

# Knowledge Distillation for Model-Agnostic Meta-Learning

Min Zhang and Donglin Wang<sup>1</sup> and Sibio Gai<sup>2,3</sup>

**Abstract.** Recently, model-agnostic meta-learning (MAML) and its variants have drawn much attention in few-shot learning. In this paper, we investigate how to improve the performance of a portable MAML network so that it can be used in handheld devices, such as small robots, mobile phones, and laptops. We propose a novel approach named portable model-agnostic meta-learning (P-MAML), where valuable knowledge is distilled from a teacher MAML network to a portable student MAML. Moreover, data augmentation and ResNet architecture are employed in the teacher MAML network so as to avoid overfitting and enhance efficiency. To the best of our knowledge, this is the *first* work to consider a portable meta-learning model through knowledge distillation (KD) to learn a good initialization. Extensive experimental results on three real datasets show that our P-MAML algorithm greatly enhances the accuracy through KD from the teacher network. As shown, P-MAML with KD improves the performance of one-shot learning as high as 10% in comparison to that without KD.

## 1 INTRODUCTION

Current learning algorithms based on deep neural networks (DNNs) require a mass of data to achieve state-of-the-art performance in many classification tasks. In comparison, each human being learns new concepts and skills much faster and more efficiently. For example, children can quickly distinguish a cat from a dog just by being told a few times. Therefore, it is possible to design a machine learning model with a similar principle to learn new tasks quickly with only a couple of training examples.

For this purpose, meta-learning methods have been proposed. Meta-learning is also well-known as “learning to learn”. This is because the meta-learning algorithm aims to train a meta-learner by using some similar tasks. During the fine-tuning process, a well-trained meta-learner is capable of adapting to or generalizing new tasks and environments that have never been encountered during the training process. Tasks can be a well-defined family of machine learning problems: regression, classification, reinforcement learning and so on. For example, here are a couple of concrete meta-learning tasks:

- A classifier trained on non-cat images can judge whether a given image contains a cat after looking at several photos of a cat.
- A go-robot is able to quickly learn to play chess based on similarities in board layout and logical thinking.

- A mini robot completes the desired task on an uphill surface during the test phase even though it has been only trained in a flat-surface environment.

Model-agnostic meta-learning (MAML) with four layers, the most representative method of meta-learning, has achieved positive performance in one-shot learning [9]. However, the performance of the network with a large width, deep layers and many parameters, is generally superior to those with a small width, shallow layers and fewer parameters when trained on the same dataset [18, 25]. If we use large networks in combination with the MAML method, it can be difficult to deploy such large networks on resource-limited embedded systems due to high computational complexity. So, along with the increasing demand for low-cost networks with less computation and memory, it is essential to design a smaller network that performs as good as a relatively large network [11].

Based on the model compression algorithm [5], the knowledge distillation (KD) is proposed [8], which uses a deep neural network (a teacher network) to guide the training of shallow network (a student network). The student network outperforms an identical-layer network without distillation. We believe that if we use a teacher network to mentor the four layers of MAML for training, the performance can be further improved without introducing extra complexity. In this paper, as the first work to combine MAML with KD, we propose portable model-agnostic meta-learning (P-MAML). Experiment results show that P-MAML with small capacity can improve the recognition performance of one-shot learning.

Our main contribution can be summarized as follows

- We propose P-MAML to incorporate the power of knowledge distillation into MAML, showing that P-MAML outperforms vanilla MAML on one-shot image classification.
- Data augmentation and ResNets are considered, because MAML embedded in a deep network is easy to cause overfitting and gradient-vanishing problem.
- Extensive experiments conducted on three popular datasets: Mini-ImageNet [29], CUB [23] and Omniglot [27] demonstrate the effectiveness of our proposed P-MAML.

## 2 RELATED WORK

Our work is broadly related to three research topics: one-shot image classification, meta-learning and knowledge distillation.

### 2.1 One-shot Image Classification

In recent years, the investigation of one-shot learning (OSL) has made significant progress. As follows, we retrospect the literature of OSL methods from three perspectives: data, algorithm and model.

<sup>1</sup> Corresponding author

<sup>2</sup> School of Engineering, Westlake University, Hangzhou, 310024, China, email: {zhangmin, wangdonglin, gaisibo}@westlake.edu.cn

<sup>3</sup> Institute of Advanced Technology, Westlake Institute for Advanced Study, Hangzhou, 310024, China

Training data can be augmented by using prior knowledge of existing samples. On image classification tasks, many data augmentations are completed by transforming original training samples: 1) flipping [22] takes the vertical axis passing through the image center as the symmetry axis, and exchanges the left and right pixels; 2) rotating [1] rotates the image through  $60^\circ$ ,  $90^\circ$  and  $180^\circ$ ; 3) scaling means amplification or shortening of the training set [31]; 4) noise-addition [15] represents the random disturbance of RGB of every pixel in the image. The common noise modes include salt and pepper noise, and Gaussian noise.

Algorithms can train an excellent initial parameter through multiple related tasks. Koch et al. [17] propose a method to use the siamese neural network to do few-shot image classification. In order to output the probability of two images belonging to the same class, the siamese network uses two networks to extract features from both images and provides a ranking on the similarity score in metric space. A non-parameter matching network is proposed [29], which maps a small labeled training dataset and a new task to a correct label, adapting quickly to the new class without fine-tuning. The MAML [9] is divided into the inner loop and outer loop, where the inner loop is used to calculate the update direction of each batch of tasks, and outer loop updates the meta-parameter  $\theta$  by minimizing the output of query set. Finally, it learns a good meta-learner, which can be generalized to a new task through a few updates. And its variant in [2, 10] is also a popular meta-learning method achieving high performance.

Models can update parameters via few-shot samples by designing a model structure, and directly establishing the mapping function between the input and predicted value. Multi-task learning [6] learns multiple related tasks spontaneously using specific information of each task and generic information shared across tasks. In [16], Jia et al. propose embedding learning that transform input data into a smaller embedding space, where the pair of dissimilarity and similarity can be easily identified. Embedding learning is generally composed of task-specific method, task-invariant method and a combination of task-specific and task-invariant.

## 2.2 Meta-Learning

From a macro perspective, meta-learning methods are mainly divided into three categories: metric-based, model-based and optimization-based [9, 27]. This paper focuses on the optimization-based meta-learning method, which uses few-shot labeled samples in tasks to update a meta-learner.

