Model-Agnostic Boundary-Adversarial Sampling for Test-Time Generalization in Few-Shot learning

Jaekyeom Kim, Hyoungseok Kim, and Gunhee Kim

Computer Science and Engineering, Seoul National University, Seoul, Korea
{jaekyeom,harry2636,gunhee}@snu.ac.kr,
http://vision.snu.ac.kr/projects/mabas

Abstract. Few-shot learning is an important research problem that tackles one of the greatest challenges of machine learning: learning a new task from a limited amount of labeled data. We propose a model-agnostic method that improves the test-time performance of any few-shot learning models with no additional training, and thus is free from the training-test domain gap. Based on only the few support samples in a meta-test task, our method generates the samples adversarial to the base few-shot classifier's boundaries and fine-tunes its embedding function in the direction that increases the classification margins of the adversarial samples. Consequently, the embedding space becomes denser around the labeled samples which makes the classifier robust to query samples. Experimenting on miniImageNet, CIFAR-FS, and FC100, we demonstrate that our method brings significant performance improvement to three different base methods with various properties, and achieves the state-of-the-art performance in a number of few-shot learning tasks.

Keywords: Few-shot learning, meta-learning, adversarial learning

1 Introduction

One of the greatest challenges for machine intelligence to meet human intelligence is the ability to quickly adapt to novel tasks. Humans learn how to solve new tasks with only a small amount of training, by taking advantage of the prior information they have learned for lives. Few-shot learning [11, 28] is a research problem to make the most of knowledge gained during training to deal with novel tasks with a limited number of labeled samples. Its core difficulty lies in the data deficiency for novel tasks.

In the few-shot learning problem, the base classes in training and novel classes in test are disjoint. The test phase consists of multiple tasks where each contains a small labeled *support* set and an unlabeled *query* set. The goal of each task is to predict the labels of the query data based on the support data. The meta-learning scheme [36, 12], which forms batches of tasks for the training as well, has become dominant in this field of research. Thus, the training and test phases are often called the *meta-training* and *meta-test* phases. Also, many of modern

few-shot learning methods use embedding functions (or feature extractors) such as ResNet-12 and handle the data in the embedding space.

With a recent surge of interest in few-shot learning, there have been various approaches proposed, including distance metric methods [45, 40], meta-learning methods [12, 38], and data augmentation methods [17, 39]. Among them, the data augmentation (or hallucination) methods augment the support set by generating fake labeled data for few-shot learning methods (referred to as *base models* or *base classifiers*) [7, 17, 39, 46, 51]. The key merit of this approach is applicability to a wide range of base classifiers since it can directly generate fake labeled data. However, most previous approaches have some limitations. (i) Such methods *learn* to generate additional examples with the meta-training set, and thus may not be effective if meta-test domains are far from the meta-training domain. (ii) Since they do not update the trained parameters of the base classifier models at test time, they have no chance to correct the errors that exist in the base classifiers (*e.g.* overfitting of the embedding functions to the meta-training set). (iii) These methods need to be re-trained for each base classifier to generate fake labeled data optimal for the base model.

In this work, we propose a novel model-agnostic sample generation approach for few-shot learning that does not suffer from the aforementioned limitations. The keys to our method named MABAS (*Model-Agnostic Boundary-Adversarial Sampling*) are to perform *no training* for data generation and to generate samples for embedding function fine-tuning. Given only the few labeled data (*i.e.* support samples) in a meta-test task, it creates samples adversarial to the classification boundaries of the base model targeting every meta-test class using each support sample. It then updates the embedding function in the direction that increases the classification margins of the adversarial samples; as a result, the embedding space becomes denser around the labeled samples, which makes the classifier robust to query samples. For sample generation in the embedding space is more advantageous rather than in image space for generalization to unseen classes, since adversarial gradients in the embedding space can directly attack the classification margins.

Finally, we can summarize the main contributions of this work as follows.

