Adversarial Meta-Learning

Chengxiang Yin,[1] **Jian Tang,**[2] **Zhiyuan Xu,**[1] **Yanzhi Wang,**[3]
[1]Syracuse University, [2]DiDi AI Labs, [3]Northeastern University
cyin02@syr.edu, tangjian@didiglobal.com, zxu105@syr.edu
yanz.wang@northeastern.edu

Abstract

Meta-learning enables a model to learn from very limited data to undertake a new task. In this paper, we study the general meta-learning with adversarial samples. We present a meta-learning algorithm, ADML (ADversarial Meta-Learner), which leverages clean and adversarial samples to optimize the initialization of a learning model in an adversarial manner. ADML leads to the following desirable properties: 1) it turns out to be very effective even in the cases with only clean samples; 2) it is robust to adversarial samples, i.e., unlike other meta-learning algorithms, it only leads to a minor performance degradation when there are adversarial samples; 3) it sheds light on tackling the cases with limited and even contaminated samples. It has been shown by extensive experimental results that ADML consistently outperforms three representative meta-learning algorithms in the cases involving adversarial samples, on two widely-used image datasets, MiniImageNet and CIFAR100, in terms of both accuracy and robustness.

1 Introduction

Deep learning has made tremendous successes and emerged as a *de facto* approach in many application domains, such as computer vision and natural language processing, which, however, depends heavily on huge amounts of labeled training data. The goal of meta-learning is to enable a model (especially a Deep Neural Network (DNN)) to learn from only a small number of data samples to undertake a new task, which is critically important to machine intelligence but turns out to be very challenging. Currently, a common approach to learn is to train a model to undertake a task from scratch without making use of any previous experience. Specifically, a model is initialized randomly and then updated slowly using gradient descent with a large number of training samples. This kind of time-consuming and data-hungry training process is quite different from the way how a human learns quickly from only a few samples and obviously cannot meet the requirement of meta-learning. Several methods [1, 2, 3, 4] have been proposed to address meta-learning by fixing the above issue. For example, a well-known work [1] presents a novel meta-learning algorithm called MAML (Model-Agnostic Meta-Learning), which trains a model's initial parameters carefully such that it has the maximal performance on a new task after its parameters are updated through one or just a few gradient steps with a small amount of new data. This method is claimed to be model-agnostic since it can be directly applied to any learning model that can be trained with a gradient descent procedure.

Robustness is another major concern for machine intelligence. It has been shown by [5] that learning models can be easily fooled by adversarial manipulation of actual training data to cause incorrect classifications. Therefore, adversarial samples pose a serious security threat to learning tasks, which need to be properly and effectively handled by learning models and training algorithms. We show via experiments that existing meta-leaning algorithms (such as MAML [1], Matching Networks [2] and Relation Networks [4]) are also vulnerable to adversarial samples, i.e., adversarial samples can lead to a significant performance degradation for meta-learning. An adversarial approach, called

Preprint. Under review.

MetaGAN, was presented in [6] for few-shot classification, nevertheless, in addition to learning better decision boundaries by leveraging fake data with the power of GAN [7], it paid no attention to deal with the cases involving adversarial samples. To the best of our knowledge, none of existing works on meta-learning have well addressed adversarial samples, which, however, are the main focus of this paper.

In this paper, we extend meta-learning to a whole new dimension by studying how to quickly train a model (especially a DNN) for a new task using a small dataset with both clean and adversarial samples. Since both meta-learning and adversarial learning have been studied recently, a straightforward solution is to simply combine an existing meta-learning algorithms (e.g., MAML [1]) with adversarial training (e.g., [8]). However, we show such a approach does not work well by our experimental results. We present a novel ADversarial Meta-Learner (ADML), which utilizes antagonistic correlations between clean and adversarial samples to let the inner gradient update arm-wrestle with the meta-update to obtain a good and robust initialization of model parameters. Hence, "adversarial" in ADML refers to not only adversarial samples but also the way of updating the learning model. The design of ADML leads to several desirable properties. First, it turns out to be very effective even in the cases with only clean samples. Second, unlike other meta-learning algorithms, ADML is robust to adversarial samples since it only suffers from a minor performance degradation when encountering adversarial samples, and it consistently outperforms three representative meta-learning algorithms [1, 2, 4] in such cases. Most importantly, it opens up an interesting research direction and sheds light on dealing with the cases with limited and even contaminated samples, which are common in real life. We conducted a comprehensive empirical study for performance evaluation using two widely-used image datasets, MiniImageNet [2] and CIFAR100 [9]. Experimental results well justify the effectiveness and superiority of ADML in terms of both accuracy and robustness.

2 Related Work

Meta-Learning: Research on meta-learning has a long history, which can be traced back to some early works [10, 11]. Meta-learning, a standard methodology to tackle few-shot learning problems, has recently attracted extensive attention due to its important roles in achieving human-level intelligence. Several specialized models [2, 12, 3, 4] have been proposed for meta-learning, particularly for few-shot classification, by comparing similarity among data samples. Specifically, Koch *et al.* [12] leveraged a Siamese Networks to rank similarity between input samples and predict if two samples belong to the same class. In addition, Relation Networks [4] was proposed to classify query images by computing relation scores, which can be extended to few-shot learning. In [2], Vinyals *et al.* presented a neural network model, Matching Networks, which learn an embedding function and use the cosine distance in an attention kernel to measure similarity. Another work [3] leveraged a similar approach to few-shot classification but used the Euclidean distance with their embedding function.

Another popular approach to meta-learning is to develop a meta-learner to optimize key hyper-parameters (e.g., initialization) of the learning model. Specifically, Finn *et al.* [1] presented a model-agnostic meta-learner, MAML, to optimize the initialization of a learning model with the objective of maximizing its performance on a new task after updating its parameters with a small number of samples. Several other methods [13, 14, 15, 16] utilize an additional neural network, such as LSTM, to serve as the meta-learner. A seminal work [13] developed a meta-learner based on LSTMs and showed how the design of an optimization algorithm can be cast as a learning problem. Ravi *et al.* [14] proposed another LSTM-based meta-learner to learn a proper parameter update and a general initialization for the learning model. Compared to LSTM, a neural network [15] is equipped with a large external memory (such as Neural Turing Machine (NTM) [17]), which has also been leveraged for meta-learning. A recent work [16] presented a class of simple and generic meta-learners that use a novel combination of temporal convolutions and soft attention.