By focusing on the distribution of each category in the whole task space, meta-learning can obtain prior knowledge and quickly adapt to new tasks, so that it can solve few-shot problems. In [1], a memory-augmented neural network can change the bias through weight updating and adjust the output result through cache representations in memory stores. Benaim et al. [4] propose a unsupervised domain translation method that has two steps: first, they train a variational autoencoder for domain B; then, given a new sample  $x$ , a variational autoencoder is created for domain A by directly adapting the layers that are close to the image in order to directly fit  $x$ , and other layers are indirectly adapted.

Different from the above meta-learning methods, several works adopt feature extraction or domain transfer methods to enhance the performance of meta-learning. In [32], Zhou et al. propose deep meta-learning that can achieve good performance by integrating the representation of deep learning into meta-learning. Instead of using original data, they feed the meta-learner with deep features extracted by the ResNet50. Sun et al. [28] show a novel few-shot learning

method named meta-transfer learning (MTL) that learns data information from a deep neural network and then transfers to a shallow neural network.

These algorithms do improve the performance of meta-learning in few-shot image classification, but meanwhile suffer from the problem of high computational complexity.

## 2.3 Knowledge Distillation

Knowledge distillation aims to transfer knowledge from a teacher network with deep layers and good performance to a high-accuracy student network with shallow layers, which has drawn much attention in recent years.

In [5], Bucilu et al. propose model compression, in which the purpose is to compress large and complicated ensembles into a smaller and simpler model, without significant loss in performance. Hinton et al. [8] propose knowledge distillation, in which a student network is trained by the soft output (also called dark knowledge) of an ensemble of a teacher network. Compared with a one-hot label, the softening softmax from a teacher network contains more information about different classes among data, and hence helps the student network achieve better performance.

There have been lots of literature working on knowledge distillation. In [24], Romero et al. propose the training of a student network using both the final output and intermediate representations of a teacher network, which adds a regressor on intermediate layers to match different size of teacher's and student's outputs. Ba and Caruana [3] show that a shallow feed-forward network trained using a deep manner can learn a complex function and achieve high accuracy that previously can be only achieved using deep model. In [20], Mirzadeh et al. think that the performance of a student network will degrade when the gap between student and teacher is large, so a teacher assistant is used as an intermediate network to reduce the impact of this gap.

The process of learning between a teacher and a student network is similar to the training process of generator and discriminator of generative adversarial networks (GAN). In recent years, the combination of these two concepts has attracted the interest of researchers [14, 26, 30]. They use an adversarial-based learning strategy to update the training loss of a student network, so that the student can better learn high-dimensional semantic information of the teacher network. Furlanello et al. [12] do not utilize the conventional model-compression of knowledge distillation, in which the number of parameters of student network is the same as that of the teacher network. Surprisingly, such a born-again network outperforms the corresponding teacher dramatically.

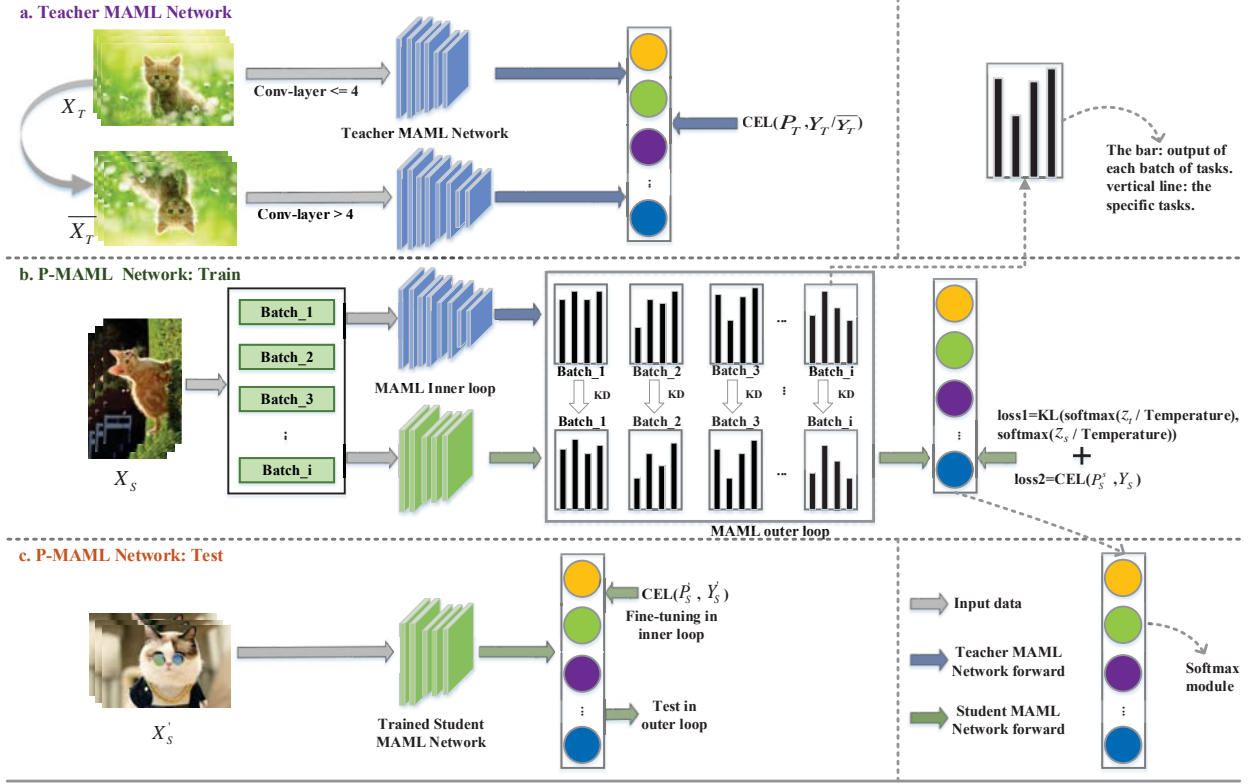
## 3 PRELIMINARY

In this section, we briefly introduce a specific meta-learning method MAML and knowledge distillation.

### 3.1 Model-Agnostic Meta-learning (MAML)

MAML is a meta-learning framework for one-shot learning. MAML aims to effectively bootstrap from a good meta-learner to learn fast on a new task, which assumes that tasks are drawn from a fixed distribution  $\mathcal{T} \sim \mathcal{P}(\mathcal{T})$ .