1. To the best of our knowledge, our approach is the first pure test-time method for few-shot learning that generates samples for embedding function fine-tuning without learning how to create samples. It simply creates samples adversarial to the classification boundaries of the base few-shot model and fine-tunes the embedding function using the new samples to improve the few-shot generalization performance.
2. Our approach is free from the training-test domain gap and integrable with any base classifier models. We apply our approach to three representative few-shot learning methods, including MetaOptNet [22], Few-Shot without Forgetting (FSwF) [16] and standard transfer learning (STL) [6, 42].
3. Our experiments demonstrate that MABAS provides all of the three few-shot learners with significant performance gains and achieves the state-of-the-art

performance in a number of tasks on three benchmarks: miniImageNet [45], CIFAR-FS [3] and FC100 [34].

2 Related Work

2.1 Few-shot Learning

We review a large body of existing few-shot learning methods in three categories: distance metric, meta-learning, and data augmentation methods as follows.

Distance metric methods tackle the few-shot learning problem by learning distance metrics to measure more similar images closer. Matching networks [45] and Prototypical networks [40] predict the labels of the query data based on their learned distances to the support samples in the embedding space. Relational networks [43] propose the embedding and relation module to learning to compare query data with support samples.

Meta-learning methods deal with the problem using the *learning to learn* paradigm, in which an outer loop optimizes meta-variables that controls the optimization of model parameters in an inner loop. MAML [12] proposes the objective of the outer-loop that can learn a good initialization for the inner-loop few-shot learners. LEO [38] introduces latent meta-variables for neural network parameters and take gradient steps within the low-dimensional latent space instead of the high-dimensional parameter space. MetaOptNet [22] uses a convex learner such as multi-class SVMs in the inner loop and update the embedding function in the outer loop to be optimal for the inner loop. MTL [42] utilizes scaling and shifting parameters to adapt the learned embedding parameters to each task differently. LGM-Net [23] encodes prior knowledge of tasks into the context encoder to generate task-specific function weights of embedding networks.

Data augmentation methods learn to augment data to resolve the problem of few-shot learning that lacks enough labeled data. Hariharan and Girshick [17] use the modes of intra-class variation of the base training classes to generate additional samples for the novel classes. $\triangle$-encoder [39] employs auto-encoders that learn to extract transferable intra-class deformations from training data. Zhang *et al*. [51] introduce a saliency map extractor that separates foregrounds with backgrounds to hallucinate datapoints. Chen *et al*. [7] propose a deformation network that fuses few-shot images with unlabeled images. Wang *et al*. [46] combine a meta-learner with a generative model to produce imaginary examples from an anchor example. The existing data augmentation methods share the limitations introduced in Section 1. Our method is free from the issues since it does not rely on learning from training data to generate samples. That our method generates samples for embedding function fine-tuning instead of simply enlarging support sets, is also a fundamental difference.

2.2 Adversarial Learning

Since neural network classifiers were known to be vulnerable to even small input perturbations [44], adversarial learning has been actively studied. Adversarial

attack methods aim at generating adversarial examples by adding perturbations to samples to fail the classifier [25, 31, 21, 5, 4, 48, 41]. Few-shot adversarial learning methods exploit adversarial signals from discriminators and generators to augment the few-shot classes [27, 1, 8, 2, 35, 53, 47, 30, 10]. Mottian *et al*. [32] design a multi-class adversarial discriminator to address the supervised adaptation problem in the few-shot domain. Zhang *et al*. [52] employ a generative adversarial network (GAN) that produces fake samples to make sharper decision boundaries. Gao *et al*. [14] model the latent distribution of novel classes with adversarial networks by preserving the covariance information.

Compared to the existing few-shot adversarial learning approaches, we propose to generate adversarial samples purely in test time with no training of additional models. Consequently, our method is orthogonal and easily adaptable to any other few-shot learning methods.

3 MABAS: Boundary-Adversarial Sample Generation

We briefly review the formulation of the few-shot classification (Section 3.1) and the idea of test-time fine-tuning of embedding functions (Section 3.2). We then propose MABAS as an adversarial learning approach to adaptively fine-tuning the embedding function to each meta-test task (Section 3.3).