Adversarial Learning: DNN models have been shown to be vulnerable to adversarial samples. Particularly, Szegedy *et al.* [5] showed that they can cause a DNN to misclassify an image by applying a certain hardly perceptible perturbation, and moreover, the same perturbation can cause a different network (trained on a different subset of the dataset) to misclassify the same input. It has also been shown by Goodfellow *et al.* in [8] that injecting adversarial samples during training can increase the robustness of DNN models. In [18], Rozsa *et al.* conducted experiments on various adversarial sample generation methods with multiple deep Convolutional Neural Networks (CNNs), and found that adversarial samples are mostly transferable across similar network topologies, and better learning models are less vulnerable. The authors of [19] introduced the first practical demonstration of a black-box attack controlling a remotely hosted DNN without either the model internals or its training data. More recently, Kurakin *et al.* [20] studied adversarial learning at scale by proposing an algorithm to train a large scale model, Inception v3, on the ImageNet dataset, which has been shown to significantly increase the robustness against adversarial samples. In addition, the authors of [21] extended adversarial training to the text domain by applying perturbations to word embeddings in an RNN rather than to the original input itself, which has been shown to achieve the state-of-the-art results on multiple benchmark semi-supervised and purely supervised tasks.

To the best of our knowledge, meta-learning has not been studied in the setting with adversarial samples. We not only show a straightforward solution does not work well but also present a novel and effective method, ADML.

3 Adversarial Meta-Learning

3.1 Problem Statement

The regular machine learning problem seeks a model that maps observations $\mathbf{x}$ to output $\mathbf{y}$; and a training algorithm optimizes the parameters of the model with a training dataset, whose generalization is then evaluated on a testing dataset. While in the setting of meta-learning, the learning model is expected to be trained with limited data to be able to adapt to a new task quickly. Meta-learning includes meta-training and meta-testing. In the *meta-training*, we use a set $\mathcal{T}$ of T tasks, each of which has a loss function $\mathcal{L}_i$, and a dataset $\mathcal{D}_i$ (with limited data) that is further split into $\mathbf{D}_i$ and $\mathbf{D}'_i$ for training and testing respectively. For example, in our experiments, each task is a 5-way classification task.

We aim to develop a meta-learner (i.e., a learning algorithm) that takes as input the datasets $\mathcal{D} = \{\mathcal{D}_1, \cdots, \mathcal{D}_T\}$ and returns a model with parameters $\boldsymbol{\theta}$ that maximizes the average classification accuracy on the corresponding testing sets $\mathcal{D}' = \{\mathbf{D}'_1, \cdots, \mathbf{D}'_T\}$. Note that here these testing data are also used for meta-training. In the *meta-testing*, we evaluate the generalization of the learned model with parameters $\boldsymbol{\theta}$ on new tasks, whose corresponding training and testing datasets may include adversarial samples. The learned model is expected to learn quickly from just one (1-shot) or K (K-shot) training samples of a new task and deliver highly-accurate results on its testing samples. An ideal meta-learner is supposed to return a learning model that can deal with new tasks with only clean samples; and suffers from only a minor performance degradation for new tasks with adversarial samples. Note that we only consider classification here since so far only classification has been addressed in the context of adversarial learning [20]. We believe the proposed ADML can be easily extended to other scenarios as long as adversarial samples can be properly generated.

3.2 Adversarial Meta-Learner (ADML)

We formally present the proposed ADML as Algorithm 1 for *meta-training*. We consider a model $f_{\boldsymbol{\theta}}$ parameterized by $\boldsymbol{\theta}$, which is updated iteratively. Here, an updating *episode* includes an inner gradient update process (Line 8–Line 12) and a meta-update process (Line 14). Unlike MAML, for each task, additional adversarial samples are generated and used to enhance the robustness for meta-training. Note that our algorithm is not restricted to any particular adversarial sample generation method. We used the *Fast Gradient Sign Method* (FGSM) [8] in our experiments. For task $\mathcal{T}_i$, given a clean sample $(\mathbf{x}_c, \mathbf{y}_c)$ from $\mathcal{D}_i$, its corresponding adversarial sample $(\mathbf{x}_{adv}, \mathbf{y}_{adv})$ is generated using the following equations:

$$\begin{aligned} \mathbf{x}_{adv} &= \mathbf{x}_c + \epsilon \text{sign}(\nabla_{\mathbf{x}_c} J(f_{\boldsymbol{\theta}_{pre}}, \mathbf{x}_c, \mathbf{y}_c)); \\ \mathbf{y}_{adv} &= \mathbf{y}_c. \end{aligned} \tag{1}$$

where $J(f_{\boldsymbol{\theta}_{pre}}, \mathbf{x}_c, \mathbf{y}_c)$ represents the cost used to train a classification model $f_{\boldsymbol{\theta}_{pre}}$ parameterized by $\boldsymbol{\theta}_{pre}$, and ϵ specifies the size of the adversarial perturbation (the larger the ϵ, the higher the perturbation). Note that the classification model $f_{\boldsymbol{\theta}_{pre}}$ is pre-trained based on the corresponding dataset and its parameters $\boldsymbol{\theta}_{pre}$ are fixed during the meta-training (Algorithm 1) and meta-testing.

The key idea behind ADML is to utilize antagonistic correlations between clean and adversarial samples to let the inner gradient update and the meta-update arm-wrestle with each other to obtain a good initialization of model parameters $\boldsymbol{\theta}$, which is robust to adversarial samples. Specifically, in