More formally, we define the base model to be a neural network  $f_\theta$  with meta-parameter  $\theta$ . When facing with a new task  $\mathcal{T}_i \sim \mathcal{P}(\mathcal{T})$



**Figure 1.** The framework of P-MAML ( $CEL$ : Cross-Entropy Loss;  $KL$ : Kullback Leibler Loss). First, we pre-train a teacher MAML network. Then, we distill knowledge from the teacher network into the student MAML network and train the P-MAML. Last, we test the P-MAML network in one-shot learning tasks.

from a support set  $\mathcal{T}_i^{su}$ , the meta-parameter  $\theta$  is updated to  $\theta_k^*$  after a small number of gradient updates  $k$ . This set of  $k$  updates is called the *inner-loop update process*. So, the  $\theta_k^*$ , called an inner-loop parameter, can be updated as follows

$$\theta_k^* = \theta_{k-1}^* - \alpha \nabla_{\theta} \mathcal{L}_1 \mathcal{T}_i^{su}(f_{\theta_{k-1}^*}), \quad (1)$$

where  $\alpha$  is the learning rate,  $\theta_k^*$  is the weight of base-learner after  $k$  steps towards the task  $\mathcal{T}_i$  and updated using cross entropy loss  $\mathcal{L}_1(\theta_k^*, \mathcal{T}_i^{su})$ . We define meta-objective from the query set  $\mathcal{T}_i^{qu}$ , which can be expressed as

$$\mathcal{L}_{1\mathcal{T}_i^{qu}}(\theta) = \min_{\theta} \sum_{\mathcal{T}_i \sim \mathcal{P}(\mathcal{T})} \mathcal{L}_{1\mathcal{T}_i^{qu}}(f_{\theta_k^*}). \quad (2)$$

Note that our goal is to optimize meta-parameter  $\theta$ , which is updated using  $\theta_k^*$  with multi-task information. As in Eq. (2), the meta-objective is to minimize the sum of the loss and optimize the initial meta-parameter. The optimization of this meta-objective is called the *outer loop update process*. The meta-parameter  $\theta$  is updated as follows

$$\theta \leftarrow \theta - \beta \nabla_{\theta} \sum_{\mathcal{T}_i \sim \mathcal{P}(\mathcal{T})} \mathcal{L}_{1\mathcal{T}_i^{qu}}(f_{\theta_k^*}), \quad (3)$$

where the step size  $\beta$  might be taken as a hyperparameter or learned rate in the outer update phase.

MAML-test aims to evaluate the performance of the meta-learner for new tasks through several fast adaptations. Given a new task  $\mathcal{T}_{new}$ , the initialized global  $\theta$  is trained to adapt to the sample of  $\mathcal{T}_{new}^{su}$  by a few gradient updates. Then, the test result on  $\mathcal{T}_{new}^{qu}$  is used to evaluate the learning ability of meta-learner.

## 3.2 Knowledge Distillation (KL)

Deep neural networks can extract deep features, detect more complex semantic information, and make a breakthrough in image recognition, speech recognition, machine translation and so on. However, the computational complexity and some hardware requirements of deep neural networks make it impossible for small devices. Therefore, mobile phones or small robots cannot use such a large network to run in real-time. In 2015, Hinton et al. proposed the concept of knowledge distillation to overcome this problem [8].

The idea is that a soft target (i.e. the predicted probability from softmax output) contains more information than a hard target (i.e. one hot label of the data). So a hyper-parameter temperature  $\tau > 1$  is introduced to soften a teacher network, and provide more information during the training. For example, in cat-dog classification, the probability of misidentifying a dog as a cat maybe 0.001, but the probability of misidentifying a dog as a truck maybe 0.00000001.

A higher  $\tau$  produces a softer probability distribution over classes. And the same  $\tau$  is applied to the output of student network  $P_S$ .

$$P_T^{\tau} = \text{softmax}\left(\frac{z_t}{\tau}\right), P_S^{\tau} = \text{softmax}\left(\frac{z_s}{\tau}\right). \quad (4)$$

The student network is then trained to optimize the following loss function:

$$\mathcal{L}_{KD}(\theta_S) = (1 - \lambda) * \mathcal{L}_1(y_{true}, P_S) + \lambda * \mathcal{L}_2(P_T^{\tau}, P_S^{\tau}), \quad (5)$$

where  $\theta_S$  is the parameter of student networks and  $\lambda$  is an adjustable hyper-parameter to balance the true label and KD loss.  $\mathcal{L}_1$  means the

cross-entropy loss,  $\mathcal{L}_2$  denotes the KL loss, indicating that the student network learns from the softened output of the teacher network.

## 4 METHODOLOGY

In this section, we elaborate on the proposed P-MAML, where the knowledge is distilled from a teacher MAML network into a student MAML network to improve the performance.

### 4.1 Framework

Figure 1 shows the framework of our proposed P-MAML approach, which is composed of three modules: the teacher MAML Network, the P-MAML network: Train, and the P-MAML network: Test.

On one hand, a teacher MAML network  $T$  is expected to be able to extract task-agnostic meta-level representation that captures the high-level abstract concept of the instances from many related tasks. Such knowledge can be distilled into a student MAML network  $S$  to quickly find good model parameters so as to accomplish one-shot learning. On the other hand, from a different viewpoint, the smaller student MAML network  $S$  is more likely to be applied in portable devices due to fewer parameters and lower computational cost.

Firstly, as shown in Figure 1 (a), we train a teacher MAML network, which commonly has a deep and wide network architecture. Then, as shown in Figure 1 (b), through knowledge distillation from the teacher MAML network, our proposed P-MAML, i.e. the student MAML network, is trained using a double-gradient procedure. After that, when facing with one-shot learning tasks, the trained P-MAML network can be tested, as seen in Figure 1 (c).

### 4.2 Teacher MAML Network

A teacher network commonly has a large architecture and a large capacity. The teacher MAML network is learned using double-gradient updates, as given in Eq. (1) and Eq. (3). However, as observed in Figure 1 (a), for different cases that the number of network layers is no more than 4 or greater than 4, the teacher MAML network has a different preprocessing and network architecture.

When the number of network layers is no more than 4, as shown in the upper part of Figure 1 (a), we take as input the original data  $X_T$  and train the teacher network with the label  $Y_T$ . The trained teacher network is able to implement quick adaptation for new tasks. However, when the number of network's layers is greater than 4, as shown in the lower part of Figure 1 (a), the original procedure fails in extensive experiments because 1) the amount of data is insufficient to train a deeper teacher MAML network and 2) the issue of gradient vanishing appears. Therefore, in order to solve these two problems, we propose to 1) use data augmentation to strengthen the training data and 2) employ a residual network (ResNet) to solve the gradient-vanishing problem.