3.1 The Few-Shot Classification Problem

We begin with the formulation of few-shot classification following previous work [22, 36]. The meta-test phase of the few-shot classification problem is comprised of I tasks (*i.e.* episodes): $\mathcal{D}^{\text{test}} = \{\mathcal{T}_i^{\text{test}}\}_{i=1}^{I}$. The i-th task $\mathcal{T}_i^{\text{test}} = (\mathcal{S}_i^{\text{test}}, \mathcal{Q}_i^{\text{test}})$ consists of two sets of data: the support set $\mathcal{S}_i^{\text{test}}$ and the query set $\mathcal{Q}_i^{\text{test}}$. Each task is a K-way M-shot classification problem; the support $\mathcal{S}_i^{\text{test}}$ consists of K different classes, each of which contains M labeled samples (*i.e.* $|\mathcal{S}_i^{\text{test}}| = KM$). The meta-training dataset $\mathcal{D}^{\text{train}}$ consists of the classes disjoint with the meta-test set $\mathcal{D}^{\text{test}}$. For a classifier $\mathcal{C}$ trained with $\mathcal{D}^{\text{train}}$, the meta-test accuracy is defined as $\sum_{i=1}^{I} \sum_{(\boldsymbol{x},y)\in\mathcal{Q}_i^{\text{test}}} \mathbb{1}\left(\mathcal{C}\left(\boldsymbol{x}|\mathcal{S}_i^{\text{test}}\right) = y\right)$.

3.2 Test-time Fine-tuning of Embedding Functions

Since each meta-test task consists of the classes that are never seen during meta-training, it is a common approach in few-shot learning to fine-tuning the learned parameters using the support samples of the novel task. For instance, MAML [12] and LEO [38] use the trained model as initialization and fine-tuning it to meta-test tasks, and MTL [42] applies scaling and shifting to the learned parameters for each meta-test task differently, which could be better than direct update of parameters to reduce the overfitting to a small number of samples.

Likewise, our approach aims at fine-tuning the learned parameters of the base few-shot learner adaptively to novel tasks. However, we limit to update the parameters of the *embedding function* (or the feature extractor). It is a

universally applicable idea since most recent few-shot learning models employ CNNs as their embedding functions that extract features from images [22, 29].

We formulate the iterative fine-tuning procedure of the embedding function f_ϕ with parameter $\phi \in \psi$ for the classifier $\mathcal{C}$ defined by $\mathcal{C}\left(\boldsymbol{x}|\mathcal{S}^{\text{test}}\right) = \text{argmax}_k\, h\left(f_\phi(\boldsymbol{x}), k|\psi, \mathcal{S}^{\text{test}}\right)$, as follows. For a given fine-tuning loss function $\mathcal{L}^{\text{fine-tune}}(\mathcal{S}, \phi)$, we update the classifier $\mathcal{C}$ as

$$\mathcal{C}\left(\boldsymbol{x}|\mathcal{S}^{\text{test}}\right) = \underset{k}{\text{argmax}}\, h\left(f_{\phi'}(\boldsymbol{x}), k|\psi', \mathcal{S}^{\text{test}}\right) \tag{1}$$

where $\phi' \in \psi'$. The fine-tuning loss $\mathcal{L}^{\text{fine-tune}}(\mathcal{S}, \phi)$ and the score function h differ according to the base models, and their definitions will be described in Section 4. For simplicity, we denote $\boldsymbol{z} = f_\phi(\boldsymbol{x})$. The new parameter ϕ' is obtained by multiple updates via gradient descent:

$$\phi^i = \phi^{i-1} - \beta \cdot \nabla_{\phi^{i-1}} \left(\mathcal{L}^{\text{fine-tune}}(\mathcal{S}^{\text{test}}, \phi^{i-1})\right), \tag{2}$$

for $i = 1, \ldots, U$ where U is the number of updates and β is a fine-tuning step size. We initialize $\phi^0 = \phi$ with the parameter learned during meta-training and finally set $\phi' = \phi^U$.

