-shot learning besides meta-learning and detail some impressive meta-learning methods.

A learning algorithm for few-shot learning needs to rapidly generalize to new concepts with only a few labeled samples [10], [11]. Generally, the learning model needs some prior knowledge to solve this problem. Besides meta-learning-based methods, Li *et al*. [12] proposed a probabilistic model to represent objects by decomposable components, e.g., shapes and appearances of objects. The components are obtained on seen categories and expected to generalize over novel categories as the prior knowledge. The Bayesian program learning (BPL) [13] uses a generative model to represent concepts and generates a new concept hierarchically from subparts, parts, and spatial relations. These approaches need to find a large dictionary of common parts shared by all categories. However, it is usually difficult to define common parts for unconstrained objects with vast variations.

With the help of meta-learning, deep neural networks can be applied to few-shot learning and achieve great success. Meta-learning hopes to learn how machine learning approaches perform on a wide range of learning tasks and then use this experience to learn new tasks much faster than otherwise possible [14]. When the learned experience (meta-knowledge) aims to improve the performance of the learning algorithm with a few labeled training data, the meta-learning framework can be utilized to solve the few-shot learning problem. This kind of method can be broadly divided into four categories.

1) *Data Augmentation Approaches:* A straightforward way is to learn how to generate additional data to augment the number of training data. Hariharan and Girshick [15] showed that the classifiers trained on the small data can be improved by synthesizing additional training examples for data-starved classes. Following this way, Wang *et al*. [16] combined a GAN-like generator as the part of meta-learner with the aim of learning how to hallucinate additional training images for training the task-specific classifier. Chen *et al*. [17] designed a novel image deformation network based on the meta-learning framework, which learns to produce additional training samples by fusing a pair of reference images. Antoniou *et al*. [18] developed a data augmentation GAN to generate samples for improving the classifier in the low-data regime. Inspired by the metric-based few-shot learning approaches, MatchingGAN [19] uses a learned metric to generate images based on a single or a few conditional images to augment the training data. F2GAN [20] improves the quality of generated images by enhancing the fusion ability of the model. However, how to ensure the diversity of the synthesized data and contain necessary semantic details are big challenges for data augmentation approaches.
2) *Model-Based Approaches:* In these approaches, the meta-learner can be designed as a parameterized predictor to define the base learner. For example, Ravi and Larochelle [5] used a recurrent neural network as a meta-learner to direct the updating for the base learner. Munkhdalai and Yu [21] designed a meta-learner using loss gradients from base learner

to predict parameters for it. Some methods adopt an extra memory to store the past experience, which can also be regarded as a model-based algorithm. Santoro *et al*. [22] used an external memory-augmented neural network to hold the seen examples and leveraged them to make predictions with a few examples. This kind of method is usually very complex and difficult to train. It also needs extra memory.

3) *Metric-Based Approaches:* The idea of these approaches can be considered as learning to compare, and a non-parametric similarity function is used to evaluate the similarity between examples. The meta-learner is trained to learn a useful meta-representation in the predefined metric space. Vinyasls *et al*. [23] first developed an end-to-end differentiable nearest neighbor method, i.e., matching networks to perform the comparison. Snell *et al*. [24] proposed a prototypical network that represents each category by a prototype (also known as mean embedding of the examples) and utilized the Euclidean distance to measure the similarity between test images and prototypes. Sung *et al*. [25] used a neural network as the task-agnostic metric. However, metric-based meta-learning approaches are far largely restricted to the few-shot classification because it is very difficult to design an appropriate metric to measure the similarity in other situations, e.g., regression.
4) *Gradient-Based Approaches:* Gradient-based approaches employ gradient descent methods to directly adjust the parameters of the meta-learner to learn cross-task meta-knowledge. Finn *et al*. [2] proposed model-agnostic meta-learning (MAML) for deep models. MAML aims to learn a good initialization for all tasks, and the task-specific learning model of a new task can be obtained by a few gradient steps from this initialization. However, there are still many limitations of MAML. Some works [26]–[28] are developed to further improve it. Besides MAML, some gradient-based approaches aim to learn a cross-task representation as meta-knowledge, which can generalize to new classes [29], [30]. How to efficiently optimize the gradient-based approaches is a challenge for these methods, because of second-order derivatives and differentiating through the inner loop learning process.

Although there exists the problem of time efficiency in the gradient-based approach, the number of inner loop for optimizing the base learner can be small in few-shot learning. Therefore, the time of training gradient-based approach for few-shot learning is acceptable. Compared with the metric-based approach, the gradient-based approach can be broadly used in many few-shot scenarios, such as few-shot classification [30], regression [2], and object detection [31]. Moreover, the gradient-based method does not need additional parameters or requires a particular architecture. These factors make the gradient-based method a promising and hot research topic in meta-learning. Our work depends on the gradient-based approach to improve their performance in few-shot learning.

III. Preliminary

In this section, we first describe the problem definition of meta-learning in the context of few-shot learning. Then, the two-level hierarchical framework is introduced, which is popular in the current gradient-based meta-learning approach for few-shot learning. Finally, some typical gradient-based approaches following this framework are enumerated.

A. Problem Definition

Different from conventional machine learning, the training sample in meta-learning is task (episode) rather than data instance. Meta-learning algorithm needs to learn general meta-knowledge over the multiple training episodes, which can be adopted to learn new tasks.

Following previous literature [5], [14], there exists a meta-training set containing T training tasks to train the meta-learning model, i.e., $\mathcal{S}_{\text{tr}} = \{\mathcal{T}^{(1)}, \mathcal{T}^{(2)}, \ldots, \mathcal{T}^{(T)}\}$. Each task contains a training dataset (or support set) and a test dataset (or query set), i.e., $\mathcal{T}^{(i)} = \{S_{\text{tr}}^{(i)}, S_{\text{ts}}^{(i)}\}$. The number of training data in $S_{\text{tr}}^{(i)}$ is very small in few-shot learning. According to the different learning problem, the form of the training task is different. For example, in few-shot classification, learning task $\mathcal{T}^{(i)}$ can be regarded as an N-way K-shot classification task, i.e., recognizing N categories given K samples per category. To be specific, suppose that the overall set of training categories is C_{tr}, the training dataset $S_{\text{tr}}^{(i)}$ of task $\mathcal{T}^{(i)}$ contains N categories drawn from C_{tr}, and each category includes K examples, i.e., $S_{\text{tr}}^{(i)} = \{(\mathbf{x}_j^{(i)}, y_j^{(i)})\}_{j=1}^{N\times K}$. The test dataset $S_{\text{ts}}^{(i)}$ also consists of the same N categories and each category has Q examples. As for few-shot regression, the learning task $\mathcal{T}^{(i)}$ can be considered as a K-shot regression task. The training dataset $S_{\text{tr}}^{(i)}$ and the test dataset $S_{\text{ts}}^{(i)}$ consist of K training data and Q test data, respectively, i.e., $S_{\text{tr}}^{(i)} = \{(\mathbf{x}_j^{(i)}, y_j^{(i)})\}_{j=1}^{K}$ and $S_{\text{ts}}^{(i)} = \{(\mathbf{x}_j^{(i)}, y_j^{(i)})\}_{j=1}^{Q}$.

Besides the meta-training set, we also access a meta-validation set and a meta-testing set. The meta-validation set is used to choose a model during the meta-training stage. The meta-testing set is used to evaluate the learned meta-learning model. The tasks from these three sets are normally considered to follow the same task distribution τ.

B. Hierarchical Gradient-Based Method

How to learn useful meta-knowledge over $\mathcal{S}_{\text{tr}}$ is important for the meta-learning model. Current gradient-based meta-learning methods are highly based on two-level hierarchical architecture and achieve the state-of-the-art performance in many few-shot scenarios [30], [32]. This architecture can be formulated as a bilevel optimization problem [33], and we can use the episodic training paradigm [23] to train the whole model. Following [34], the two-level meta-learning framework can be defined as

$$\min_{\theta} \sum_{i=1}^{T} \mathcal{L}^{\text{meta}}\left(\theta, \omega^{*(i)}(\theta); S_{\text{ts}}^{(i)}\right) \tag{1}$$

$$\text{s.t.} \quad \omega^{*(i)}(\theta) = \arg\min_{\omega} \ell\left(\omega; \theta, S_{\text{tr}}^{(i)}\right) \tag{2}$$

where $\mathcal{L}^{\text{meta}}$ and ℓ refer to the function of meta loss (as the outer objective in bilevel optimization) and the function of task loss (as the inner objective in bilevel optimization), respectively, and $\mathcal{L}^{\text{meta}}$ and ℓ usually adopt the same loss function. In particular, the inner part (2) aims to learn a task-specific base learner for every single task with the training dataset S_{tr} in this task, whereas the upper part (1) learns meta-knowledge from how to improve these base learners with the query sets, which can be utilized to help learn unseen tasks.

Next, we introduce three typical two-level gradient-based approaches that are used as the benchmark methods in our experiments.