Specifically, regarding the data augmentation, we rotate  $X_T$  by  $60^\circ$ ,  $90^\circ$  and  $180^\circ$  and then generate  $\overline{X_T}$  with the corresponding ground truth  $\overline{Y_T}$ . With regard to the ResNet, we add several shortcut connections in the output layer of the network.

For convenience, the teacher MAML network is denoted by  $z_t = f_t(x_t, W_t)$ , where  $f_t$  indicates a mapping function,  $x_t$  represents each of input data,  $W_t$  represents the network parameter and  $z_t$  is the output. And the final prediction probability in the teacher MAML network is obtained by  $P_t = \text{softmax}(z_t)$ .

If the teacher model is the result of an ensemble, either  $P_T$  or  $z_t$  is obtained by averaging outputs from different networks, including both arithmetic average and geometric average. Finally, meta-parameter  $W_t$  is updated in the process of back-propagation of the cross-entropy loss (CEL).

### 4.3 P-MAML Network: Train

In Figure 1 (b), there are two MAML networks: the teacher network from Figure 1 (a) has learned a good meta-parameter, and a student network has a small architecture and small capacity. How do we use the P-MAML method to train the student MAML network?

First, in MAML inner loop, we input training data  $X_S$  (generally,  $X_T$  has more samples) through a mini-batch method. The student network is trained using the support set of each batch of tasks, as given in Eq.(1). The teacher network is fine-tuned to adapt to new tasks. Then, in MAML outer loop, we update meta-parameter using the query set of each batch of tasks, as given in Eq.(3). We consider the combination of KD and MAML from the following two perspectives that are named by P-MAML<sub>last</sub> and P-MAML<sub>every</sub>:

**P-MAML<sub>last</sub>.** We propose this method to only distill the last batch of tasks from the teacher network to the student network, as the Batch<sub>i</sub> in Figure 1 (b).

**P-MAML<sub>every</sub>.** We propose this method to distill each batch of tasks from the teacher to the student network. The output of the query set of each batch of tasks from the teacher network is distilled to guide the training of the student network. For example, in Figure 1 (b), each small rectangle in the box on the upper part of the MAML outer loop represents the output of the teacher network, and each black bar represents the specific task in every batch. With the arrow direction of KD, the training loss of the student network is guided.

The student network not only learns the knowledge of the teacher network but also learns the updating direction of the teacher's meta-parameter in the iterative process. P-MAML<sub>every</sub> helps the student network to learn a good meta-learner. The student network is trained using the following Eq. (6)

$$\begin{aligned} \theta^S \leftarrow \theta^S - (1 - \lambda) \nabla_{\theta^S} \sum_{T_i \sim \mathcal{P}(T)} \mathcal{L}_{1T_i^{qu}}(f\theta_k^{S*}) \\ + \lambda \nabla_{\theta^S} \sum_{T_i \sim \mathcal{P}(T)} \mathcal{L}_{2T_i^{qu}}\left(\frac{z_t}{\tau}, \frac{z_s}{\tau}\right), \end{aligned} \quad (6)$$

where  $z_s = f_s(x_s, W_s)$  is the output of the query set of student network. The student final prediction probability is  $P_S^t = \text{softmax}(z_t)$ . The first part of Eq. (6) is a cross-entropy loss between the prediction of student network and ground-truth. The second part of Eq. (6) is a KD loss with the output of the teacher network and student network, where Kullback-Leibler divergence is used.

### 4.4 P-MAML Network: Test

We test the performance of the student MAML network with KD from the teacher MAML network, as observed in Figure 1 (c).

We define  $X'_S$  as test data of the student network to evaluate the P-MAML algorithm. In MAML inner loop, the student network is fine-tuned by minimizing the cross-entropy loss between the ground-truth  $Y'_S$  and the prediction probability  $P'_S$ . Test classification results are given in the outer loop of MAML. And, in the next section, a large number of experiments show that with the assistance of the teacher's network, the efficiency of portable MAML network can be significantly improved in terms of one-shot learning.

The corresponding algorithm is outlined in Algorithm 1. Algorithm 2 shows the pseudo-code of P-MAML\_every knowledge distillation method in this paper.

---

**Algorithm 1** P-MAML for One-shot Supervised learning

---

**Require:** Distribution of tasks  $\mathcal{P}(\mathcal{T})$ ;  
**Require:** The learning rate  $\alpha$ ;  
1: Pre-train  $\theta^T$ ; Initialize  $\theta^S$  randomly  
2: **while** not done **do**  
3:   Sample mini-batch of tasks  $\mathcal{T}_i \sim \mathcal{P}(\mathcal{T})$   
4:   Each of  $\mathcal{T}_i$  has support set  $\mathcal{T}_i^{su}$  and query set  $\mathcal{T}_i^{qu}$   
5:   **Inner loop:**  
6:    Calculate  $\nabla_{\theta^T} \mathcal{L}_{1\mathcal{T}_i^{su}}(f_{\theta^T})$  and  $\nabla_{\theta^S} \mathcal{L}_{1\mathcal{T}_i^{su}}(f_{\theta^S})$   
7:    Compute adapted parameters with gradient descent:  $\theta_k^{T*} = \theta^T - \alpha \nabla_{\theta^T} \mathcal{L}_{1\mathcal{T}_i^{su}}(f_{\theta^T})$  and  $\theta_k^{S*} = \theta^S - \alpha \nabla_{\theta^S} \mathcal{L}_{1\mathcal{T}_i^{su}}(f_{\theta^S})$   
8:    P-MAML\_every;  
9: **end while**

---



---

**Algorithm 2** P-MAML\_every

---

**Require:** Distribution of tasks  $\mathcal{P}(\mathcal{T})$ ;  
**Require:** Temperature  $\tau$ ; Balance factor  $\lambda$ ;  $\theta_k^{T*}$ ;  $\theta_k^{S*}$ ;  
**while** not done **do**  
2:   **Outer loop:**  
3:    **for all** tasks **do**  
4:      Calculate  $\mathcal{L}_{1\mathcal{T}_i^{qu}}(f_{\theta_k^{S*}})$   
5:      Save the teacher network pre-softmax output of the every batch of tasks of query set  $z_t$  and that of student network  $z_s$   
6:    **end for**  
7:     $\theta^S \leftarrow \theta^S - (1 - \lambda) \nabla_{\theta^S} \sum_{\mathcal{T}_i \sim \mathcal{P}(\mathcal{T})} \mathcal{L}_{1\mathcal{T}_i^{qu}}(f_{\theta_k^{S*}}) + \lambda \nabla_{\theta^S} \sum_{\mathcal{T}_i \sim \mathcal{P}(\mathcal{T})} \mathcal{L}_{2\mathcal{T}_i^{qu}}(\frac{z_t}{\tau}, \frac{z_s}{\tau})$   
8: **end while**

---

## 5 EXPERIMENT

In this section, we evaluate the proposed P-MAML in terms of few-shot image recognition and compare our approach with state-of-the-art baselines.