3.3 Fine-tuning by Boundary-Adversarial Samples

We assume that a base few-shot classifier is chosen and trained with training data. Our approach solely focuses on meta-test time; it first generates samples adversarial to the classification boundaries defined by the few-shot classifier in the embedding space, and use them to fine-tune only the parameter ϕ of the embedding function. Figure 1 intuitively visualizes how our approach works. For the success of few-shot learning, it is important to transfer the embedding function to the domain that lacks labeled samples. We generate boundary-adversarial samples by moving every support sample toward each of the classification boundaries. The embedding function is fine-tuned in the direction that increases the margins of the adversarial samples. After the update, the data embeddings are denser around the support embeddings, and thus the recomputed classification boundaries better separate the queries from different classes.

Generation of adversarial samples. We first define the classification margin for sample $\boldsymbol{z}$ between classes k and k' as

$$m\left(\boldsymbol{z}, k, k'|\psi, \mathcal{S}\right) \coloneqq h\left(\boldsymbol{z}, k|\psi, \mathcal{S}\right) - h\left(\boldsymbol{z}, k'|\psi, \mathcal{S}\right). \tag{3}$$

For every support sample $(\boldsymbol{x}, y) \in \mathcal{S}$ and each attack target class $k' \in \{1, \ldots, K\} \backslash y$, we create a boundary-adversarial sample $\boldsymbol{z}^{\text{adv}}_{y,k'}$ by moving $\boldsymbol{z} = f_\phi(\boldsymbol{x})$ in the direction that minimizes its margin against k' in the embedding space:

$$\boldsymbol{z}^{\text{adv}}_{y,k'} \coloneqq \boldsymbol{z} - \delta \cdot \nabla_{\boldsymbol{z}}\, m\left(\boldsymbol{z}, y, k'|\psi, \mathcal{S}\right) \tag{4}$$

where δ is a step size. $\boldsymbol{z}^{\text{adv}}_{y,k'}$ is obtained by applying single-step gradient descent to $\boldsymbol{z}$ in the direction that minimizes the margin (or score gap) of the embedding

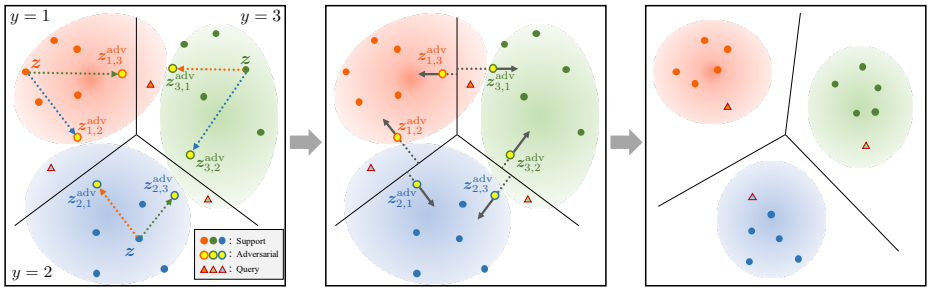


Fig. 1. Conceptual visualization of our MABAS approach in the embedding space. The solid circles, triangles, and yellow circles are the support, query, and boundary-adversarial samples, respectively. Each shaded area represents a region to which most samples from the class are mapped by the embedding function. The black lines are the classification boundaries computed from the supports. *Left*: The boundary-adversarial samples are generated by moving each support sample toward the classification boundaries (Equation (4)). *Middle*: The embedding function is updated in the direction that increases the margins of the adversarial samples (Equation (6)) while holding the support embeddings (Equation (5)). *Right*: After the fine-tuning of the embedding function, data embeddings are denser around the support embeddings, and the classification boundaries are updated to better separate queries from different classes.

between y and k'. The single-step update is sufficient for the generation as the fine-tuning process is alternation between the adversarial sample generation and the embedding function update. The resulting $\boldsymbol{z}^{\text{adv}}_{y,k'}$ is an adversarial sample based on $\boldsymbol{z}$ against the target class k'. It can be regarded as an augmented data of class y located near to the classification boundary between y and k'. When deriving the adversarial gradient in Equation (4), we fix h even in the case of differentiable base learners. We will elaborate it with some examples of differentiable base models in Section 4.