Algorithm 1 Adversarial Meta-Learner (ADML)

```
1:  Require: $\alpha_1/\alpha_2$ and $\beta_1/\beta_2$: The step sizes for inner gradient update and meta-update respectively
2:  Require: $\mathcal{D}$: The datasets for meta-training
3:  Require: $< \mathcal{L}_i(\cdot) >$: The loss function for task $\mathcal{T}_i, \forall i \in \{1, \cdots, T\}$
4:  Randomly initialize $\boldsymbol{\theta}$;
5:  while not done do
6:      Sample batch of tasks $< \mathcal{T}_i >$ from task set $\mathcal{T}$;
7:      for all $\mathcal{T}_i$ do
8:          Sample $K$ clean samples $\left\{(\mathbf{x}_c{}^1, \mathbf{y}_c{}^1), ..., (\mathbf{x}_c{}^K, \mathbf{y}_c{}^K)\right\}$ from $\mathbf{D}_i$;
9:          Generate $K$ adversarial samples $\left\{(\mathbf{x}_{adv}{}^1, \mathbf{y}_{adv}{}^1), ..., (\mathbf{x}_{adv}{}^K, \mathbf{y}_{adv}{}^K)\right\}$ based on another $K$ clean samples from $\mathbf{D}_i$ to form a dataset $\overline{\mathbf{D}_i} := \{\mathbf{D}_{adv_i}, \mathbf{D}_{c_i}\}$ for the inner gradient update, containing $K$ adversarial samples and $K$ clean samples;
10:         Compute updated model parameters with gradient descent respectively:
            $\boldsymbol{\theta}'_{adv_i} := \boldsymbol{\theta} - \alpha_1 \nabla_{\boldsymbol{\theta}} \mathcal{L}_i(f_{\boldsymbol{\theta}}, \mathbf{D}_{adv_i})$; $\boldsymbol{\theta}'_{c_i} := \boldsymbol{\theta} - \alpha_2 \nabla_{\boldsymbol{\theta}} \mathcal{L}_i(f_{\boldsymbol{\theta}}, \mathbf{D}_{c_i})$;
11:         Sample $k$ clean samples $\left\{(\mathbf{x}_c{}^1, \mathbf{y}_c{}^1), ..., (\mathbf{x}_c{}^k, \mathbf{y}_c{}^k)\right\}$ from $\mathbf{D}'_i$;
12:         Generate $k$ adversarial samples $\left\{(\mathbf{x}_{adv}{}^1, \mathbf{y}_{adv}{}^1), ..., (\mathbf{x}_{adv}{}^k, \mathbf{y}_{adv}{}^k)\right\}$ based on another $k$ clean samples from $\mathbf{D}'_i$ to form a dataset $\overline{\mathbf{D}'_i} := \left\{\mathbf{D}'_{adv_i}, \mathbf{D}'_{c_i}\right\}$ for the meta-update, containing $k$ adversarial samples and $k$ clean samples;
13:     end for
14:     Update $\boldsymbol{\theta} := \boldsymbol{\theta} - \beta_1 \nabla_{\boldsymbol{\theta}} \sum_{\mathcal{T}_i \sim \mathcal{T}} \mathcal{L}_i(f_{\boldsymbol{\theta}'_{adv_i}}, \mathbf{D}'_{c_i})$; $\boldsymbol{\theta} := \boldsymbol{\theta} - \beta_2 \nabla_{\boldsymbol{\theta}} \sum_{\mathcal{T}_i \sim \mathcal{T}} \mathcal{L}_i(f_{\boldsymbol{\theta}'_{c_i}}, \mathbf{D}'_{adv_i})$;
15: end while
```

the inner gradient update, we compute the new model parameters (updated in two directions) $\boldsymbol{\theta}'_{adv_i}$ and $\boldsymbol{\theta}'_{c_i}$ based on generated adversarial samples $\mathbf{D}_{adv_i}$, and clean samples $\mathbf{D}_{c_i}$ in training set $\mathbf{D}_i$ of task $\mathcal{T}_i$ respectively using gradient decent (Line 10). In the meta-update process, we update the model parameters $\boldsymbol{\theta}$ by optimizing the losses $\mathcal{L}_i(f_{\boldsymbol{\theta}'_{adv_i}})$ and $\mathcal{L}_i(f_{\boldsymbol{\theta}'_{c_i}})$ of the model with updated parameters $\boldsymbol{\theta}'_{adv_i}$ and $\boldsymbol{\theta}'_{c_i}$ with respect to $\boldsymbol{\theta}$ based on the clean samples $\mathbf{D}'_{c_i}$ in testing set $\mathbf{D}'_i$ of task $\mathcal{T}_i$ and the corresponding adversarial samples $\mathbf{D}'_{adv_i}$ respectively:

$$\begin{aligned}
\min_{\boldsymbol{\theta}} \sum_{\mathcal{T}_i \sim \mathcal{T}} \mathcal{L}_i(f_{\boldsymbol{\theta}'_{adv_i}}, \mathbf{D}'_{c_i}) &= \min_{\boldsymbol{\theta}} \sum_{\mathcal{T}_i \sim \mathcal{T}} \mathcal{L}_i(f_{\boldsymbol{\theta} - \alpha_1 \nabla_{\boldsymbol{\theta}} \mathcal{L}_i(f_{\boldsymbol{\theta}}, \mathbf{D}_{adv_i})}, \mathbf{D}'_{c_i}); \\
\min_{\boldsymbol{\theta}} \sum_{\mathcal{T}_i \sim \mathcal{T}} \mathcal{L}_i(f_{\boldsymbol{\theta}'_{c_i}}, \mathbf{D}'_{adv_i}) &= \min_{\boldsymbol{\theta}} \sum_{\mathcal{T}_i \sim \mathcal{T}} \mathcal{L}_i(f_{\boldsymbol{\theta} - \alpha_2 \nabla_{\boldsymbol{\theta}} \mathcal{L}_i(f_{\boldsymbol{\theta}}, \mathbf{D}_{c_i})}, \mathbf{D}'_{adv_i}).
\end{aligned} \tag{2}$$

Note that in the meta-update, $\boldsymbol{\theta}$ is optimized in an *adversarial* manner: the gradient of the loss of the model with $\boldsymbol{\theta}'_{adv_i}$ (updated using adversarial samples $\mathbf{D}_{adv_i}$) is calculated based on clean samples $\mathbf{D}'_{c_i}$, while the gradient of the loss of the model with $\boldsymbol{\theta}'_{c_i}$ (updated using $\mathbf{D}_{c_i}$) is calculated based on adversarial samples $\mathbf{D}'_{adv_i}$. The arm-wrestling between the inner gradient update and the meta-update brings an obvious benefit: the model adapted to adversarial samples (through the inner gradient update using adversarial samples) is made suitable also for clean samples through the optimization of $\boldsymbol{\theta}$ in the meta-update based on the clean samples, and vice versa. So "*adversarial*" in ADML refers to not only adversarial samples but also the way of meta-training.

The meta-update of the model parameters $\boldsymbol{\theta}$ is performed as the last step of each episode (Line 14). Through the arm-wrestling between the inner gradient update and the meta-update in the meta-training process, $\boldsymbol{\theta}$ will be updated to a certain point, such that the average loss given by both adversarial samples and clean samples of all the tasks is minimized. In addition, we set the step sizes $\alpha_1 = \alpha_2 = 0.01, \beta_1 = \beta_2 = 0.001$, and set $\mathcal{L}_i(\cdot)$ of each classification task $\mathcal{T}_i$ to be the cross-entropy loss. K and k are task-specific, whose settings are discussed in the next section. It can be easily seen that ADML preserves the model-agnostic property of MAML because both the inner gradient update and the meta-update processes are fully compatible with any learning model that can be trained by gradient descent.