### 5.1 Datasets

We conduct experiments on three public datasets: MiniImagenet<sup>4</sup>, CUB<sup>5</sup>, and Omniglot<sup>6</sup>. Their information is detailed as follows:

- **MiniImagenet:** This dataset was proposed by Vinyals et al. [29] for one-shot learning evaluation. It has 100 classes and contains 600 samples of  $84 * 84 * 3$  color images for each class. These 100 classes are randomly divided into 64 base, 16 validation, and 20 novel classes for meta-training, validation and test [9, 13].
- **CUB:** This dataset is a fine-grained classification dataset, where we use CUB-200-2011 in this paper [7]. In total, there are 200 classes with 11,788 images, and are randomly splitted 100 base, 50 validation, and 50 novel classes.
- **Omniglot:** It consists of 20 instances of 1623 characters from 50 different alphabets, in which each of instances is collected by a different person. To meet our requirements, We first augment the classes by rotations in 90, 180, 270 degrees, resulting in 6492 classes [27]. Then, these classes are split into 4112 for meta-training, 688 for meta-validation and 1692 for meta-test.

<sup>4</sup> [http://image-net.org/image/ILSVRC2015/ILSVRC2015\\_CLS-LOC.tar.gz](http://image-net.org/image/ILSVRC2015/ILSVRC2015_CLS-LOC.tar.gz).

<sup>5</sup> <http://www.vision.caltech.edu/visipedia-data/CUB-200-2011>.

<sup>6</sup> [https://github.com/brendenlake/omniglot/blob/master/python/images\\_evaluation.zip](https://github.com/brendenlake/omniglot/blob/master/python/images_evaluation.zip).

### 5.2 Implementation details

The N-way, K-shot image recognition is considered on the CUB, MiniImagenet, and Omniglot datasets, where a gradient update is computed using a data batch with  $N \times K$  samples.

The *Conv1*, *Conv2* and *Conv4* are frameworks for a student network, while *Conv4*, *Conv6*, *Conv8*, *ResNet10* and *ResNet18* are frameworks for a teacher network. We take the *Conv4* as an example. Our model’s architecture has 4 modules with a  $3 \times 3$  convolutions and 64 filters. Then, the batch normalization, a ReLU function, and  $2 \times 2$  max-pooling are considered in this model. Besides, the last layer uses soft-max. Finally, we use the cross-entropy as the loss function for all cases.

In the meta-training stage, we train 60,000 episodes. Four-task for each batch is selected on MiniImagenet and CUB, while, Omniglot datasets have 32-task for every batch. We use the validation set to select the training episodes with the best accuracy and model parameter for meta-test. In each episode, we randomly sample 5-way for meta-training and meta-test. In the meta-test stage, we pick 800 tasks to test and find average results using the parameter of the best model.

### 5.3 Baselines

We consider to evaluate the performance of the following methods:

- Matching Nets [29]. This model can map a small set of dimensions and an unmarked test sample to its corresponding label, avoiding to adjust the new label category in this process.
- Meta-SGD [19]. Meta-SGD can not only initiate learning for learners, but also learn the updated direction and learning rate of learners.
- MAML [9]. MAML model can learn a good initialization and adapt to a new task through a few updates.
- P-MAML\_last. We propose this method to only distill the last batch of tasks from the teacher to student network.
- P-MAML\_every. We propose this method to distill each batch of tasks from the teacher to student network.

### 5.4 Experimental Results on One-Shot Learning

In this subsection, we report experimental results on three datasets, where the hyper-parameter  $\tau$  and  $\lambda$  are set as 10 and 0.9 all cases respectively<sup>7</sup>.

#### 5.4.1 Performance on CUB Dataset

In order to evaluate our P-MAML, we show the performance of P-MAML and baselines in Table 1. Overall speaking, it is concluded that P-MAML achieves better performance in one-shot learning than baselines owing to the knowledge distillation.

More concretely, it is observed from the upper part in Table 1 that P-MAML\_last with 2 layers gets an improvement by 3% while P-MAML\_every with 2 layers gets an improvement by 5% compared to MAML with the same layers (*Conv2\_MAML*).

In the lower part of Table 1, we focus on the 4-layer student network and a teacher network with various layers. It is observed that all P-MAML networks exceed the performance of baselines. When the layer of teacher network increases, the accuracy of one-shot learning gets improved. And P-MAML\_every shows its superiority over P-MAML\_last because more knowledge is distilled from various representations in the teacher network. One point to be noted is that

<sup>7</sup> The selection of  $\tau$  and  $\lambda$  are shown in Section 5.6

when considering a teaching assistant for an 18-layer teacher network, the accuracy gets enhanced by 4% because the difference between teacher and student is too large in this case [20].

Furthermore, in order to visualize such improvement, we depict the test accuracy of various approaches in the upper part of Figure 2. It is clearly observed that knowledge distillation greatly improves the performance of one-shot learning.

**Table 1.** Test Accuracy on CUB and MiniImagenet. ‘18-8-4’ stands for distilling teacher network (ResNet18) to student network (Conv4) with the help of teacher assistant (Conv8).