In the meta-test task, for every $(\boldsymbol{x}, y) \in \mathcal{S}$, $K-1$ adversarial samples are generated one for each target class ($\{1, \ldots, K\} \setminus y$), and thus $|\mathcal{S}|(K-1) = MK(K-1)$ adversarial samples are created in total at each fine-tuning step.

Fine-tuning of the embedding function. With the adversarial samples, we update the parameter ϕ of the embedding function via the gradient descent in Equation (2). The fine-tuning loss $\mathcal{L}^{\text{fine-tune}}$ is defined as

$$\mathcal{L}^{\text{fine-tune}}(\mathcal{S}, \phi) \coloneqq \mathcal{L}^{\text{adv}}(\mathcal{S}, \phi) + \eta \cdot \frac{1}{|\mathcal{S}|} \sum_{(\boldsymbol{x}, y) \in \mathcal{S}} \|f_\phi(\boldsymbol{x})\|^2, \tag{5}$$

whose first term is the adversarial loss term and second term is the regularizer with a coefficient η. We define $\mathcal{L}^{\text{adv}}$ as

$$\mathcal{L}^{\text{adv}}(\mathcal{S}, \phi) \coloneqq \frac{1}{|\mathcal{S}|} \sum_{(\boldsymbol{x}, y) \in \mathcal{S}} \left[\frac{1}{K-1} \sum_{k' \neq y} \left\{ \alpha_{\boldsymbol{x}, y} - \min_{k \neq y} m\left(\boldsymbol{z}^{\text{adv}}_{y,k'}, y, k | \psi, \mathcal{S}\right) \right\}_+ \right] \tag{6}$$

where $\alpha_{\boldsymbol{x},y} = \frac{1}{K-1}\sum_{k\neq y} m(\boldsymbol{z}, y, k|\psi, \mathcal{S})$. The objective $\mathcal{L}^{\text{adv}}$ chooses the minimum margin per adversarial sample and increases the margin but not larger than the anchor $\alpha_{\boldsymbol{x},y}$. The anchor $\alpha_{\boldsymbol{x},y}$ is the average margin for $(\boldsymbol{x}, y)$ and provides a reasonable upper limit for the adversarial samples' margins. We treat $\alpha_{\boldsymbol{x},y}$ as a constant threshold rather than letting fine-tuning gradients flow through $\alpha_{\boldsymbol{x},y}$, since otherwise it might dominate the objective and disturb the pushing of the adversarial samples against the boundaries.

The second regularization term in Equation (5) not only prevents the excessive expansion of the supports' embedding space but also stabilizes the updates of the embedding space.

4 Application to Various Few-Shot Methods

To show the flexibility and generality of our MABAS approach, we apply it to three representative few-shot learning methods, including MetaOptNet [22], Few-Shot without Forgetting (FSwF) [16] and the standard transfer learning (STL) method [6, 42]. These methods show diverse characteristics; MetaOptNet and FSwF have differentiable base learners while the STL does not. Also, the classifiers of MetaOptNet and STL are linear, whereas FSwF is not. In this section, we present the key ideas of each base method and how our approach is integrated with them.