MAML: MAML is a significant gradient-based meta-learning method, which has been applied to many fields [35]–[37]. The initialization of the deep model is regarded as meta-knowledge in MAML. Thus, the goal of MAML is to meta-learn the initial model parameter θ, which could generalize over the task distribution. The loss function of MAML can be written as

$$\min_{\theta} \frac{1}{T} \sum_{i=1}^{T} \mathcal{L}\left(w^{(i)}(\theta); S_{\text{ts}}^{(i)}\right) \tag{3}$$

$$\text{s.t.} \quad w^{(i)}(\theta) = \theta - \alpha \nabla_{\theta} \mathcal{L}\left(\theta; S_{\text{tr}}^{(i)}\right) \tag{4}$$

where α is the step size and $\mathcal{L}$ is the loss function, e.g., cross-entropy loss in few-shot classification. $\theta - \alpha \nabla_{\theta} \mathcal{L}(\theta; S_{\text{tr}}^{(i)})$ means one step of inner updating and aims to obtain a task-specific learning model (base learner) for task $\mathcal{T}^{(i)}$. When encountering a new task $\mathcal{T}^{(j)}$, the task-specific predictor can be easily obtained in a single (or a few) inner gradient step from the initial θ. In fact, MAML is a special case of the bilevel gradient-based meta-learning method, which is analyzed in [27]. However, there exists a problem of calculating second-order derivative and storing Hessian matrix when we optimize MAML. To solve them, we can just update top layers in the inner loop instead of the whole model when MAML is applied to deep networks. Raghu *et al*. [38] implemented this idea by ANIL and showed that ANIL can achieve the same performance compared with the MAML.

MetaOptNet and R2D2: MetaOptNet [30] and R2D2 [29] use support vector machine [39] and ridge regression as the base learner in (2), respectively. Both of them want to learn a cross-task meta-representation by (1), which can help the base learner of the new task learn from a few training data.

IV. Method

As mentioned above, high-quality meta-knowledge, which can generalize well over the task distribution τ, is very important for the meta-learning algorithm to improve the performance of the base learner in the low-data regime. Considering that the number of training data in each few-shot task is $1 \sim 25$ and too limited data certainly lead to a biased representation of the whole dataset, e.g., a few training data from one category just contains a small part of the variations, and there are still many other unseen variations in this category, high-quality meta-knowledge in the few-shot setting should overcome this problem and make the model trained on the small training dataset work well on the test data, which may contain other unseen variations. Although the gradient-based method indeed achieves success in few-shot learning, most of them ignore the fact that we can further improve the meta-learning method by enhancing the ability on eliminating the effect of the biased representation.

According to this consideration, we think about this problem from a data-distribution perspective. The distribution of the training data in a few-shot learning task can be regarded as following a biased data-sampling distribution due to the limited data. Therefore, there exists a data-distribution discrepancy between the training and test data. We hope to design a method to reinforce the ability of meta-knowledge on alleviating the discrepancy to further improve the traditional meta-learning model.

However, accurately estimating the distribution from a few data is very difficult. We propose a proxy task to achieve this goal. Specifically, we assume that the model trained by a certain dataset can represent the distribution of this dataset. If the data-distribution discrepancy of two datasets is small, the learning models of the two datasets are near to each other in the hypothesis space. Hence, instead of directly aligning the distributions, we propose a Con-MetaReg to help the meta-learning model learn how to keep the models trained by the training and test data consistent in each few-shot task. Under this constraint, the base learner trained by the training data is supposed to be close to the learning model of the test data. Moreover, the data distributions of the training and test data are implicitly aligned.

Next, we introduce our method in detail. For task $\mathcal{T}^{(i)}$, the base learner $M_{\text{tr}}^{(i)}$ trained by the support set $S_{\text{tr}}^{(i)}$ is first obtained, the same as the traditional meta-learning method. Then, we exploit the query set $S_{\text{ts}}^{(i)}$ in task $\mathcal{T}^{(i)}$ to train a new specific learning model $M_{\text{ts}}^{(i)}$. The difference between $M_{\text{tr}}^{(i)}$ and $M_{\text{ts}}^{(i)}$ can be considered as a metric to measure the data-distribution discrepancy between the support and the query set. In our method, we directly use the Frobenius norm of the difference between parameters of $M_{\text{tr}}^{(i)}$ and $M_{\text{ts}}^{(i)}$ as the meta-regularization and minimize it to help eliminate the data-distribution discrepancy for better meta-knowledge. Though the form of $M_{\text{tr}}^{(i)}$ and $M_{\text{ts}}^{(i)}$ are various in different meta-learning models, e.g., $\omega^{*(i)}(\theta)$ in (2) and $w^{(i)}(\theta)$ in (4), the proposed regularization can be easily calculated. For example, Con-MetaReg in the bilevel gradient-based method can be defined as

$$\begin{aligned} &\min_{\theta} \sum_{i=1}^{T} \mathcal{L}^{\text{meta}}\left(\theta, M_{\text{tr}}^{(i)}(\theta); S_{\text{ts}}^{(i)}\right) + \delta \left\| M_{\text{tr}}^{(i)}(\theta) - M_{\text{ts}}^{(i)}(\theta) \right\|_F \\ &\text{s.t.} \quad M_{\text{tr}}^{(i)}(\theta) = \arg\min_{\omega} \ell\left(\omega; \theta, S_{\text{tr}}^{(i)}\right) \\ &\qquad M_{\text{ts}}^{(i)}(\theta) = \arg\min_{\omega} \ell\left(\omega; \theta, S_{\text{ts}}^{(i)}\right) \end{aligned} \tag{5}$$

where δ is the regularization parameter and $\|\cdot\|_F$ is the Frobenius norm. If we consider that the base learner is a neural network, e.g., MAML or ANIL, which contains K layers. Algorithm 1 summarized the proposed Con-MetaReg in this situation.

Algorithm 1 Consistent Meta-Regularization for MAML or ANIL With Deep Networks

Require: $\mathcal{S}_{\text{tr}}$: a meta-training set, $\{\theta_k\}_{k=1}^{K}$: a deep network containing K layers, J: layers needing to be updated in the inner optimization

Require: δ: regularization parameter, α: step size of the inner optimization, β: step size of the outer optimization

Randomly initialize $\boldsymbol{\theta} = \{\theta_k\}_{k=1}^{K}$

Set $\boldsymbol{\theta}_{meta} = \{\theta_k\}_{k=1}^{J}$, $M_{\text{tr}} = \{\theta_k\}_{k=J}^{K}$, and $M_{\text{ts}} = \{\theta_k\}_{k=J}^{K}$

while not done **do**

  for task $\mathcal{T}^{(i)} = \{S_{\text{tr}}^{(i)}, S_{\text{ts}}^{(i)}\}$ in $\mathcal{S}_{\text{tr}}$ **do**

    Computer adapted parameters of M_{tr} with gradient descent:
    $M_{\text{tr}}^{(i)'} \leftarrow M_{\text{tr}} - \alpha \nabla_{M_{\text{tr}}} \mathcal{L}_{\{\boldsymbol{\theta}_{meta}, M_{\text{tr}}\}}(S_{\text{tr}}^{(i)})$

    Computer meta loss $\mathcal{L}^{meta} = \mathcal{L}_{\{\boldsymbol{\theta}_{meta}, M_{\text{tr}}^{(i)'}\}}(S_{\text{ts}}^{(i)})$

    Computer adapted parameters of M_{ts} with gradient descent:
    $M_{\text{ts}}^{(i)'} \leftarrow M_{\text{ts}} - \alpha \nabla_{M_{\text{ts}}} \mathcal{L}_{\{\boldsymbol{\theta}_{meta}, M_{\text{ts}}\}}(S_{\text{ts}}^{(i)})$

  end for

  Update
  $\boldsymbol{\theta} \leftarrow \boldsymbol{\theta} - \beta \nabla_{\boldsymbol{\theta}} (\mathcal{L}^{meta} + \delta \| M_{\text{tr}}^{(i)'} - M_{\text{ts}}^{(i)'} \|_F)$

end while

Remarks: In fact, because the label space of different tasks is different, the data distribution in each task is naturally diverse. For example, let consider the N-way K-shot classification task, and different tasks contain different N categories, causing that the data distributions in tasks vary. Therefore, a method that can precisely estimate various data distributions with a few data is required. In our method, representing the data distribution by a learning model can be adaptive to different situations, making it appropriate to solve this problem.

V. Explanation by Linear Regression

In this section, a useful analysis is given to help us gain insight into the nature of the proposed Con-MetaReg. We compare and analyze the solutions for MAML and MAML-CM in linear few-shot regression following [40], i.e., learning a good initialization of the linear regression model over multiple regression tasks. MAML-CM represents MAML integrated with Con-MetaReg. This analysis illustrates that Con-MetaReg can provide nontrivial gains by considering the differences between support and query data and their labels in each few-shot task.