Datasets	CUB	magenet
Model Structure	5-way-1-shot	5-way-1-shot
<i>Conv2_MAML</i>	38.01%±0.79%	37.37%±0.71%
<i>P-MAML_last_4-2</i>	<b>41.56%±0.83%</b>	<b>39.65%±1.49%</b>
<i>P-MAML_every_4-2</i>	<b>43.02%±0.55%</b>	<b>40.93%±1.23%</b>
<i>Conv4_Meta-SGD</i> ( [19])	53.34%±0.97%	50.47%±1.87%
<i>Conv4_MAML</i> ( [9])	55.98%±0.89%	47.65%±0.81%
<i>Conv4_Matching Nets</i> ( [29])	56.53%±0.99%	43.56%±0.84%
<i>P-MAML_last_6-4</i>	<b>59.98%±0.95%</b>	<b>48.19%±1.22%</b>
<i>P-MAML_every_6-4</i>	<b>61.39%±0.98%</b>	<b>50.16%±1.18%</b>
<i>P-MAML_last_8-4</i>	<b>62.05%±1.26%</b>	<b>49.59%±0.91%</b>
<i>P-MAML_every_8-4</i>	<b>64.52%±0.68%</b>	<b>51.62%±0.89%</b>
<i>P-MAML_last_10-4</i>	<b>61.02%±0.94%</b>	<b>48.09%±0.90%</b>
<i>P-MAML_every_10-4</i>	<b>62.03%±1.44%</b>	<b>49.76%±1.21%</b>
<i>P-MAML_last_18-4</i>	<b>60.85%±0.84%</b>	<b>47.90%±0.69%</b>
<i>P-MAML_every_18-4</i>	<b>61.24%±0.85%</b>	<b>48.94%±0.72%</b>
<i>P-MAML_last_18-8-4</i>	<b>64.05%±1.02%</b>	<b>50.87%±0.65%</b>
<i>P-MAML_every_18-8-4</i>	<b>65.97%±0.92%</b>	<b>53.02%±0.97%</b>

#### 5.4.2 Performance on MiniImagenet dataset

In Table 1, we show the performance of the MiniImagenet dataset. Overall speaking, our method is better than baselines.

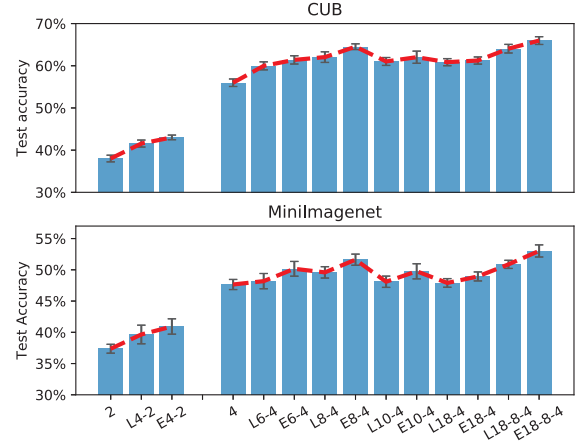
To be specific, the upper part of Table 1 shows that the knowledge distillation from 4 layers to 2 layers. *P-MAML\_every* enhance 4% compared to *MAML* with 2 layers.

As same as CUB dataset, we consider the distillation performance from different-layer teacher network to 4-layer student network in the lower part of Table 1. One point to be noted is that the distillation performance of 10-layer network is 3% lower than that of 8-layer network, which is caused by the larger difference between the teacher network and the student network. In such case, we use the teacher assistant to help students training. Finally, the best distillation performance of *P-MAML\_every* with a teacher assistant is 7% higher than the same 4-layer *MAML* network, and increases by 9% on average compared with other baselines.

In addition, in the lower part of Figure 2, we also show the visualization of test results. It is further shown that 1) knowledge distillation greatly improves the performance of one-shot learning and 2) more guidance can lead to better performance.

#### 5.4.3 Performance on Omniglot dataset

The results of 5-way-1-shot can be observed from the second column of Table 2. *P-MAML* is slightly better than the state-of-the-art models on all classification tasks. In this dataset, we only use 4-layer teacher network, because 98% recognition rate has been achieved with one-shot sample of this structure.



**Figure 2.** Test Accuracy. *L* denotes *P-MAML\_last*; *E* denotes *P-MAML\_every*; ‘t-s’ in horizon axis indicates that the teacher *MAML* has *t* layers while the student *MAML* has *s* layers. Please see the name method in Table 1.

Moreover, the student network with 2 layers can increase by 4% to 93.02% with the help of knowledge distillation of every batch tasks. Even for 1-layer student network with *P-MAML\_last*, it has a 1% improvement.

## 5.5 Experimental Results on Few-Shot Learning

In addition to one-shot learning, we also study few-shot learning (1-shot, 3-shot, 5-shot, 7-shot, 9-shot) on the Omniglot dataset to further verify the feasibility of *P-MAML* in this paper. These experimental results are shown in Table 2.

In horizontal comparison, under the condition of 5-way classification, the accuracy of image classification is enhanced with the increase of the number of samples. This is because more data can train a better meta-learner. As observed, the vertical comparison in Table 2 depicts a rising trend. This is because when using the same samples, 1) deep network can extract more features and 2) *P-MAML* can bring better data information.

More concretely, it is observed from the upper part in Table 1 that *P-MAML\_every* with 1 layer gets an advancement by 9.6% while *P-MAML\_last* with 1 layer gets a growth by 7% compared to *MAML* with the same layer (*Conv2\_MAML*) on 5-way-9-shot.

In the lower part of Table 1, the knowledge distillation is achieved from 4-layer teacher network to 2-layer student network. *P-MAML\_every* shows its prominence over *P-MAML\_last* and baselines. For example, the *Conv2\_MAML* without KD is 2% lower than that of *P-MAML\_last* while *P-MAML\_last* is lower than *P-MAML\_every* by 3% on average when considering 5-way 3-shot classification. One interesting thing to be noted is that the performance of the 2-layer student network *P-MAML\_every* on 5-way-7-shot is better than the *Conv2\_MAML* on the 5-way-9-shot.