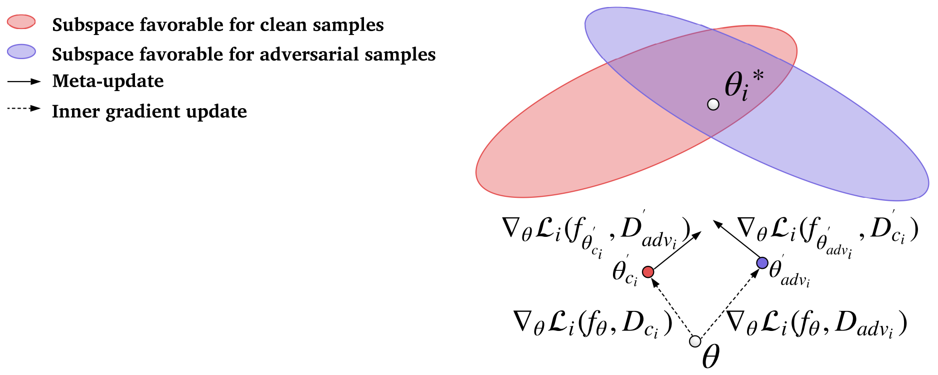
Subspace favorable for clean samples
Subspace favorable for adversarial samples
Meta-update
Inner gradient update
θ_i^*
$\nabla_\theta \mathcal{L}_i(f_{\theta'_{c_i}}, D'_{adv_i})$
$\nabla_\theta \mathcal{L}_i(f_{\theta'_{adv_i}}, D'_{c_i})$
θ'_{c_i}
θ'_{adv_i}
$\nabla_\theta \mathcal{L}_i(f_\theta, D_{c_i})$
$\nabla_\theta \mathcal{L}_i(f_\theta, D_{adv_i})$
θ

Figure 1: Illustration of design philosophy of ADML

We further illustrate the design philosophy of our algorithm in Figure 1. For each task $\mathcal{T}_i$, in the inner gradient update, ADML first drags $\boldsymbol{\theta}$ via gradient descent to the direction of the subspace that is favorable for adversarial samples (marked with the purple color) as well as another subspace that is favorable for clean samples (marked with the red color) to reach two points $\boldsymbol{\theta}'_{adv_i}$ and $\boldsymbol{\theta}'_{c_i}$ respectively (i.e., Line 10). Then in the meta-update, based on $\boldsymbol{\theta}'_{adv_i}$ and $\boldsymbol{\theta}'_{c_i}$, ADML further optimizes $\boldsymbol{\theta}$ to its antithetic subspaces respectively (i.e., Line 14), and hopefully $\boldsymbol{\theta}$ can reach the optimal point $\boldsymbol{\theta}_i^*$, which is supposed to fall into the intersection of the subspace pair and is able to achieve a good trade-off between clean and adversarial samples to boost the overall performance on both samples. Note that here we only show the updates via a single task. Using all the tasks in $\mathcal{T}$, $\boldsymbol{\theta}$ can be optimized to a point with the smallest average distance to the intersections of all the subspace pairs, and thus can be quickly adapted to new tasks even with adversarial samples.

As mentioned before, a rather straightforward solution to the above adversarial meta-learning problem is to simply combine a meta-learner (e.g., MAML [1]) with adversarial training (e.g., [8]). Specifically, we mix adversarial and clean samples to form both $\mathbf{D}_i$ (used in the inner gradient update) and $\mathbf{D}'_i$ (used in the meta-update), which are then used to calculate $\boldsymbol{\theta}'_i$ and update $\boldsymbol{\theta}$ using the following equations (just like MAML) respectively:

$$\begin{aligned} &\boldsymbol{\theta}'_i = \boldsymbol{\theta} - \alpha \nabla_{\boldsymbol{\theta}} \mathcal{L}_i(f_{\boldsymbol{\theta}}, \mathbf{D}_i); \\ &\boldsymbol{\theta} \leftarrow \boldsymbol{\theta} - \beta \nabla_{\boldsymbol{\theta}} \sum_{\mathcal{T}_i \sim \mathcal{T}} \mathcal{L}_i(f_{\boldsymbol{\theta}'_i}, \mathbf{D}'_i). \end{aligned} \tag{3}$$

We call this method *MAML-AD*, which is used as a baseline for performance evaluation. However, it has been shown by our experimental results that although MAML-AD can slightly mitigate the problem, it still suffers from a significant performance degradation for new tasks with adversarial samples. This clearly shows that simply involving adversarial samples during the meta-training does not necessarily enhance the model's robustness; and well justifies that our idea of doing the inner gradient update and the meta-update in an adversarial way is necessary.

4 Performance Evaluation

The goal of our evaluation is to test and verify three properties of ADML: 1) ADML can learn quickly from limited data via a few gradient updates for a new task, and it is effective even in the cases with only clean samples; 2) ADML suffers from a minor performance degradation and yields much better performance than other meta-learning algorithms when encountering adversarial samples; and 3) ADML maintains stable performance when the perturbation of adversarial samples (i.e., ϵ) escalates. In this section, we first introduce the experimental setup, and then present and analyze the results.

4.1 Experimental Setup

In our experiments, we employed two commonly-used image benchmarks, MiniImageNet [2], and CIFAR100 [9]. MiniImageNet is a benchmark for few-shot learning, which includes 100 classes

Table 1: Average classification accuracies on MiniImageNet (5-way, 1-shot)