First, we present some definitions for the linear regression task. Supposed that a task $\mathcal{T}^{(i)}$ contains $\mathcal{D}^{(i)} = \{(\mathbf{x}_j^{(i)}, y_j^{(i)}\}_{j=1}^{n}$, where $\mathbf{x}_j^{(i)} \in \mathbb{R}^d$, $y_j^{(i)} \in \mathbb{R}$, we consider that a linear regression model with squared loss is used to learn the function between training data and their ground truth in $\mathcal{T}^{(i)}$, such as

$$\ell^{(i)} = \frac{1}{2} \min_{\mathbf{w}} \mathbb{E}_{(\mathbf{x},y)\in\mathcal{D}^{(i)}} \|\mathbf{w}^T\mathbf{x} - y\|^2. \tag{6}$$

However, in the few-shot setting, training data are limited in each task, and we want to learn a cross-task meta-initialization of $\mathbf{w}$, which can generalize well on new linear regression tasks, by MAML. Following the definitions in Section III-A, each training task contains a support set S_{tr} and a query set S_{ts}. The optimization problem of MAML with one inner gradient updating over T tasks can be written as

$$\min_{\mathbf{w}} \frac{1}{T} \sum_{i=1}^{T} \mathcal{L}^{\text{meta}}\left(U_{\text{tr}}^{(i)}(\mathbf{w}); S_{\text{ts}}^{(i)}\right) \quad \text{where } U_{\text{tr}}^{(i)}(\mathbf{w}) = \mathbf{w} - \alpha \nabla \ell^{(i)}\left(\mathbf{w}; S_{\text{tr}}^{(i)}\right) \tag{7}$$

where $\mathcal{L}^{\text{meta}}$ is the meta loss, and we also use squared loss. $\mathbf{w}$ is the initialized cross-task weight that MAML wants to learn. $U_{\text{tr}}^{(i)}$ is the base learner of task $\mathcal{T}^{(i)}$ (one-step gradient descent updating from the meta-weight $\mathbf{w}$).

Compared with MAML, MAML-CM integrated with Con-MetaReg is formulated as

$$\min_{\mathbf{w}} \frac{1}{T} \sum_{i=1}^{T} \mathcal{L}^{\text{meta}}\left(U_{\text{tr}}^{(i)}(\mathbf{w}); S_{\text{ts}}^{(i)}\right) + \frac{1}{2\alpha} \left\| U_{\text{ts}}^{(i)}(\mathbf{w}) - U_{\text{tr}}^{(i)}(\mathbf{w}) \right\|_F^2 \tag{8}$$

where

$$U_{\text{tr}}^{(i)}(\mathbf{w}) = \mathbf{w} - \alpha \nabla \ell^{(i)}\left(\mathbf{w}; S_{\text{tr}}^{(i)}\right)$$
$$U_{\text{ts}}^{(i)}(\mathbf{w}) = \mathbf{w} - \alpha \nabla \ell^{(i)}\left(\mathbf{w}; S_{\text{ts}}^{(i)}\right)$$

with $U_{\text{ts}}^{(i)}$ the specific learning model for the query set $S_{\text{ts}}^{(i)}$.

In our analysis, we denote $\mathbf{X}^{(i)} = [\mathbf{x}_1^{(i)}, \ldots, \mathbf{x}_n^{(i)}]^T, \mathbf{X}^{(i)} \in \mathbb{R}^{n\times d}$ and $\mathbf{y}^{(i)} = [y_1^{(i)}, \ldots, y_n^{(i)}]^T, \mathbf{y}^{(i)} \in \mathbb{R}^n$ and use the matrix form to represent the linear regression task with squared loss. Specifically, considering a collection of objective functions: $\{f_i : \mathbf{w} \in \mathbb{R}^d \to \mathbb{R}\}_{i=1}^{T}$

$$f_i(\mathbf{w}) = \frac{1}{2}\mathbf{w}^T\mathbf{A}^{(i)}\mathbf{w} + \mathbf{w}^T\mathbf{b}^{(i)} + c^{(i)} \tag{9}$$

each function $f_i(\mathbf{w})$ can be regarded as a linear regression task in (6), corresponding to $\mathbf{A}^{(i)} = \mathbf{X}^{(i)^T}\mathbf{X}^{(i)}$, $\mathbf{b}^{(i)} = -\mathbf{X}^{(i)^T}\mathbf{y}^{(i)}$, $c^{(i)} = \mathbf{y}^{(i)^T}\mathbf{y}^{(i)}$. Next, we study the difference between MAML and MAML-CM.

MAML: We first show the learned meta-weight in MAML by solving the optimization problem in (7). The exact form of $U_{\text{tr}}^{(i)}$ can be obtained as

$$U_{\text{tr}}^{(i)} = \mathbf{w} - \alpha\mathbf{A}_{\text{tr}}^{(i)}\mathbf{w} - \alpha\mathbf{b}_{\text{tr}}^{(i)}$$

and in the case of the quadratic objective, this leads to

$$\begin{aligned}\mathcal{L}^{\text{meta}}&\left(U_{\text{tr}}^{(i)}(\mathbf{w}); S_{\text{ts}}^{(i)}\right) \\ &= \frac{1}{2}\left(\mathbf{w} - \alpha\mathbf{A}_{\text{tr}}^{(i)}\mathbf{w} - \alpha\mathbf{b}_{\text{tr}}^{(i)}\right)^T \mathbf{A}_{\text{ts}}^{(i)}\left(\mathbf{w} - \alpha\mathbf{A}_{\text{tr}}^{(i)}\mathbf{w} - \alpha\mathbf{b}_{\text{tr}}^{(i)}\right) \\ &\quad + \left(\mathbf{w} - \alpha\mathbf{A}_{\text{tr}}^{(i)}\mathbf{w} - \alpha\mathbf{b}_{\text{tr}}^{(i)}\right)^T \mathbf{b}_{\text{ts}}^{(i)} + c_{\text{ts}}^{(i)}.\end{aligned}$$

The corresponding gradient can be written as

$$\begin{aligned}\nabla\mathcal{L}^{\text{meta}}&\left(U_{\text{tr}}^{(i)}(\mathbf{w}); S_{\text{ts}}^{(i)}\right) \\ &= \left(I - \alpha\mathbf{A}_{\text{tr}}^{(i)}\right)\mathbf{A}_{\text{ts}}^{(i)}\left(I - \alpha\mathbf{A}_{\text{tr}}^{(i)}\right)\mathbf{w} + \left(I - \alpha\mathbf{A}_{\text{tr}}^{(i)}\right)^2\mathbf{b}_{\text{ts}}^{(i)}.\end{aligned}$$

For notational convenience, we define

$$\mathbf{A}_{\dagger} := \frac{1}{T}\sum_{i=1}^{T}\left(\mathbf{I}-\alpha\mathbf{A}_{\mathrm{tr}}^{(i)}\right)^{2}\mathbf{A}_{\mathrm{ts}}^{(i)}$$

$$\mathbf{b}_{\dagger} := \frac{1}{T}\sum_{i=1}^{T}\left(\mathbf{I}-\alpha\mathbf{A}_{\mathrm{tr}}^{(i)}\right)^{2}\mathbf{b}_{\mathrm{ts}}^{(i)}.$$

Finally, the solution to the optimization problem of MAML (7) is given: $\mathbf{w}^{*}_{\mathrm{MAML}} = -\mathbf{A}_{\dagger}^{-1}\mathbf{b}_{\dagger}$.

MAML-CM: We define the overall meta loss of MAML-CM as $\mathcal{L}_{\mathrm{cm}}^{\mathrm{meta}}$. According to the results of MAML, we can easily get the gradient of $\mathcal{L}_{\mathrm{cm}}^{\mathrm{meta}}$ with respect to $\mathbf{w}$