## 5.6 Analysis on Hyper-Parameters and Complexity

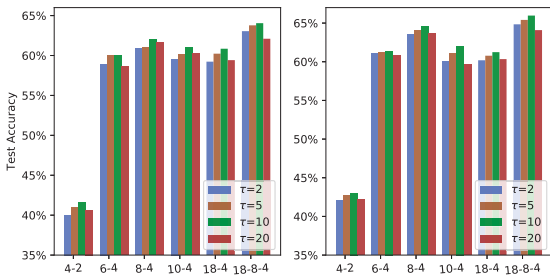
### 5.6.1 Choose the Best Hyper-Parameter $\tau$

We compare the experimental results at different temperature values, such as 2, 5, 10 and 20, on CUB dataset. Following these related works [8, 21], we know that hyper-parameter  $\lambda$  tends to KD loss,

Datasets	Omniglot				
Model Structure	5-way-1-shot	5-way-3-shot	5-way-5-shot	5-way-7-shot	5-way-9-shot
<i>Conv1</i> _MAML	73.39% $\pm$ 1.97%	75.39% $\pm$ 0.66%	77.80% $\pm$ 1.75%	78.02% $\pm$ 0.83%	80.96% $\pm$ 0.53%
P-MAML <sub>last</sub> _4-1	<b>74.01%<math>\pm</math>0.92%</b>	<b>76.01%<math>\pm</math>0.82%</b>	<b>79.89%<math>\pm</math>1.46%</b>	<b>81.97%<math>\pm</math>0.63%</b>	<b>83.27%<math>\pm</math>0.55%</b>
P-MAML <sub>every</sub> _4-1	<b>76.63%<math>\pm</math>1.61%</b>	<b>79.45%<math>\pm</math>0.64%</b>	<b>82.72%<math>\pm</math>1.28%</b>	<b>84.89%<math>\pm</math>1.52%</b>	<b>90.16%<math>\pm</math>1.38%</b>
<i>Conv2</i> _MAML	89.85% $\pm$ 1.35%	92.61% $\pm$ 0.78%	95.60% $\pm$ 0.92%	97.78% $\pm$ 0.54%	98.13% $\pm$ 0.56%
P-MAML <sub>last</sub> _4-2	<b>91.99%<math>\pm</math>1.12%</b>	<b>93.53%<math>\pm</math>1.06%</b>	<b>96.67%<math>\pm</math>0.64%</b>	<b>98.39%<math>\pm</math>0.83%</b>	<b>98.99%<math>\pm</math>1.38%</b>
P-MAML <sub>every</sub> _4-2	<b>93.02%<math>\pm</math>0.45%</b>	<b>95.34%<math>\pm</math>0.83%</b>	<b>97.45%<math>\pm</math>0.64%</b>	<b>99.09%<math>\pm</math>1.21%</b>	<b>99.56%<math>\pm</math>1.08%</b>

**Table 2.** Test accuracy on Omniglot dataset, where different model architectures are considered.

which will obtain a good performance. Therefore, when comparing the effect of different temperatures, we fixed  $\lambda$  as 0.9.



**Figure 3.** Test accuracy with temperatures on CUB dataset when  $\lambda = 0.9$ .

We illustrate the average accuracy of 800 test tasks in Figure 3, with P-MAML<sub>last</sub> on the left and P-MAML<sub>every</sub> on the right. It is observed from the bar chart of Figure 3 that the temperature does have a considerable effect on the classification accuracy.

From the overall trend of per distillation module in Figure 3,  $\tau = 10$  shows the highest performance. For example,  $\tau = 10$  reaches 65.97% for P-MAML<sub>every</sub> and 64.05% for P-MAML<sub>last</sub>, where the best accuracy is obtained by the student network (18-8-4) with teacher assistant. To sum up, for all our experiments, we select temperature value  $\tau = 10$ .

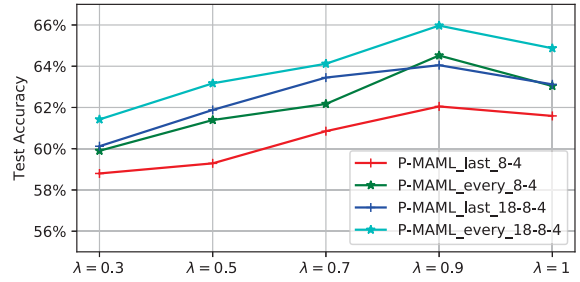
### 5.6.2 Analysis on Hyper-Parameter $\lambda$

We evaluate the effect of different  $\lambda$  in the best two distillation networks, i.e. the distillation in cases of 8-4 and 18-8-4.

In Figure 4, with an increasing  $\lambda$ , from 0.3 to 0.9 with a space of 0.2, the test accuracy improves gradually. This is because the adjustment factor  $\lambda$  is used to adjust the ratio of  $\mathcal{L}_1$  and  $\mathcal{L}_2$  in total  $\mathcal{L}$ . The larger the  $\lambda$  is, the more the global  $\mathcal{L}$  is inclined to the output of the teacher network ( $\mathcal{L}_2$ ), so that the student network can learn more data information in the training process. As the ablation study with  $\lambda = 1$ , the accuracy rate decreases, indicating that the teacher network and real label guidance are very helpful. To sum up, we select  $\lambda = 0.9$  for P-MAML.

### 5.6.3 Computational Complexity

Please be noted that P-MAML has the same computational complexity as the standard MAML because they have an identical update



**Figure 4.** Different  $\lambda$  values on CUB dataset when  $\tau = 10$ .

procedure except the novel loss function from knowledge distillation. On the other hand, if considering a fixed accuracy, the required number of layers in P-MAML is less than that of standard MAML. From this viewpoint, we may claim that P-MAML is more efficient.

The number of parameters in these networks is shown in Table 3. It is observed that the number of parameters in *ResNet18* is almost 100 times of that of *Conv4*, which indicates that its training time and memory requirements are very large.

**Table 3.** Computational complexity of different model architecture.

Model	<i>Conv4</i>	<i>Conv6</i>	<i>Conv8</i>	<i>ResNet10</i>	<i>ResNet18</i>
Params	0.4843M	0.7808M	1.0772M	19.6334M	44.7163M

## 6 Conclusion

In this paper, we propose a portable meta-learning approach for small devices, named P-MAML, which considers knowledge distillation into meta-learning and enables to accomplish one-shot learning using different model architectures. The student MAML network learns from the teacher MAML network in two different ways: 1) knowledge is distilled in every batch of tasks of teacher’s network (P-MAML<sub>every</sub>) and 2) knowledge is only distilled from then last batch of tasks of teacher’s network (P-MAML<sub>last</sub>). And the assistant network is considered to reduce the gap between a large teacher network and student work.

Extensive experiments on three real datasets demonstrate the effectiveness and superiority of our approach in one-shot image recognition in comparison to state-of-the-art baselines. Experiment results show that P-MAML<sub>every</sub> exceeds P-MAML<sub>last</sub>.