Method	Meta-testing	$\epsilon = 2$		$\epsilon = 0.2$	
		Clean	Adversarial	Clean	Adversarial
MAML [1]	Clean	$48.47 \pm 1.78\%$	$28.63 \pm 1.54\%$	$48.47 \pm 1.78\%$	$42.13 \pm 1.75\%$
	Adversarial	$28.93 \pm 1.62\%$	$30.73 \pm 1.66\%$	$42.23 \pm 1.85\%$	$40.17 \pm 1.76\%$
MAML-AD	Clean	$43.13 \pm 1.88\%$	$32.33 \pm 1.74\%$	$43.13 \pm 1.88\%$	$36.80 \pm 1.76\%$
	Adversarial	$32.47 \pm 1.60\%$	$37.87 \pm 1.74\%$	$37.63 \pm 1.64\%$	$37.13 \pm 1.75\%$
Matching Nets [2]	Clean	$43.87 \pm 0.41\%$	$30.02 \pm 0.39\%$	$43.88 \pm 0.48\%$	$36.14 \pm 0.40\%$
	Adversarial	$30.45 \pm 0.44\%$	$30.80 \pm 0.43\%$	$36.58 \pm 0.49\%$	$35.03 \pm 0.39\%$
Relation Nets [4]	Clean	$\mathbf{49.67 \pm 0.85\%}$	$32.32 \pm 0.58\%$	$\mathbf{49.45 \pm 0.84\%}$	$43.03 \pm 0.74\%$
	Adversarial	$32.59 \pm 0.79\%$	$32.85 \pm 0.63\%$	$42.98 \pm 0.85\%$	$40.89 \pm 0.79\%$
ADML (Ours)	Clean	$48.00 \pm 1.87\%$	$\mathbf{43.00 \pm 1.88\%}$	$48.00 \pm 1.87\%$	$\mathbf{43.20 \pm 1.70\%}$
	Adversarial	$\mathbf{40.10 \pm 1.73\%}$	$\mathbf{40.70 \pm 1.74\%}$	$\mathbf{44.00 \pm 1.83\%}$	$\mathbf{41.20 \pm 1.75\%}$

and each of them has 600 samples. CIFAR100 was created originally for object recognition tasks, whose data are suitable for meta-learning, and just like MiniImageNet, it has 100 classes, each of which contains 600 images. Similar as in [1], we considered 1-shot and 5-shot 5-way classification tasks. 5 samples per class were used for the inner gradient update during meta-training of a 5-shot learning model (one for 1-shot learning model). Thus K in ADML was set to 25 for 5-shot learning and 5 for 1-shot learning. 15 samples per class were used for the meta-update, thus we set $k = 75$. During the meta-testing, the learning model was trained using samples of 5 unseen classes, then we tested it by using it to classify new instances into these 5 classes. MiniImageNet was divided into 64, 16 and 20 classes for training, validation (for tuning hyperparamters) and testing respectively. We randomly sampled 5 classes from them to form each classification task. Since CIFAR100 has not been used for meta-learning before, we created the meta-learning version of CIFAR100, which has the same settings as MiniImageNet.

For FGSM [8], we leveraged a well-trained VGG16 network [22] for image classification (pre-trained on ImageNet [23] and CIFAR100 respectively) to generate adversarial samples. The parameter ϵ was set to 2 when generating adversarial samples for the meta-training, and was set to 2 and 0.2 for the meta-testing. Note that while the FGSM is leveraged to generate adversarial samples, the proposed ADML is agnostic to the particular choice of adversarial sample generation method.

We compared ADML against three representative meta-learning algorithms, including MAML [1], Matching Networks [2], and Relation Networks [4]. Moreover, for fair comparisons, we compared ADML with another adversarial meta-learner MAML-AD (introduced in the last section), which can be considered as a rather straightforward extension of MAML. For the implementation of ADML, we followed the architecture used by [1] for image embedding, which contains four 3×3 convolutional blocks with batch normalizations, ReLU activations and 2×2 max-poolings. In our experiments, we used the implementation at [24] for MAML, the Full Contextual Embeddings (FCE) implementation at [25] for Matching Networks, and the implementation at [26] for Relation Networks.

4.2 Experimental Results

To fully test the effectiveness of ADML, we conducted a comprehensive empirical study, which covers various possible cases. The experimental results on MiniImageNet and CIFAR100 are presented in Tables 1–2 and Tables 3–4 (in supplementary materials) respectively. Each entry in these tables gives the average classification accuracy (with 95% confidence intervals) of the corresponding test case, and the best results for each test case are marked in bold.

The experiments were conducted in six different test cases (combinations): "*Clean-Clean*", "*Clean-Adversarial*", "*Adversarial-Clean*", "*Adversarial-Adversarial*", "40%-*Clean*" and "40%-*Adversarial*". The first part of each combination (corresponding to a row) represents the training data used in the inner gradient update (or support set for Matching Networks and Relation Networks) during the meta-testing, while the second part (corresponding to a column) represents the testing data (or query set for Matching Networks and Relation Networks) for evaluation. "Clean" means clean samples only; "Adversarial" means adversarial samples only; and "40%" means that 40% samples of each class are adversarial and the rest 60% are clean, which represents intermediate cases. Note that the combinations, "40%-Clean" and "40%-Adversarial", do not exist for 1-shot

Table 2: Average classification accuracies on MiniImageNet (5-way, 5-shot)

Method	Meta-testing	$\epsilon = 2$		$\epsilon = 0.2$	
		Clean	Adversarial	Clean	Adversarial
MAML [1]	Clean	$61.45 \pm 0.91\%$	$36.65 \pm 0.88\%$	$61.47 \pm 0.91\%$	$53.05 \pm 0.86\%$
	40%	$56.74 \pm 0.93\%$	$43.05 \pm 0.86\%$	$59.25 \pm 0.91\%$	$54.67 \pm 0.91\%$
	Adversarial	$41.49 \pm 0.95\%$	$45.46 \pm 0.97\%$	$55.19 \pm 0.95\%$	$53.33 \pm 0.92\%$
MAML-AD	Clean	$57.13 \pm 0.96\%$	$41.65 \pm 0.92\%$	$57.09 \pm 0.96\%$	$49.71 \pm 0.88\%$
	40%	$54.07 \pm 0.91\%$	$48.74 \pm 0.91\%$	$56.52 \pm 0.90\%$	$52.08 \pm 0.90\%$
	Adversarial	$43.21 \pm 0.91\%$	$52.07 \pm 0.96\%$	$51.23 \pm 0.90\%$	$51.36 \pm 0.94\%$
Matching Nets [2]	Clean	$55.99 \pm 0.47\%$	$33.73 \pm 0.39\%$	$55.55 \pm 0.44\%$	$44.91 \pm 0.40\%$
	40%	$49.88 \pm 0.45\%$	$35.67 \pm 0.44\%$	$52.72 \pm 0.45\%$	$45.65 \pm 0.42\%$
	Adversarial	$36.24 \pm 0.45\%$	$37.91 \pm 0.40\%$	$47.77 \pm 0.45\%$	$46.19 \pm 0.44\%$
Relation Nets [4]	Clean	$\mathbf{63.85 \pm 0.73\%}$	$38.37 \pm 0.64\%$	$\mathbf{63.86 \pm 0.73\%}$	$55.39 \pm 0.68\%$
	40%	$56.53 \pm 0.77\%$	$41.04 \pm 0.67\%$	$59.02 \pm 0.70\%$	$55.06 \pm 0.68\%$
	Adversarial	$42.74 \pm 0.79\%$	$46.08 \pm 0.63\%$	$56.85 \pm 0.73\%$	$53.65 \pm 0.69\%$
ADML (Ours)	Clean	$59.38 \pm 0.99\%$	$\mathbf{57.03 \pm 0.98\%}$	$59.40 \pm 0.99\%$	$\mathbf{56.07 \pm 0.96\%}$
	40%	$\mathbf{58.12 \pm 0.90\%}$	$\mathbf{55.22 \pm 0.98\%}$	$\mathbf{59.67 \pm 0.89\%}$	$\mathbf{56.49 \pm 0.92\%}$
	Adversarial	$\mathbf{58.06 \pm 0.96\%}$	$\mathbf{55.27 \pm 0.92\%}$	$\mathbf{57.44 \pm 0.88\%}$	$\mathbf{54.47 \pm 0.93\%}$