$$\begin{aligned}
&\nabla\mathcal{L}_{\mathrm{cm}}^{\mathrm{meta}}\left(U_{\mathrm{tr}}^{(i)}(\mathbf{w}), U_{\mathrm{ts}}^{(i)}(\mathbf{w}); S_{\mathrm{ts}}^{(i)}\right)\\
&= \nabla_{\mathbf{w}}\Bigg[\frac{1}{2}\left(\mathbf{w}-\alpha\mathbf{A}_{\mathrm{tr}}^{(i)}\mathbf{w}-\alpha\mathbf{b}_{\mathrm{tr}}^{(i)}\right)^{T}\mathbf{A}_{\mathrm{ts}}^{(i)}\left(\mathbf{w}-\alpha\mathbf{A}_{\mathrm{tr}}^{(i)}\mathbf{w}-\alpha\mathbf{b}_{\mathrm{tr}}^{(i)}\right)\\
&\qquad+\left(\mathbf{w}-\alpha\mathbf{A}_{\mathrm{tr}}^{(i)}\mathbf{w}-\alpha\mathbf{b}_{\mathrm{tr}}^{(i)}\right)^{T}\mathbf{b}_{\mathrm{ts}}^{(i)}\\
&\qquad+\frac{1}{2}\left\|\left(\mathbf{A}_{\mathrm{ts}}^{(i)}-\mathbf{A}_{\mathrm{tr}}^{(i)}\right)\mathbf{w}+\left(\mathbf{b}_{\mathrm{ts}}^{(i)}-\mathbf{b}_{\mathrm{tr}}^{(i)}\right)\right\|_{F}^{2}\Bigg]\\
&= \left(\mathbf{I}-\alpha\mathbf{A}_{\mathrm{tr}}^{(i)}\right)\mathbf{A}_{\mathrm{ts}}^{(i)}\left(\mathbf{I}-\alpha\mathbf{A}_{\mathrm{tr}}^{(i)}\right)\mathbf{w}+\left(I-\alpha\mathbf{A}_{\mathrm{tr}}^{(i)}\right)^{2}\mathbf{b}_{\mathrm{ts}}^{(i)}\\
&\quad+\left[\left(\mathbf{A}_{\mathrm{ts}}^{(i)}-\mathbf{A}_{\mathrm{tr}}^{(i)}\right)^{2}\mathbf{w}+\left(\mathbf{A}_{\mathrm{ts}}^{(i)}-\mathbf{A}_{\mathrm{tr}}^{(i)}\right)\left(\mathbf{b}_{\mathrm{ts}}^{(i)}-\mathbf{b}_{\mathrm{tr}}^{(i)}\right)\right]\\
&= \left[\left(\mathbf{I}-\alpha\mathbf{A}_{\mathrm{tr}}^{(i)}\right)^{2}\mathbf{A}_{\mathrm{ts}}^{(i)}+\left(\mathbf{A}_{\mathrm{ts}}^{(i)}-\mathbf{A}_{\mathrm{tr}}^{(i)}\right)^{2}\right]\mathbf{w}\\
&\quad+\left[\left(\mathbf{I}-\alpha\mathbf{A}_{\mathrm{tr}}^{(i)}\right)^{2}\mathbf{b}_{\mathrm{ts}}^{(i)}+\left(\mathbf{A}_{\mathrm{ts}}^{(i)}-\mathbf{A}_{\mathrm{tr}}^{(i)}\right)\left(\mathbf{b}_{\mathrm{ts}}^{(i)}-\mathbf{b}_{\mathrm{tr}}^{(i)}\right)\right].
\end{aligned}$$

For notational convenience, we define

$$\mathbf{A}_{\ddagger} := \frac{1}{T}\sum_{i=1}^{T}\left[\left(\mathbf{I}-\alpha\mathbf{A}_{\mathrm{tr}}^{(i)}\right)^{2}\mathbf{A}_{\mathrm{ts}}^{(i)}+\left(\mathbf{A}_{\mathrm{ts}}^{(i)}-\mathbf{A}_{\mathrm{tr}}^{(i)}\right)^{2}\right]$$

$$\mathbf{b}_{\ddagger} := \frac{1}{T}\sum_{i=1}^{T}\left[\left(\mathbf{I}-\alpha\mathbf{A}_{\mathrm{tr}}^{(i)}\right)^{2}\mathbf{b}_{\mathrm{ts}}^{(i)}+\left(\mathbf{A}_{\mathrm{ts}}^{(i)}-\mathbf{A}_{\mathrm{tr}}^{(i)}\right)\left(\mathbf{b}_{\mathrm{ts}}^{(i)}-\mathbf{b}_{\mathrm{tr}}^{(i)}\right)\right].$$

Finally, the solution to optimization problem of MAML-CM in (8) is given: $\mathbf{w}^{*}_{\text{MAML-CM}} = -\mathbf{A}_{\ddagger}^{-1}\mathbf{b}_{\ddagger}$.

Remarks: Although this setting is simple and restrictive, it can also explain some insights into our method. In general, $\mathbf{w}^{*}_{\mathrm{MAML}} \neq \mathbf{w}^{*}_{\text{MAML-CM}}$ based on our analysis. Next, we point out the difference between them. Let us consider that there exists a discrepancy between the training and test data in the few-shot task; however, there exists a mapping matrix between them, i.e., $\mathbf{X}_{\mathrm{ts}}^{(i)} = \mathbf{M}^{(i)}\mathbf{X}_{\mathrm{tr}}^{(i)}$.

Then, we can compare the solutions of these two methods more carefully. First, we pay attention to

$$\mathbf{A}_{\dagger} := \frac{1}{T}\sum_{i=1}^{T}\left(\mathbf{I}-\alpha\mathbf{A}_{\mathrm{tr}}^{(i)}\right)^{2}\mathbf{A}_{\mathrm{ts}}^{(i)}$$

$$\mathbf{A}_{\ddagger} := \frac{1}{T}\sum_{i=1}^{T}\left[\left(\mathbf{I}-\alpha\mathbf{A}_{\mathrm{tr}}^{(i)}\right)^{2}\mathbf{A}_{\mathrm{ts}}^{(i)}+\left(\mathbf{X}_{\mathrm{tr}}^{(i)^{T}}\left(\mathbf{M}^{(i)^{T}}\mathbf{M}^{(i)}-\mathbf{I}\right)\mathbf{X}_{\mathrm{tr}}^{(i)}\right)^{2}\right].$$

Note that the different term between $\mathbf{A}_{\dagger}$ and $\mathbf{A}_{\ddagger}$ takes care of the difference between support and query set in each few-shot task. Depending on the difference, MAML-CM can dynamically adapt the meta-initialization $\mathbf{w}^{*}_{\text{MAML-CM}}$.

TABLE I

MEAN SQUARED ERROR (MSE) OF FEW-SHOT REGRESSION, LOWER IS BETTER. ANIL-CM IS OUR METHOD

Methods	1-shot	5-shot
ANIL	7.366	1.439
ANIL + CM	7.355 (.011 ↑)	1.121 (.318 ↑)

Next, we think about

$$\mathbf{b}_{\dagger} := \frac{1}{T}\sum_{i=1}^{T}\left(\mathbf{I}-\alpha\mathbf{A}_{\mathrm{tr}}^{(i)}\right)^{2}\mathbf{b}_{\mathrm{ts}}^{(i)}$$

$$\mathbf{b}_{\ddagger} := \frac{1}{T}\sum_{i=1}^{T}\Bigg[\left(\mathbf{I}-\alpha\mathbf{A}_{\mathrm{tr}}^{(i)}\right)^{2}\mathbf{b}_{\mathrm{ts}}^{(i)} - \left(\mathbf{A}_{\mathrm{ts}}^{(i)}-\mathbf{A}_{\mathrm{tr}}^{(i)}\right)\mathbf{X}_{\mathrm{tr}}^{(i)^{T}}\left(\mathbf{M}^{(i)^{T}}\mathbf{y}_{\mathrm{ts}}^{(i)}-\mathbf{y}_{\mathrm{tr}}^{(i)}\right)\Bigg].$$

Compared with $\mathbf{b}_{\dagger}$, $\mathbf{b}_{\ddagger}$ even considers the difference between the ground truth of support and query set, apart from the discrepancy between samples.

This example and analysis reveal that there is a clear separation in solutions between MAML and MAML-CM in the case of linear few-shot regression. Con-MetaReg can improve MAML by considering the differences between training and test data and their corresponding labels in each few-shot task.

Apart from analyzing the linear setting, in our experiments, improved performance of Con-MetaReg in different meta-learning approaches is noted empirically with nonconvex loss landscapes, such as neural networks.

VI. EXPERIMENTS

In order to comprehensively investigate our method, we evaluate the proposed method in three challenging scenarios, i.e., few-shot regression, few-shot classification, and few-shot fine-grained classification. Three state-of-the-art gradient-based meta-learning methods, i.e., ANIL [38], MetaOptNet [30], and R2D2 [29], are chosen as the benchmark algorithms. Moreover, every benchmark meta-learning method is implemented based on neural networks leading to determine the performance of our method in the nonconvex loss landscapes. To accurately and clearly show the effect of the proposed meta-regularization on the meta-learning model, we remove the tricks used in benchmark methods that are adopted to improve their performance to the state of the art.

Next, we describe some details about datasets and experimental settings. In each scenario, we show the performances of the benchmark algorithms integrated with Con-MetaReg or not to verify the superiority of our method. Adopting different datasets in our experiment can also evaluate the performance of our method in different settings of data inconsistency.