4.1 MetaOptNet

Original methodology. MetaOptNet [22] uses a differentiable SVM solver for few-shot classification. In each meta-training or meta-test task, MetaOptNet solves the multi-class SVM problem for the support data to make predictions for the query data. More specifically, given a task with support $\mathcal{S}$ and query $\mathcal{Q}$, it solves the K-class SVM problem [9] for support samples $(\boldsymbol{x}_n, y_n) \in \mathcal{S},\ n = 1, \ldots, N$, whose objective is defined by

$$\begin{aligned} &\underset{\boldsymbol{w}_1,\ldots,\boldsymbol{w}_K,\xi_1,\ldots,\xi_N}{\text{minimize}} \ \frac{1}{2}\sum_k \|\boldsymbol{w}_k\|_2^2 + C\sum_n \xi_n \\ &\text{s.t.} \ \ (\boldsymbol{w}_{y_n} - \boldsymbol{w}_k)^\top f_\phi(\boldsymbol{x}_n) \geq 1 - \mathbb{1}(y_n = k) - \xi_n, \ \ \forall n, k. \end{aligned} \tag{7}$$

The score function for the task is defined using the SVM solution $\boldsymbol{w}_1, \ldots, \boldsymbol{w}_K$:

$$h\left(f_\phi(\boldsymbol{x}), k|\psi, \mathcal{S}\right) \coloneqq \boldsymbol{w}_k^\top f_\phi(\boldsymbol{x}). \tag{8}$$

The parameter ϕ of the embedding function f_ϕ is trained with the classification loss in the meta-training phase, and not updated in the meta-test time.

Boundary-adversarial fine-tuning. By plugging Equation (8) into Equations (4) and (6), we obtain $\boldsymbol{z}^{\text{adv}}_{y,k'}$ and $\mathcal{L}^{\text{adv}}$ for MetaOptNet as

$$\boldsymbol{z}^{\text{adv}}_{y,k'} = \boldsymbol{z} - \delta \cdot \nabla_{\boldsymbol{z}} \left((\boldsymbol{w}_y - \boldsymbol{w}_{k'})^\top \boldsymbol{z}\right) = \boldsymbol{z} - \delta \cdot (\boldsymbol{w}_y - \boldsymbol{w}_{k'}), \tag{9}$$

$$\mathcal{L}^{\text{adv}}(\mathcal{S}, \phi) = \frac{1}{|\mathcal{S}|}\sum_{(\boldsymbol{x},y)\in\mathcal{S}} \left[\frac{1}{K-1}\sum_{k'\neq y}\left\{\alpha_{\boldsymbol{x},y} - \min_{k\neq y}\left((\boldsymbol{w}_y - \boldsymbol{w}_k)^\top \boldsymbol{z}^{\text{adv}}_{y,k'}\right)\right\}_+\right].$$

As mentioned in Section 3.3, although $\boldsymbol{w}_1, \ldots, \boldsymbol{w}_K$ are differentiable with respect to $\boldsymbol{z}$, we assume they are fixed constants to derive the attack gradient as $\nabla_{\boldsymbol{z}}((\boldsymbol{w}_y - \boldsymbol{w}_{k'})^\top \boldsymbol{z}) = \boldsymbol{w}_y - \boldsymbol{w}_{k'}$ in Equation (9). Only the embedding function parameter ϕ is fine-tuned iteratively, as described in Section 3.

Variation. We also apply our approach to a modified version of MetaOptNet, which uses the SVM solution for $\{(\frac{1}{M}\sum_{n:y_n=k} f_\phi(\boldsymbol{x}_n), k)|$ for $k = 1, \ldots, K\}$ instead of $\{(f_\phi(\boldsymbol{x}), y)|(\boldsymbol{x}, y) \in \mathcal{S}\}$. The difference is that K class prototypes (*i.e.* per-class average embeddings of the supports) are used instead of KM support embeddings as input to the multi-class SVM solver, inspired by [40]. Except it, the derivation is the same. This variation solves the multi-class SVM problem with fewer samples but still shows competitive results. We refer to this variation as MetaOptNet-Proto for the rest of this paper.