## REFERENCES

- [1] Santoro Adam, Bartunov Sergey, Botvinick Matthew, Wierstra Daan, and Lillicrap Timothy, ‘Meta-learning with memory-augmented neural networks’, in *Proceedings of the 33rd International Conference on Machine Learning (ICML)*, (2016).
- [2] Antreas Antoniou, Harrison Edwards, and Amos J.Storkey, ‘How to train your MAML’, in *7th International Conference on Learning Representations (ICLR)*, (2019).
- [3] Jimmy Ba and Rich Caruana, ‘Do deep nets really need to be deep?’, in *Advances in Neural Information Processing Systems (NIPS)*, (2014).
- [4] Sagie Benaim and Lior Wolf, ‘One-shot unsupervised cross domain translation’, in *Advances in Neural Information Processing Systems (NIPS)*, (2018).
- [5] Cristian Bucila, Rich Caruana, and Alexandru Niculescu-Mizil, ‘Model compression’, in *Proceedings of the Twelfth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD)*, (2006).
- [6] Rich Caruana, ‘Multitask learning’, *Machine Learning*, **28**(1), 41–75, (1997).
- [7] Wah Catherine, Branson Steve, Welinder Peter, Perona Pietro, and Belongie Serge, ‘The caltech-ucsd birds200-2011 dataset’, *California Institute of Technology*, (2011).
- [8] Geoffrey E.Hinton, Oriol Vinyals, and Jeffrey Dean, ‘Distilling the knowledge in a neural network’, *In CoRR*, **abs/1503.02531**, (2015).
- [9] Chelsea Finn, Pieter Abbeel, and Sergey Levine, ‘Model-agnostic meta-learning for fast adaptation of deep networks’, in *Proceedings of the 34th International Conference on Machine Learning (ICML)*, (2017).
- [10] Chelsea Finn, Kelvin Xu, and Sergey Levine, ‘Probabilistic model-agnostic meta-learning’, in *Advances in Neural Information Processing Systems (NIPS)*, (2018).
- [11] Santo Fortunato and Darko Hric, ‘Community detection in networks: A user guide’, *In CoRR*, **abs/1608.00163**, (2016).
- [12] Tommaso Furlanello, Zachary Chase Lipton, Michael Tschannen, Laurent Itti, and Anima Anandkumar, ‘Born-again neural networks’, in *Proceedings of the 35th International Conference on Machine Learning (ICML)*, (2018).
- [13] Erin Grant, Chelsea Finn, Sergey Levine, and Trevor Darrell, ‘Recasting gradient-based meta-learning as hierarchical bayes’, in *6th International Conference on Learning Representations (ICLR)*, (2018).
- [14] Md.Akmal Haidar and Mehdi Rezagholizadeh, ‘Textkd-gan: Text generation using knowledge distillation and generative adversarial networks’, *In CoRR*, **abs/1905.01976**, (2019).
- [15] Md Zahidul Islam and Ljiljana Brankovic, ‘Privacy preserving data mining: A noise addition framework using a novel clustering technique’, *Knowl.-Based System*, (2011).
- [16] Yangqing Jia, Evan Shelhamer, and Jeff Donahue, ‘Caffe: Convolutional architecture for fast feature embedding’, in *Proceedings of the ACM International Conference on Multimedia*, (2014).
- [17] Gregory Koch, Richard Zemel, and Ruslan Salakhutdinov, ‘Siamese neural networks for one-shot image recognition’, in *Proceedings of the 32th International Conference on Machine Learning (ICML)*, (2015).
- [18] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E.Hinton, ‘Imagenet classification with deep convolutional neural networks’, in *Advances in Neural Information Processing Systems (NIPS)*, (2012).
- [19] Zhenguo Li, Fengwei Zhou, Fei Chen, and Hang Li, ‘Meta-sgd: Learning to learn quickly for few shot learning’, *In CoRR*, **abs/1707.09835**, (2017).
- [20] Seyed-Iman Mirzadeh, Mehrdad Farajtabar, Ang Li, and Hassan Ghasemzadeh, ‘Improved knowledge distillation via teacher assistant: Bridging the gap between student and teacher’, *In CoRR*, **abs/1902.03393**, (2019).
- [21] Baoyun Peng, Xiao Jin, Jiaheng Liu, and Shunfeng Zhou, ‘Correlation congruence for knowledge distillation’, *In CoRR*, **abs/1904.01802**, (2019).
- [22] Hang Qi, Matthew Brown, and David G.Lowe, ‘Low-shot learning with imprinted weights’, in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, (2018).
- [23] Sachin Ravi and Hugo Larochelle, ‘Optimization as a model for few-shot learning’, in *5th International Conference on Learning Representations (ICLR)*, (2017).
- [24] Adriana Romero, Nicolas Ballas, Samira Ebrahimi Kahou, and Antoine Chassang, ‘Fitnets: Hints for thin deep nets’, in *3rd International Conference on Learning Representations (ICLR)*, (2015).
- [25] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, and Fei-Fei Li, ‘Imagenet large scale visual recognition challenge’, *International Journal of Computer Vision*, (2015).
- [26] Zhiqiang Shen, Zhankui He, and Xiangyang Xue, ‘MEAL: multi-model ensemble via adversarial learning’, in *The Thirty-Third AAAI Conference on Artificial Intelligence (AAAI)*, (2019).
- [27] Jake Snell, Kevin Swersky, and Richard S.Zemel, ‘Prototypical networks for few-shot learning’, in *Advances in Neural Information Processing Systems (NIPS)*, (2017).
- [28] Qianru Sun, Yaoyao Li, Tat-Seng Chua, and Bernt Schiele, ‘Meta-transfer learning for few-shot learning’, in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, (2019).
- [29] Oriol Vinyals, Charles Blundell, Tim Lillicrap, Koray Kavukcuoglu, and Daan Wierstra, ‘Matching networks for one shot learning’, in *Advances in Neural Information Processing Systems (NIPS)*, (2016).
- [30] Zheng Xu, Yen-Chang Hsu, and Jiawei Huang, ‘Training shallow and thin networks for acceleration via knowledge distillation with conditional adversarial networks’, in *6th International Conference on Learning Representations (ICLR)*, (2018).
- [31] Yabin Zhang, Hui Tang, and Kui Jia, ‘Fine-grained visual categorization using meta-learning optimization with sample selection of auxiliary data’, in *15th European Conference Computer Vision (ECCV)*, (2018).
- [32] Fengwei Zhou, Bin Wu, and Zhenguo Li, ‘Deep meta-learning: Learning to learn in the concept space’, *In CoRR*, **abs/1802.03596**, (2018).