A. Few-Shot Regression

1) Experimental Setting: We start with a simple regression problem to illustrate the effect of Con-MetaReg in regression. Each task involves regressing from the input to the output of a linear function $f(x) = a * x + b$, where a and b of the lines are varied between tasks. The constants a and b are uniformly sampled within $[0.0, 3.0]$ and $[-9.0, 9.0]$, respectively, and

TABLE II

AVERAGE ACCURACIES OF DIFFERENT META-LEARNING METHODS WITH CON-METAREG OR NOT ON MINIIMAGENET AND TIEREDIMAGENET. ± REPRESENTS 95% CONFIDENCE INTERVAL. CM MEANS THE PROPOSED CON-METAREG

Methods	Embedding	miniImageNet 5-way		tieredImageNet 5-way	
		1-shot	5-shot	1-shot	5-shot
ANIL	64-64-64-64	47.68 ± 0.46	60.32 ± 0.44	48.59 ± 0.51	63.99 ± 0.49
ANIL + CM	64-64-64-64	47.81 ± 0.47 (0.13 ↑)	61.62 ± 0.43 (1.30 ↑)	48.05 ± 0.52	63.73 ± 0.49
MetaOptNet	64-64-64-64	42.24 ± 0.44	55.94 ± 0.42	44.93 ± 0.48	56.70 ± 0.45
MetaOptNet + CM	64-64-64-64	43.64 ± 0.43 (1.4 ↑)	57.56 ± 0.41 (1.62 ↑)	45.17 ± 0.49 (0.24 ↑)	59.10 ± 0.46 (2.4 ↑)
R2D2	64-64-64-64	43.57 ± 0.43	57.32 ± 0.42	45.53 ± 0.48	60.50 ± 0.45
R2D2 + CM	64-64-64-64	43.64 ± 0.45 (0.07 ↑)	59.19 ± 0.41 (1.87 ↑)	45.38 ± 0.49	61.76 ± 0.46 (1.26 ↑)
MetaOptNet	ResNet-12	46.35 ± 0.48	61.18 ± 0.42	47.20 ± 0.52	61.83 ± 0.46
MetaOptNet + CM	ResNet-12	52.58 ± 0.50 (6.23 ↑)	64.00 ± 0.42 (2.82 ↑)	54.07 ± 0.52 (6.87 ↑)	62.85 ± 0.46 (1.02 ↑)
R2D2	ResNet-12	49.84 ± 0.48	65.33 ± 0.43	51.86 ± 0.52	66.15 ± 0.46
R2D2 + CM	ResNet-12	53.68 ± 0.50 (3.84 ↑)	66.88 ± 0.42 (1.55 ↑)	54.41 ± 0.53 (2.55 ↑)	68.23 ± 0.47 (2.08 ↑)

data point x is sampled uniformly from $[-5.0, 5.0]$, during the training and testing. The loss is the mean-squared error between the prediction $f(x)$ and the true value. The regressor is a neural network model with two hidden layers of size 40 with ReLU nonlinearities. Since R2D2 and MetaOptNet are designed for classification, we just use ANIL as the benchmark method. Two fully connected layers with 40 and 20 hidden units are used as the classification head in ANIL. All the models are trained by 3000 iterations, and each iteration contains ten tasks. The number of test (or query) data of each training task is ten. We use Adam [41] as the meta-optimizer. The meta-learning rate is 0.001. The inner learning rate is 0.01 and the inner updating step is three. The value of the regularization parameter δ is one.

2) Experimental Result: During the meta-test stage, 2000 new tasks are randomly sampled. In each task, 100 data points are sampled from $[-5.0, 5.0]$ by equal distance as the test dataset (or query set) S_{ts}. Table I shows the average performance of ANIL and ANIL-CM over the sampled 2000 new tasks in one- and five-shot settings. ANIL-CM means ANIL integrated with Con-MetaReg. Compared with ANIL, ANIL-CM can achieve better performance. Results verify that our method can work well in the regression problem.

B. Few-Shot Classification on ImageNet Derivatives

1) Dataset: We evaluate our method on two derivatives of the ImageNet dataset [42], i.e., miniImageNet [23] and tieredImageNet [43].

1) MiniImageNet is a standard benchmark for few-shot image classification, consisting of 100 randomly chosen classes from ILSVRC-2012 [44]. These classes are randomly split into 64, 16, and 20 classes for meta-training, meta-validation, and meta-testing, respectively. Each class contains 600 images.
2) TieredImageNet benchmark is a larger subset of ImageNet, composed of 608 classes grouped into 34 high-level categories. These are divided into 20 categories for meta-training, 6 categories for meta-validation, and eight categories for meta-testing. This corresponds to 351, 97, and 160 classes for meta-training, meta-validation, and meta-testing, respectively. Categories in three splits are totally different, making this dataset more challenging.

2) Experimental Setting: First, we introduce the model configurations. The same four-layer ConvNet in [24] are used as a kind of embedding model, which has four modules with a 3×3 convolution with 64 filters, followed by a batch normalization, a ReLU nonlinearity, and a 2×2 max pooling. Similar to [45], ResNet-12 is also used as a kind of embedding model to show the effect of the deeper embedding model on the proposed method. There are four residual blocks in ResNet-12 with 64, 128, 256, and 512 filters, and each block consists of three {3×3 convolution with k filters, batch normalization, ReLU} followed a 2×2 max-pooling layer. In ANIL, we adopt two fully connected layers containing 800 and the number of classes hidden units, respectively, as the classification head.

All the images are resized to 84×84. Adam with a learning rate of 0.001 is used as the meta-optimizer for all the methods. The inner learning rate of ANIL is 0.01, and the step of inner gradient descent is five. All the models are trained by 30 000 iterations on miniImagNet and 60 000 iterations on tieredImageNet, and each iteration includes one training task. The number of test data in each training task is ten. The regularization parameter δ in ANIL-CM is one, and the regularization parameter δ in R2D2-CM and MetaOptNet-CM is 5.

3) Experimental Result: We insert Con-MetaReg into ANIL, MetaOptNet, and R2D2 to validate its effectiveness. Table II reports the results on the five-way few-shot classification on miniImageNet and tieredImageNet. All the reported results are averaged over 2000 tasks randomly sampled from the meta-testing set. Each task contains ten queries of per category.

MiniImageNet: First, we pay attention to the results on miniImageNet. As seen, by mitigating the data-distribution discrepancy, Con-MetaReg can improve the performance of ANIL, MetaOptNet, and R2D2 on miniImageNet in five-way one-shot and five-shot settings with ConvNet and Resnet-12. When we use ConvNet as the embedding model, the effect of Con-MetaReg is more obvious in the five-way five-shot than five-way one-shot setting. Although deeper embedding model, i.e., ResNet-12, can extra more effective meta-representation to

TABLE III

AVERAGE ACCURACIES OF DIFFERENT META-LEARNING METHODS WITH CON-METAREG OR NOT ON OMNIGLOT. ± REPRESENTS 95% CONFIDENCE INTERVAL

Methods	20-way	
	1-shot	5-shot
ANIL	79.87 ± 0.29	90.38 ± 0.22
ANIL + CM	85.63 ± 0.25 (5.76 ↑)	92.40 ± 0.19 (2.02 ↑)
R2D2	85.01 ± 0.22	94.94 ± 0.09
R2D2 + CM	87.46 ± 0.20 (2.35 ↑)	95.09 ± 0.09 (0.15 ↑)

help baseline models achieve higher accuracies than ConvNet, Con-MetaReg can also effectively improve the performance of three baseline models, that is to say, merely adopting a powerful embedding model cannot completely eliminate the data inconsistency between the training and test data.

We observe that Con-MetaReg provides a higher increase of the accuracy with ResNet-12 than ConvNet in the one-shot setting. This might be due to that the ability of the representation of ConvNet is limited. Hence, it is challenging for the linear base learner to represent the data distribution, especially in the one-shot setting (extreme lack of training data to learn a good base learner), suppressing Con-MetaReg to alleviate the data inconsistency.

TieredImageNet: Similar phenomena appear on tieredImageNet. When ResNet-12 is used as the embedding model, the effect of our method is obvious in the five-way one-shot and five-shot settings, especially in the one-shot setting. However, with ConvNet, ANIL-CM does not outperform ANIL, and similar results also occur in R2D2 and R2D2-CM in the one-shot setting, yet the performance of Con-MetaReg is competitive. This result might due to that tieredImageNet is more challenging than miniImageNet and the capacity of ConvNet is limited, accurately representing the data distribution by the base learner with ConvNet is more difficult than miniImageNet. When ResNet-12 is adopted as the embedding model, the superiority of Con-MetaReg appears.

C. Few-Shot Classification on Omniglot

1) Dataset: Omniglot [46] is a dataset containing 20 instances of 1623 handwritten characters from 50 alphabets. Each instance is drawn by a different human subject. We follow the procedure in [23] by resizing the grayscale images to 28×28 and augmenting the character classes with rotations in multiples of 90°. The same 1200 characters as in [23] are selected for training and the remaining classes for testing.

2) Experimental Setting: We conduct the experiments on the 20-way classification setting. The same four-layer ConvNet in our experiments on ImageNet derivatives is used as the embedding model. Because of the time consumption for optimizing support vector machine in high-way setting, ANIL and R2D2 are adopted as the baseline models to verify the superiority of our method. All the models are trained by 30 000 tasks via Adam with a learning rate of 0.001. The number of the test data in each training task is ten. The inner learning rate of ANIL is 0.1, and the step of inner gradient descent is five.

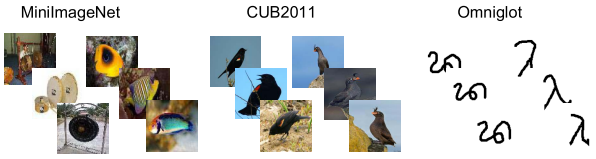

Fig. 2. Illustration to show images from different datasets. The images in the same column belong to the same category.