learning since there is only one sample per class. Based on the results in Tables 1–2, we can make the following observations:

1) Just like MAML, ADML is indeed an effective meta-learner since it leads to quick learning from a small amount of new data for a new task. In the "Clean-Clean" cases, the general condition of meta-learning, ADML delivers desirable results, which are very close to the state-of-the-art given by MAML and Relation Networks, and consistently better than those of MAML-AD and Matching Networks. For example, in the case of 5-way 1-shot classification with $\epsilon = 2$ (Table 1), ADML gives an average classification accuracy of 48.00%, which is very close to that given by MAML (i.e., 48.47%) and Relation Networks (49.67%), and it performs better than MAML-AD (43.13%) and Matching Networks (43.87%). *Note that the proposed ADML focuses on the cases with adversarial samples, and it is reasonable that it performs slightly worse on the cases with only clean samples.*

2) ADML is robust to adversarial samples since it only suffers from a minor performance degradation when encountering adversarial samples. For example, for the 5-way 5-shot classification with $\epsilon = 2$ (Table 2), ADML gives classification accuracies of 57.03%, 58.06%, 55.27%, 58.12% and 55.22% in the five test cases respectively. Compared to the "Clean-Clean" case (i.e., 59.38%), the performance degradation is only 4.16% in the worst-case and 2.64% on average. However, the classification accuracies given by the other meta-learning algorithms, including MAML-AD, drop substantially when there are adversarial samples. For example, for MAML, the accuracy drops from 61.45% ("Clean-Clean") to 36.65% ("Clean-Adversarial") when injecting adversarial samples for testing, which represents a significant degradation of 24.80%. Similar observations can be made for MAML-AD, Matching Networks and Relation Networks, which represent substantial degradations of 15.48%, 22.26% and 25.48% respectively.

3) ADML consistently outperforms all the other meta-learning algorithms in the test cases with adversarial samples. For instance, in the "Clean-Adversarial" cases of 5-way 5-shot learning with $\epsilon = 2$ (Table 2), ADML achieves an accuracy of 57.03%, which represents 20.38%, 15.38%, 23.30% and 18.66% improvements over MAML, MAML-AD, Matching Networks and Relation Networks respectively. This clearly shows the superiority of the adversarial meta-training procedure of the proposed ADML compared to MAML-AD, the straightforward adversarial meta-learner. Even so, MAML-AD still generally performs better than the rest meta-training algorithms, when dealing with adversarial samples.

4) When the perturbation of adversarial samples escalates, ADML maintains stable performance. For example, for 5-way 1-shot learning, when ϵ increases from 0.2 to 2 (Table 1), ADML only leads to minor degradations of 0.2%, 3.9% and 0.5% in the corresponding three cases involving adversarial samples. However, much more significant degradations can be observed when the other meta-learning algorithms are applied. For instance, the accuracies of MAML suffer from 13.50%, 13.30% and 9.44% drops in these three cases when increasing ϵ from 0.2 to 2.

5) As we might expect, when lower perturbations exerted, the adversarial samples are not significantly different from the corresponding clean samples, which brings about relatively close results

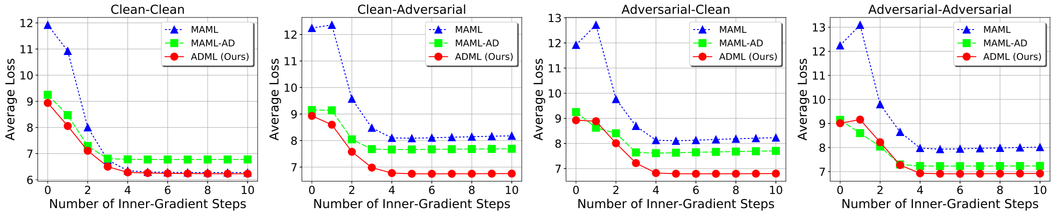

Figure 2: Average loss over the gradient update step for 5-way 1-shot learning on MiniImageNet

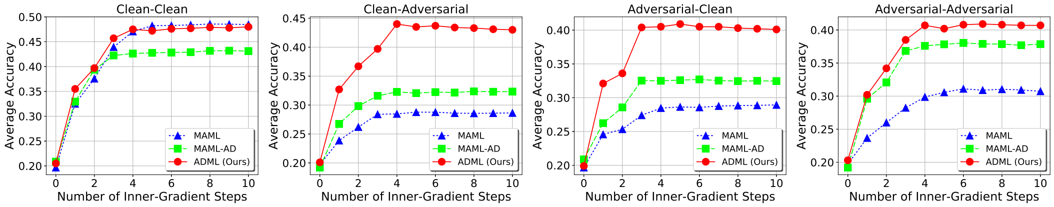

Figure 3: Top-1 accuracy over the gradient update step for 5-way 1-shot learning on MiniImageNet

in different test cases, not only for ADML, but also for the other meta-learning algorithms. For instance, for the 5-way 5-shot classification with $\epsilon = 0.2$ (Table 2), MAML gives a smaller gap of 8.42% between "Clean-Clean" and "Clean-Adversarial", compared with that of 24.80% when $\epsilon = 2$.