Setting of δ. Although fixing an adversarial step size δ in Equation (9) works well with MetaOptNet(-Proto), we can formulate δ so that the change of the margin per adversarial step is fixed at λ, using the fact that the multi-class SVM is linear and the maximum margin is 1:

$$\boldsymbol{z}^{\text{adv}}_{y,k'} = \boldsymbol{z} - \delta_{y,k'} \cdot (\boldsymbol{w}_y - \boldsymbol{w}_{k'}), \quad \text{where } \delta_{y,k'} = \frac{\lambda}{\|\boldsymbol{w}_y - \boldsymbol{w}_{k'}\|^2}. \tag{10}$$

With this definition, $m(\boldsymbol{z}, y, k'|\psi, \mathcal{S}) - m(\boldsymbol{z}^{\text{adv}}_{y,k'}, y, k'|\psi, \mathcal{S}) = \lambda$ regardless of y and k' during all fine-tuning updates.

4.2 Few-Shot without Forgetting

Original methodology. The Few-Shot without Forgetting (FSwF) [16] learns not only the embedding function f_ϕ but also the attention-based classification weight generator. The role of the weight generator G with parameter θ is to compute the classification weight vector $\boldsymbol{w}_k$ for the novel class k using two types of input: (i) the classification weights for the B base classes (*i.e.* $\boldsymbol{v}_1, \ldots, \boldsymbol{v}_B$) trained from $\mathcal{D}^{\text{train}}$ and (ii) the support of novel class k (*i.e.* $S_k = \{(\boldsymbol{x}, y)|y = k, \forall(\boldsymbol{x}, y) \in \mathcal{S}\}$) in each few-shot task:

$$G_\theta(\mathcal{S}, k, \boldsymbol{v}_1, \ldots, \boldsymbol{v}_B) = \theta^{avg} \odot \boldsymbol{w}_k^{avg} + \theta^{att} \odot \boldsymbol{w}_k^{att}, \tag{11}$$

$$\boldsymbol{w}_k^{avg} = \frac{1}{|S_k|} \sum_{(\boldsymbol{x},y)\in S_k} \frac{f_\phi(\boldsymbol{x})}{\|f_\phi(\boldsymbol{x})\|}, \tag{12}$$

$$\boldsymbol{w}_k^{att} = \frac{1}{|S_k|} \sum_{(\boldsymbol{x},y)\in S_k} \sum_{b=1}^{B} \text{Att}(\theta^{att} \frac{f_\phi(\boldsymbol{x})}{\|f_\phi(\boldsymbol{x})\|}, \boldsymbol{\theta}_b^{key}) \cdot \frac{\boldsymbol{v}_b}{\|\boldsymbol{v}_b\|}, \tag{13}$$

where $\odot$ is the element-wise product and Att() is the attention kernel. The learnable parameters include ϕ and $\theta = \{\theta^{avg}, \theta^{att}, \boldsymbol{\theta}_1^{key}, \ldots, \boldsymbol{\theta}_B^{key}\}$, where θ is trained on meta-training tasks and not updated in the meta-test phase. Finally, the novel class weight vector is $\boldsymbol{w}_k = G_\theta(\mathcal{S}, k, \boldsymbol{v}_1, \ldots, \boldsymbol{v}_B)$.

The score function of FSwF for the task becomes

$$h\left(f_\phi(\boldsymbol{x}), k|\psi, \mathcal{S}\right) \coloneqq \frac{\boldsymbol{w}_k^\top f_\phi(\boldsymbol{x})}{\|\boldsymbol{w}_k\|\|f_\phi(\boldsymbol{x})\|}. \tag{14}$$

Boundary-adversarial fine-tuning. By applying the definition of h from Equation (14) to Equations (4) and (6), $\boldsymbol{z}_{y,k'}^{\text{adv}}$ for FSwF is defined by

$$\begin{aligned}\boldsymbol{z}_{y,k'}^{\text{adv}} &= \boldsymbol{z} - \delta \cdot \nabla_{\boldsymbol{z}} \left(\left(\frac{\boldsymbol{w}_y}{\|\boldsymbol{w}_y\|} - \frac{\boldsymbol{w}_{k'}}{\|\boldsymbol{w}_{k'}\|} \right)^\top \frac{\boldsymbol{z}}{\|\boldsymbol{z}\|} \right) \\ &= \boldsymbol{z} - \delta \cdot \left(\frac{I_d}{\|\boldsymbol{z}\|} - \frac{\boldsymbol{z}\boldsymbol{z}^\top}{\|\boldsymbol{z}\|^3} \right) \left(\frac{\boldsymbol{w}_y}{\|\boldsymbol{w}_y\|} - \frac{\boldsymbol{w}_{k'}}{\|\boldsymbol{w}_{k'}\|} \right)\end{aligned} \tag{15}$$