TABLE IV

AVERAGE ACCURACIES OF DIFFERENT META-LEARNING METHODS WITH CON-METAREG OR NOT ON CUB2011. ± REPRESENTS 95% CONFIDENCE INTERVAL

Methods	5-way	
	1-shot	5-shot
ANIL	59.24 ± 0.55	72.70 ± 0.46
ANIL + CM	59.89 ± 0.54 (0.65 ↑)	74.35 ± 0.45 (1.65 ↑)
MetaOptNet	47.41 ± 0.47	66.86 ± 0.42
MetaOptNet + CM	51.83 ± 0.50 (4.42 ↑)	67.57 ± 0.41 (0.71 ↑)
R2D2	48.70 ± 0.47	68.52 ± 0.40
R2D2 + CM	51.23 ± 0.49 (2.53 ↑)	70.05 ± 0.40 (1.53 ↑)

3) Experimental Result: We report the results in Table III. All the results are averaged over 2000 new tasks, and per class in the query set of each task consists of 15 query images. The results show that our method can also improve the performance of different meta-learning methods in the high-way setting.

D. Few-Shot Fine-Grained Classification

1) Dataset: For few-shot fine-grained classification, we use the fine-grained image classification benchmark CUB-200-2011 [47] (referred to as CUB2011 hereafter). This dataset contains 200 classes and 11 788 images in total. We follow the same class split proposed in [48].

2) Experimental Setting: We use ConvNet in Section VI-B as the embedding model. The same image size is adopted. Adam with the same learning rate in few-shot classification on ImageNet derivatives is also used to optimize ANIL, R2D2, and MetaOptNet. All the models are trained by 30 000 iterations, and each iteration includes one training task. The number of the test data in each training task is also ten. The regularization parameter δ of ANIL-CM is 0.1 and 5 for R2D2-CM and MetaOptNet-CM.

3) Experimental Result: Table IV shows the results on CUB2011. All the results are averaged over 2000 new tasks, and each task contains 15 query images per category. The proposed Con-MetaReg can improve the performance of all the benchmark meta-learning methods in five-way one-shot and five-shot settings.

There exist various data inconsistencies in these four classification datasets. Compared with miniImageNet and tieredImageNet, CUB2011 and Omniglot contain smaller intraclass differences, where the images in support and query set are much similar. The experiments on different datasets prove that the proposed method can work well in different scenarios. Fig. 2 summarizes the differences between three datasets.

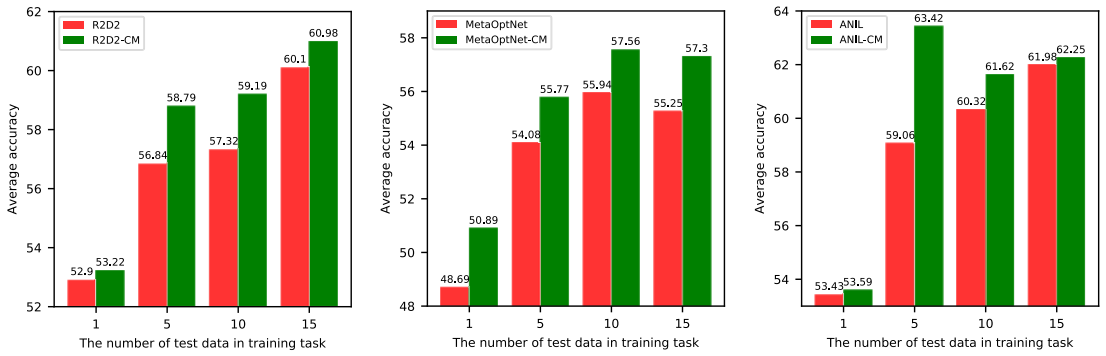

Fig. 3. Average accuracy of five-way five-shot classification task on miniImageNet. The horizontal axis represents the different numbers of the query data in each training task during the meta-training stage.

VII. Discussion

In this section, we discuss and answer the following questions: how do the number of the query data and the classification way influence our method? when does Con-MetaReg help improve the performance in the learning process? what is the effect of different tricks used in meta-learning methods on Con-MetaReg? how sensitive is the performance of Con-MetaReg to regularization parameter δ? can Con-MetaReg be easily optimized? Discussing these problems gives a comprehensive understanding of Con-MetaReg.

A. Influence of the Number of the Query Data

According to (5), the performance of our method is influenced by whether the learning model can accurately represent the data distribution or not. As we know, the number of the query data can be changed in the meta-training stage, and providing more query data means that we can obtain a learning model to accurately estimate the data distribution of the query data. Thus, we investigate the influence of the number of query data on our method.

We conduct the experiments on five-way five-shot classification. The number of the query data in each training task is set $1, 5, 10$, and 15. Fig. 3 shows the results of three benchmark meta-learning methods with Con-MetaReg or not. All the results are averaged over 2000 tasks, and ConvNet is used as the embedding model. As shown in Fig. 3, our method outperforms the corresponding benchmark method with different numbers of the query data, even with just one query data. The experimental results on five-shot linear regression show the same conclusion.

Note that in the traditional meta-learning methods, $S_{\text{ts}}^{(i)}$ is only used to optimize the meta-learner. Increasing the number of the query data can help learn better meta-knowledge, which is also proved in [49]. The results of three baseline methods trained by different numbers of the query data also confirm this conclusion. Similar to the traditional meta-learning methods, our method also generally follows this rule.

B. Influence of the Different Training Ways

Compared with five-way classification, higher way, e.g., ten ways, classification brings a challenge for our method because more base learners need to be aligned. In this section, we show the experimental results on the ten-way five-shot classification to determine the performance of Con-MetaReg in such a challenging scenario.

TABLE V

Results of Ten-Way Classification on miniImageNet. These Results Are Used to Show the Performance of Our Method on Different Ways

Methods	*Con-MetaReg*	5-shot Acc.
ANIL		46.78 ± 0.28
	$\surd$	47.25 ± 0.28
MetaOptNet		40.90 ± 0.27
	$\surd$	42.01 ± 0.27
R2D2		38.02 ± 0.26
	$\surd$	41.66 ± 0.26

Table V shows the results averaged over 2000 new tasks on miniImageNet. The same experimental settings in Section VI-B are followed. We can observe that with Con-MetaReg, the performance of different baselines can be improved.

C. Influence of Con-MetaReg on the Learning Process

We show the classification accuracies of the baseline models with our method or not in the different training epochs to exhibit how Con-MetaReg improves the performance during the learning process.

All the results are averaged on 2000 new tasks. As shown in Fig. 4, we can observe that with our method, R2D2-CM and MetaOptNet-CM achieve higher accuracy at the beginning of the training and take the lead until convergence. Although the superiority of ANIL-CM is not obvious at the beginning, the advantage of our method stands out after the 15th epoch. The analogous conclusion can be found on CUB2011, as shown in Fig. 5.

D. Influence of the Different Tricks

In order to achieve state-of-the-art performance, different meta-learning models adopt some customized tricks. For example, MetaOptNet and R2D2 use the learnable scale factor to adjust the prediction score predicted by the base learner, which is widely used in few-shot classification [50], [51]. ANIL and MAML utilize multiple tasks to train the meta-learner. However, in our experiments, the tricks in baseline methods are removed to accurately and clearly show the effect

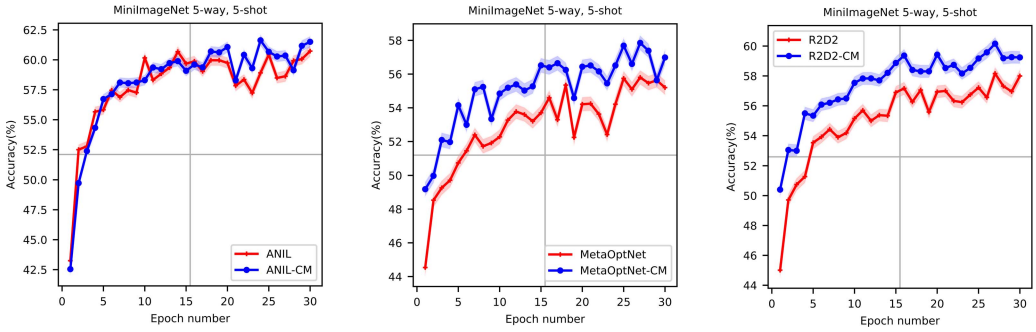

Fig. 4. Classification accuracies on 2000 test tasks in different training epochs on MiniImageNet. The shaded region denotes the 95% confidence interval.