6) As expected, the classification accuracy increases dramatically when going from 1-shot to 5-shot learning. Particularly, when $\epsilon = 2$, if we do 5-shot learning with ADML, we can achieve classification accuracies of 59.38%, 57.03%, 58.06% and 55.27% in the corresponding four test cases respectively, which represent 11.38%, 14.03%, 17.96% and 14.57% improvements over 1-shot learning. This observation implies that more training samples (even if they may be adversarial samples) lead to better classification accuracies.

Similar observations can be made for the results corresponding to CIFAR100 (i.e., Tables 3–4 in supplementary materials). In addition, we show how the loss and Top-1 accuracy change in Figures 2 and 3 during the meta-testing. Specifically, these two figures show that when ADML, MAML and MAML-AD are applied on MiniImageNet, how the losses and Top-1 accuracies change with the gradient update step during the meta-testing in the four test cases of 5-way 1-shot learning with $\epsilon = 2$. We observe that, for all the cases, the losses of the models learned with ADML drop sharply after only several gradient updates, and stabilize at small values during the meta-testing, which are generally lower than those of the other two methods. Moreover, the Top-1 accuracies of the models learned with ADML rise sharply after only several gradient updates, and stabilize at values, which are generally higher than those of the other two methods (a little bit lower than that of MAML in "Clean-Clean" case). *Similar trends can be observed in Figures 4–5 in supplementary materials for 5-way 5-shot learning with $\epsilon = 2$.* These observations further confirm that ADML is suitable for meta-learning since it can quickly learn and adapt from small data for a new task through only a few gradient updates.

5 Conclusions

In this paper, we proposed a novel method called ADML (ADversarial Meta-Learner) for meta-learning with adversarial samples, which features an *adversarial* way for optimizing model parameters $\boldsymbol{\theta}$ during meta-training through the arm-wrestling between inner gradient update and meta-update using both clean and adversarial samples. A comprehensive empirical study has been conducted for performance evaluation using two widely-used datasets, MiniImageNet and CIFAR100. The extensive experimental results have showed that 1) ADML is an effective meta-learner even in the cases with only clean samples; 2) a straightforward adversarial meta-learner, namely, MAML-AD, does not work well with adversarial samples; in addition, 3) ADML is robust to adversarial samples and outperforms other meta-learning algorithms including MAML on adversarial meta-learning tasks; and most importantly, 4) it opens up an interesting research direction and sheds light on dealing with the difficult cases with limited and even contaminated samples.

Broader Impact

This work presents a novel method called ADML (ADversarial Meta-Learner) for meta-learning to deal with the cases involving adversarial samples. The proposed algorithm sheds light on tackling the cases with limited and even contaminated samples, which are challenging but common in real life. For example, for some rare events, there may be only a small number of photos taken in a bad environment (such as mist, rain, etc) which may pose challenges on the learning ability of a model. The proposed ADML is well suited for such cases. Meanwhile, ADML may have some potential negative impact. The failure of ADML can cause incorrect classifications, which may lead to false alarms or wrong decision-makings that could further result in extra and unnecessary labor work. In addition, our task/method does not leverage biases in the data.

References

[1] Chelsea Finn, Pieter Abbeel, and Sergey Levine. Model-agnostic meta-learning for fast adaptation of deep networks. In *ICML*, pages 1126–1135. JMLR. org, 2017.

[2] Oriol Vinyals, Charles Blundell, Timothy Lillicrap, Daan Wierstra, et al. Matching networks for one shot learning. In *NIPS*, pages 3630–3638, 2016.

[3] Jake Snell, Kevin Swersky, and Richard Zemel. Prototypical networks for few-shot learning. In *NIPS*, pages 4077–4087, 2017.

[4] Flood Sung, Yongxin Yang, Li Zhang, Tao Xiang, Philip HS Torr, and Timothy M Hospedales. Learning to compare: Relation network for few-shot learning. In *Proceedings of the IEEE CVPR*, pages 1199–1208, 2018.

[5] Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian J. Goodfellow, and Rob Fergus. Intriguing properties of neural networks. In *ICLR*, 2014.

[6] Ruixiang Zhang, Tong Che, Zoubin Ghahramani, Yoshua Bengio, and Yangqiu Song. Metagan: An adversarial approach to few-shot learning. In *NIPS*, pages 2365–2374, 2018.

[7] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *NIPS*, pages 2672–2680, 2014.

[8] Ian J. Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples. In *ICLR*, 2015.

[9] Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. 2009.

[10] Devang K Naik and Richard J Mammone. Meta-neural networks that learn by learning. In *IJCNN*, volume 1, pages 437–442. IEEE, 1992.

[11] Sebastian Thrun and Lorien Pratt. *Learning to learn*. Springer Science & Business Media, 1998.

[12] Gregory Koch, Richard Zemel, and Ruslan Salakhutdinov. Siamese neural networks for one-shot image recognition. In *ICML*, volume 2. Lille, 2015.

[13] Marcin Andrychowicz, Misha Denil, Sergio Gomez, Matthew W Hoffman, David Pfau, Tom Schaul, Brendan Shillingford, and Nando De Freitas. Learning to learn by gradient descent by gradient descent. In *NIPS*, pages 3981–3989, 2016.

[14] Sachin Ravi and Hugo Larochelle. Optimization as a model for few-shot learning. In *ICLR*, 2017.

[15] Adam Santoro, Sergey Bartunov, Matthew Botvinick, Daan Wierstra, and Timothy Lillicrap. Meta-learning with memory-augmented neural networks. In *ICML*, pages 1842–1850, 2016.

[16] Nikhil Mishra, Mostafa Rohaninejad, Xi Chen, and Pieter Abbeel. A simple neural attentive meta-learner. In *ICLR*, 2018.

[17] Alex Graves, Greg Wayne, and Ivo Danihelka. Neural turing machines. *arXiv preprint arXiv:1410.5401*, 2014.

[18] Andras Rozsa, Manuel Günther, and Terrance E Boult. Are accuracy and robustness correlated. In *Proceedings of the IEEE ICMLA*, pages 227–232. IEEE, 2016.

[19] Nicolas Papernot, Patrick McDaniel, Ian Goodfellow, Somesh Jha, Z Berkay Celik, and Ananthram Swami. Practical black-box attacks against machine learning. In *Asia CCS*, pages 506–519, 2017.

[20] Alexey Kurakin, Ian J. Goodfellow, and Samy Bengio. Adversarial machine learning at scale. In *ICLR*, 2017.