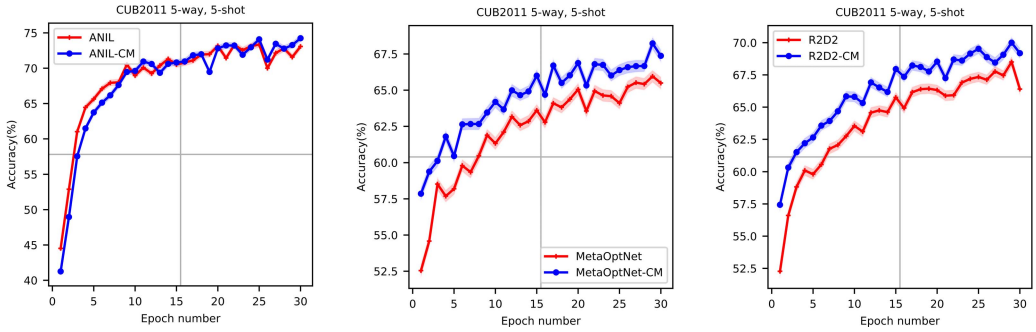

Fig. 5. Classification accuracies on 2000 test tasks in different training epochs on CUB2011. The shaded region denotes the 95% confidence interval.

TABLE VI

RESULTS OF FIVE-WAY FIVE-SHOT CLASSIFICATION ON MINIIMAGENET. THESE RESULTS ARE USED TO SHOW THE INFLUENCE OF DIFFERENT TRICKS ON OUR METHOD

Methods	Scale factor	*Con-MetaReg*	Acc.
MetaOptNet			55.94 ± 0.42
	√		64.82 ± 0.40
	√	√	65.27 ± 0.40
R2D2			57.32 ± 0.42
	√		65.18 ± 0.40
	√	√	65.66 ± 0.40
Methods	**Multi task**	***Con-MetaReg***	**Acc.**
ANIL			60.32 ± 0.44
	√		62.28 ± 0.45
	√	√	62.56 ± 0.45

Fig. 6. Some analysis results in few-shot linear regression. (a) Influence of the number of test data on Con-MetaReg. (b) Impact of the regularization parameter on our method.

of our method. In a sense, Con-MetaReg can be regarded as a kind of trick that is model-agnostic and can be applied to different learning problems.

Considering that with the customized tricks, R2D2, MetaOptNet, and ANIL had achieved state-of-the-art performance, and in this section, we investigate that whether our method can further improve the performance. The results are shown in Table VI. Inserting the customized tricks into baseline models indeed improves the performance. Our method can still work in such a difficult setting, although the increasing accuracy is lower than the results in Table II. Note that the scale factor can largely improve the performance of meta-learning models in the classification problem, while it tailors for classification. We also determine the effect of the scale factor in the regression problem. The results in Table VII show that compared with our method, the superiority of the scale factor is not obvious.

E. Influence of the Regularization Parameter

To evaluate the influence of the regularization parameter δ on our method, we train ANIL-CM and R2D2-CM with different values of the regularization parameter δ on miniImageNet. Also, the performances of these models on the meta-testing set are shown in Fig. 7. The shaded region denotes the 95% confidence interval.

We can find that the performance of ANIL-CM and R2D2-CM are both influenced by the value of the regularization parameter. Contrasted to the results without Con-MetaReg ($\delta = 0$), the classification accuracies increase when Con-MetaReg is inserted into R2D2 and ANIL. However, with increasing the value of the regularization parameter, the performances of ANIL-CM and R2D2-CM appear downtrend at $\delta = 1.6$ and $\delta = 4.0$, respectively. Fig. 7(b) shows the influence of the regularization parameter on our method

TABLE VII
EFFECT OF THE LEARNABLE SCALE FACTOR ON FEW-SHOT REGRESSION

Methods	Scale factor	*Con-MetaReg*	Acc.	
			1-shot	5-shot
ANIL			7.366	1.439
	√		7.397	1.402
		√	7.355	1.121

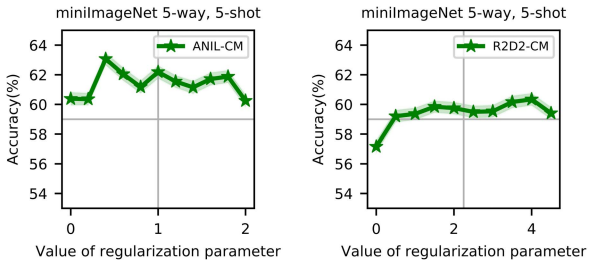

Fig. 7. Classification accuracies of ANIL-CM and R2D2-CM on miniImageNet with different regularization parameters.

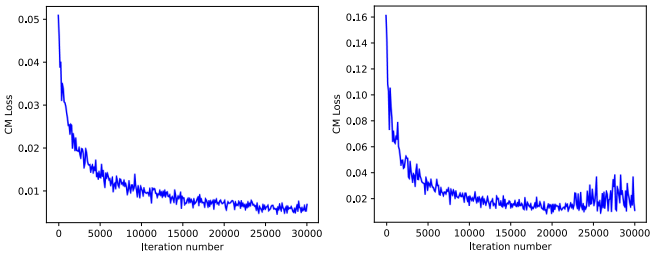

Fig. 8. Loss curves of Con-MetaReg with different embedding models. Left: R2D2-CM with ConvNet. Right: R2D2-CM with ResNet-12.

in few-shot linear regression. In the beginning, Con-MetaReg improves the performance of ANIL and, however, declines after $\delta = 1.0$.

F. Loss Curves of Con-MetaReg

In this section, we exhibit the loss curves of Con-MetaReg in the meta-training stage to investigate whether the proposed meta-regularization can be easily optimized.

Fig. 8 shows the loss curves of our method with different embedding models on miniImageNet. We can find that the values of the proposed meta-regularization keep decline whatever we use four-layer ConvNet or ResNet-12 as the feature extractor. In other words, the proposed meta-regularization is easily optimized irrespective of the architecture of the embedding model.

VIII. CONCLUSION

In this article, we take the prior understanding of the good meta-knowledge in few-shot learning into consideration, i.e., effective meta-knowledge should alleviate the data inconsistency between the training and test data in each task, caused by the limited data. Based on this fact, we propose a novel meta-regularization from the data distribution perspective to help meta-learning models learn better meta-knowledge. In our method, the learning models trained by the training and test data are used to represent their corresponding data distributions, and the Frobenius norm is adopted to align the models for implicitly alleviating the gap between the distributions. For a clear understanding, we compare the solutions of MAML and MAML-CM to demonstrate the advantages of our method. The experimental results also validate that our meta-regularization can improve the performance of different state-of-the-art meta-learning methods in various few-shot scenarios. In the future, we will explore designing a new metric method to make the proposed method more robust in different situations.

REFERENCES

[1] X. Li, L. Yu, H. Chen, C.-W. Fu, L. Xing, and P.-A. Heng, "Transformation-consistent self-ensembling model for semisupervised medical image segmentation," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 32, no. 2, pp. 1–12, Feb. 2020.

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

[3] T. Yu *et al.*, "Meta-world: A benchmark and evaluation for multi-task and meta reinforcement learning," in *Proc. Conf. Robot. Learn. (CoRL)*, 2019, pp. 1094–1100.

[4] M. Yin, G. Tucker, M. Zhou, S. Levine, and C. Finn, "Meta-learning without memorization," in *Proc. Int. Conf. Learn. Represent. (ICLR)*, 2020, pp. 1–21.

[5] S. Ravi and H. Larochelle, "Optimization as a model for few-shot learning," in *Proc. Int. Conf. Learn. Represent. (ICLR)*, 2017, pp. 1–11.

[6] N. Lai, M. Kan, C. Han, X. Song, and S. Shan, "Learning to learn adaptive classifier-predictor for few-shot learning," *IEEE Trans. Neural Netw. Learn. Syst.*, early access, Aug. 5, 2020, doi: 10.1109/TNNLS.2020.3011526.

[7] J. Bronskill, J. Gordon, J. Requeima, S. Nowozin, and R. E. Turner, "TaskNorm: Rethinking batch normalization for meta-learning," 2020, *arXiv:2003.03284*. [Online]. Available: https://arxiv.org/abs/2003.03284

[8] J. Vanschoren, "Meta-learning: A survey," 2018, *arXiv:1810.03548*. [Online]. Available: https://arxiv.org/abs/1810.03548

[9] Q. Wang, W. Li, and L. Van Gool, "Semi-supervised learning by augmented distribution alignment," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 1466–1475.

[10] Y. Wang, Q. Yao, J. T. Kwok, and L. M. Ni, "Generalizing from a few examples: A survey on few-shot learning," *ACM Comput. Surv.*, vol. 53, no. 3, p. 63, Jun. 2020.

[11] H.-G. Jung and S.-W. Lee, "Few-shot learning with geometric constraints," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 31, no. 11, pp. 4660–4672, Nov. 2020.

[12] L. Fei-Fei, R. Fergus, and P. Perona, "One-shot learning of object categories," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 28, no. 4, pp. 594–611, Apr. 2006.

[13] B. M. Lake, R. Salakhutdinov, and J. B. Tenenbaum, "Human-level concept learning through probabilistic program induction," *Science*, vol. 350, no. 6266, pp. 1332–1338, Dec. 2015.

[14] T. Hospedales, A. Antoniou, P. Micaelli, and A. Storkey, "Meta-learning in neural networks: A survey," 2020, *arXiv:2004.05439*. [Online]. Available: https://arxiv.org/abs/2004.05439

[15] B. Hariharan and R. Girshick, "Low-shot visual recognition by shrinking and hallucinating features," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2017, pp. 3037–3046.

[16] Y.-X. Wang, R. Girshick, M. Hebert, and B. Hariharan, "Low-shot learning from imaginary data," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 7278–7286.

[17] Z. Chen, Y. Fu, Y.-X. Wang, L. Ma, W. Liu, and M. Hebert, "Image deformation meta-networks for one-shot learning," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 8680–8689.

[18] A. Antoniou, A. Storkey, and H. Edwards, "Data augmentation generative adversarial networks," 2017, *arXiv:1711.04340*. [Online]. Available: https://arxiv.org/abs/1711.04340

[19] Y. Hong, L. Niu, J. Zhang, and L. Zhang, "Matchinggan: Matching-based few-shot image generation," in *Proc. IEEE Int. Conf. Multimedia Expo (ICME)*, Jul. 2020, pp. 1–6.

[20] Y. Hong, L. Niu, J. Zhang, W. Zhao, C. Fu, and L. Zhang, "F2GAN: Fusing- and-filling GAN for few-shot image generation," in *Proc. ACM Int. Conf. Multimedia (ACMMM)*, 2020, pp. 2535–2543.

[21] T. Munkhdalai and H. Yu, "Meta networks," in *Proc. 34th Int. Conf. Mach. Learn.*, vol. 70, Aug. 2017, pp. 2554–2563.

[22] A. Santoro, S. Bartunov, M. Botvinick, D. Wierstra, and T. P. Lillicrap, "Meta-learning with memory-augmented neural networks," in *Proc. Int. Conf. Mach. Learn. (ICML)*, 2016, pp. 1842–1850.

[23] O. Vinyals, C. Blundell, T. Lillicrap, K. Kavukcuoglu, and D. Wierstra, "Matching networks for one shot learning," in *Proc. Adv. Neural Inf. Process. Syst. (NIPS)*, 2016, pp. 3630–3638.

[24] J. Wang and Y. Zhai, "Prototypical siamese networks for few-shot learning," in *Proc. IEEE 10th Int. Conf. Electron. Inf. Emergency Commun. (ICEIEC)*, Jul. 2020, pp. 4077–4087.

[25] F. Sung, Y. Yang, L. Zhang, T. Xiang, P. H. S. Torr, and T. M. Hospedales, "Learning to compare: Relation network for few-shot learning," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 1199–1208.

[26] Z. Li, F. Zhou, F. Chen, and H. Li, "Meta-SGD: Learning to learn quickly for few-shot learning," 2017, *arXiv:1707.09835*. [Online]. Available: https://arxiv.org/abs/1707.09835

[27] A. Rajeswaran, C. Finn, S. M. Kakade, and S. Levine, "Meta-learning with implicit gradients," in *Proc. Adv. Neural Inf. Process. Syst. (NIPS)*, 2019, pp. 113–124.

[28] H. Lee *et al.*, "Learning to balance: Bayesian meta-learning for imbalanced and out-of-distribution tasks," in *Proc. Int. Conf. Learn. Represent. (ICLR)*, 2020, pp. 1–15.

[29] L. Bertinetto, J. F. Henriques, P. H. S. Torr, and A. Vedaldi, "Meta-learning with differentiable closed-form solvers," in *Proc. Int. Conf. Learn. Represent. (ICLR)*, 2019, pp. 1–15.

[30] K. Lee, S. Maji, A. Ravichandran, and S. Soatto, "Meta-learning with differentiable convex optimization," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 10657–10665.

[31] X. Wu, D. Sahoo, and S. Hoi, "Meta-RCNN: Meta learning for few-shot object detection," in *Proc. 28th ACM Int. Conf. Multimedia*, Oct. 2020, pp. 1679–1687.

[32] P. Tian, Z. Wu, L. Qi, L. Wang, Y. Shi, and Y. Gao, "Differentiable meta-learning model for few-shot semantic segmentation," in *Proc. AAAI Conf. Artif. Intell. (AAAI)*, Apr. 2020, pp. 12087–12094.

[33] A. Sinha, P. Malo, and K. Deb, "A review on bilevel optimization: From classical to evolutionary approaches and applications," *IEEE Trans. Evol. Comput.*, vol. 22, no. 2, pp. 276–295, Apr. 2018.

[34] L. Franceschi, P. Frasconi, S. Salzo, R. Grazzi, and M. Pontil, "Bilevel programming for hyperparameter optimization and meta-learning," in *Proc. Int. Conf. Mach. Learn. (ICML)*, vol. 80, 2018, pp. 1563–1572.

[35] M. Al-Shedivat, T. Bansal, Y. Burda, I. Sutskever, I. Mordatch, and P. Abbeel, "Continuous adaptation via meta-learning in nonstationary and competitive environments," in *Proc. Int. Conf. Learn. Represent. (ICLR)*, 2018, pp. 1–21.

[36] K. Javed and M. White, "Meta-learning representations for continual learning," in *Proc. Adv. Neural Inf. Process. Syst. (NIPS)*, 2019, pp. 1818–1828.

[37] C. Finn, "Learning to learn with gradients," Ph.D. dissertation, Dept. Elect. Eng. Comput. Sci., UC Berkeley, Berkeley, CA, USA, 2018.

[38] A. Raghu, M. Raghu, S. Bengio, and O. Vinyals, "Rapid learning or feature reuse? Towards understanding the effectiveness of MAML," in *Proc. Int. Conf. Learn. Represent. (ICLR)*, 2020, pp. 1–21.

[39] C. Cortes and V. Vapnik, "Support-vector networks," *Mach. Learn.*, vol. 20, no. 3, pp. 273–297, 1995.

[40] C. Finn, A. Rajeswaran, S. M. Kakade, and S. Levine, "Online meta-learning," in *Proc. Int. Conf. Mach. Learn. (ICML)*, vol. 97, 2019, pp. 1920–1930.

[41] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *Proc. Int. Conf. Learn. Represent. (ICLR)*, 2015, pp. 1–15.

[42] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "ImageNet: A large-scale hierarchical image database," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2009, pp. 248–255.

[43] M. Ren *et al.*, "Meta-learning for semi-supervised few-shot classification," in *Proc. Int. Conf. Learn. Represent. (ICLR)*, 2018, pp. 1–15.

[44] O. Russakovsky *et al.*, "ImageNet large scale visual recognition challenge," *Int. J. Comput. Vis.*, vol. 115, no. 3, pp. 211–252, Dec. 2015.

[45] B. N. Oreshkin, P. R. López, and A. Lacoste, "TADAM: Task dependent adaptive metric for improved few-shot learning," in *Proc. Adv. Neural Inf. Process. Syst. (NIPS)*, 2018, pp. 719–729.

[46] B. Lake, R. Salakhutdinov, J. Gross, and J. Tenenbaum, "One shot learning of simple visual concepts," in *Proc. Ann. Meeting. Conf. Cogn. Sci. Soc. (CogSci)*, vol. 33, no. 33, 2011, pp. 1–7.

[47] C. Wah, S. Branson, P. Welinder, P. Perona, and S. Belongie, "The caltech-UCSD birds-200-2011 dataset," California Inst. Technol., Pasadena, CA, USA, Tech. Rep. CNS-TR-2011-001, 2011.

[48] W. Chen, Y. Liu, Z. Kira, Y. F. Wang, and J. Huang, "A closer look at few-shot classification," in *Proc. Int. Conf. Learn. Represent. (ICLR)*, 2019, pp. 1–17.

[49] M. Balcan, M. Khodak, and A. Talwalkar, "Provable guarantees for gradient-based meta-learning," in *Proc. Int. Conf. Mach. Learn. (ICML)*, vol. 97, 2019, pp. 424–433.

[50] H. Qi, M. Brown, and D. G. Lowe, "Low-shot learning with imprinted weights," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 5822–5830.

[51] S. Gidaris and N. Komodakis, "Dynamic few-shot visual learning without forgetting," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 4367–4375.

Pinzhuo Tian is currently pursuing the Ph.D. degree with the State Key Laboratory for Novel Software Technology, Department of Computer Science Technology, Nanjing University, Nanjing, China.

His research interests lie in machine learning, including meta-learning and transfer learning.

Wenbin Li received the Ph.D. degree from the Department of Computer Science and Technology, Nanjing University, Nanjing, China, in 2019.

He is currently an Assistant Researcher with the Department of Computer Science and Technology, Nanjing University. His research interests include machine learning and computer vision, particularly in metric learning, few-shot learning, and their applications to face recognition and image classification.

Yang Gao (Member, IEEE) received the Ph.D. degree from the Department of Computer Science and Technology, Nanjing University, Nanjing, China, in 2000.

He is currently a Professor with the Department of Computer Science and Technology, Nanjing University. He has published more than 100 papers in top conferences and journals in and outside of China. His research interests include artificial intelligence and machine learning.