[21] Takeru Miyato, Andrew M. Dai, and Ian J. Goodfellow. Adversarial training methods for semi-supervised text classification. In *ICLR*, 2017.

[22] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. In *ICLR*, 2015.

[23] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, Alexander C. Berg, and Li Fei-Fei. ImageNet Large Scale Visual Recognition Challenge. *IJCV*, 115(3):211–252, 2015.

[24] Chelsea Finn. `https://github.com/cbfinn/maml`, 2017.

[25] Albert Berenguel Centeno. `https://github.com/gitabcworld/MatchingNetworks`, 2017.

[26] Flood Sung. `https://github.com/floodsung/LearningToCompare_FSL`, 2018.

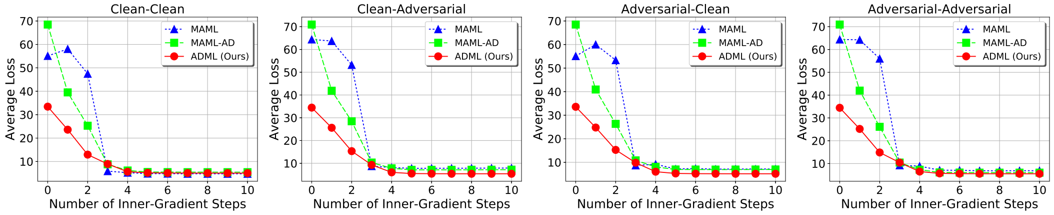

Figure 4: Average loss over the gradient update step for 5-way 5-shot learning on MiniImageNet

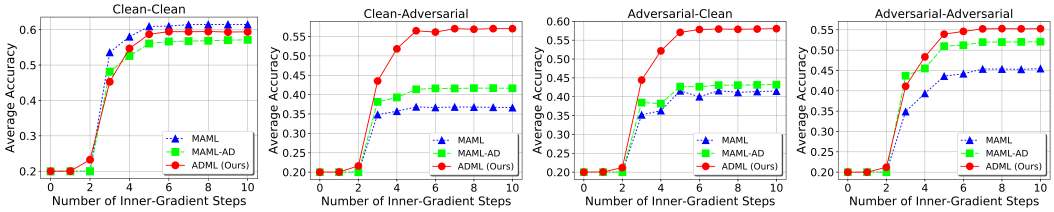

Figure 5: Top-1 accuracy over the gradient update step for 5-way 5-shot learning on MiniImageNet

Table 3: Average classification accuracies on CIFAR100 (5-way, 1-shot)

Method	Meta-testing	$\epsilon = 2$		$\epsilon = 0.2$	
		Clean	Adversarial	Clean	Adversarial
MAML [1]	Clean	57.67 ± 1.76%	26.40 ± 1.55%	57.67 ± 1.76%	43.30 ± 1.68%
	Adversarial	28.13 ± 1.56%	28.23 ± 1.64%	43.03 ± 1.76%	39.00 ± 1.70%
MAML-AD	Clean	52.70 ± 1.89%	36.20 ± 1.65%	52.70 ± 1.89%	39.17 ± 1.82%
	Adversarial	37.27 ± 1.72%	41.67 ± 1.86%	37.80 ± 1.70%	37.60 ± 1.78%
Matching Nets [2]	Clean	47.94 ± 0.56%	25.06 ± 0.36%	47.68 ± 0.52%	39.03 ± 0.51%
	Adversarial	24.82 ± 0.46%	27.72 ± 0.43%	40.08 ± 0.57%	37.79 ± 0.44%
Relation Nets [4]	Clean	**58.68 ± 0.92%**	31.11 ± 0.93%	**58.72 ± 0.90%**	45.03 ± 0.76%
	Adversarial	30.85 ± 0.92%	30.52 ± 0.59%	45.85 ± 1.01%	41.40 ± 0.80%
ADML (Ours)	Clean	55.70 ± 2.00%	**50.90 ± 1.84%**	55.70 ± 2.00%	**49.30 ± 1.76%**
	Adversarial	**54.50 ± 1.69%**	**50.60 ± 1.83%**	**52.90 ± 1.92%**	**45.00 ± 1.79%**

Table 4: Average classification accuracies on CIFAR100 (5-way, 5-shot)

Method	Meta-testing	$\epsilon = 2$		$\epsilon = 0.2$	
		Clean	Adversarial	Clean	Adversarial
MAML [1]	Clean	74.03 ± 0.89%	31.29 ± 0.78%	74.03 ± 0.89%	54.15 ± 1.00%
	40%	65.69 ± 0.92%	36.14 ± 0.84%	68.99 ± 0.94%	55.79 ± 0.98%
	Adversarial	33.34 ± 0.90%	43.66 ± 0.86%	59.08 ± 1.00%	53.93 ± 0.96%
MAML-AD	Clean	67.71 ± 0.96%	44.61 ± 0.90%	67.73 ± 0.96%	56.07 ± 0.95%
	40%	64.85 ± 0.90%	53.59 ± 0.88%	65.93 ± 0.93%	57.96 ± 0.93%
	Adversarial	48.37 ± 0.99%	58.92 ± 0.97%	59.45 ± 1.00%	56.33 ± 0.98%
Matching Nets [2]	Clean	62.95 ± 0.46%	28.14 ± 0.37%	62.58 ± 0.49%	47.14 ± 0.45%
	40%	54.39 ± 0.48%	28.64 ± 0.36%	57.86 ± 0.48%	47.01 ± 0.48%
	Adversarial	29.40 ± 0.44%	32.77 ± 0.42%	53.34 ± 0.52%	46.50 ± 0.46%
Relation Nets [4]	Clean	**75.52 ± 0.66%**	35.37 ± 0.55%	**75.22 ± 0.70%**	55.75 ± 0.68%
	40%	66.85 ± 0.79%	36.70 ± 0.54%	68.67 ± 0.80%	55.33 ± 0.69%
	Adversarial	40.46 ± 0.88%	39.82 ± 0.57%	60.52 ± 0.82%	55.50 ± 0.69%
ADML (Ours)	Clean	69.90 ± 0.88%	**65.68 ± 0.87%**	69.90 ± 0.88%	**59.15 ± 0.90%**
	40%	**67.61 ± 0.93%**	**62.83 ± 0.88%**	**69.20 ± 0.88%**	**60.44 ± 0.93%**
	Adversarial	**65.26 ± 0.98%**	**64.18 ± 0.86%**	**61.93 ± 0.95%**	**59.80 ± 0.84%**