where $\boldsymbol{z} \in \mathbb{R}^d$, and the adversarial loss $\mathcal{L}^{\text{adv}}$ becomes

$$\begin{aligned}\mathcal{L}^{\text{adv}}(\mathcal{S}, \phi) = \frac{1}{|\mathcal{S}|} \sum_{(\boldsymbol{x},y)\in\mathcal{S}} \Bigg[&\frac{1}{K-1} \sum_{k'\neq y} \\ &\left\{ \alpha_{\boldsymbol{x},y} - \min_{k\neq y} \left(\frac{\boldsymbol{w}_y}{\|\boldsymbol{w}_y\|} - \frac{\boldsymbol{w}_k}{\|\boldsymbol{w}_k\|} \right)^\top \frac{\boldsymbol{z}_{y,k'}^{\text{adv}}}{\|\boldsymbol{z}_{y,k'}^{\text{adv}}\|} \right\}_+ \Bigg].\end{aligned} \tag{16}$$

Similarly to the derivation for MetaOptNet in Section 4.1, the adversarial gradient in Equation (15) is derived while fixing the classification weights of $\boldsymbol{w}_y$ and $\boldsymbol{w}_{k'}$ with respect to $\boldsymbol{z}$.

4.3 Standard Transfer Learning

Original methodology. STL [6, 42] is a standard transfer learning approach to the few-shot classification problem. It learns the embedding function f_ϕ during meta-training, and obtains the linear classification weight matrix $\mathbf{W} = [\boldsymbol{w}_1; \ldots; \boldsymbol{w}_K] \in \mathbb{R}^{d\times K}$ per meta-test task using f_ϕ for $\boldsymbol{z} \in \mathbb{R}^d$:

$$[\boldsymbol{w}_1; \ldots; \boldsymbol{w}_K] = \operatorname*{argmin}_{[\boldsymbol{w}'_1;\ldots;\boldsymbol{w}'_K]} \frac{1}{|\mathcal{S}|} \sum_{(\boldsymbol{x},y)\in\mathcal{S}} -\log \left(\frac{\exp(\boldsymbol{w}_y'^\top f_\phi(\boldsymbol{x}))}{\sum_k \exp(\boldsymbol{w}_k'^\top f_\phi(\boldsymbol{x}))} \right) \tag{17}$$

for the support $\mathcal{S}$. The STL computes $[\boldsymbol{w}_1; \ldots; \boldsymbol{w}_K]$ using the gradient descent on Equation (17), and the procedure is not differentiable.

Finally, the score function h of STL for the given task becomes

$$h\left(f_\phi(\boldsymbol{x}), k|\psi, \mathcal{S}\right) \coloneqq \boldsymbol{w}_k^\top f_\phi(\boldsymbol{x}). \tag{18}$$

Boundary-adversarial fine-tuning. By the definition of h from Equation (18), the boundary-adversarial sample generation for STL is derived as

$$\boldsymbol{z}_{y,k'}^{\text{adv}} = \boldsymbol{z} - \delta \cdot \nabla_{\boldsymbol{z}} \left((\boldsymbol{w}_y - \boldsymbol{w}_{k'})^\top \boldsymbol{z} \right) = \boldsymbol{z} - \delta \cdot (\boldsymbol{w}_y - \boldsymbol{w}_{k'}), \tag{19}$$

and $\mathcal{L}^{\text{adv}}$ is

$$\mathcal{L}^{\text{adv}}(\mathcal{S}, \phi) = \frac{1}{|\mathcal{S}|} \sum_{(\boldsymbol{x},y)\in\mathcal{S}} \left[\frac{1}{K-1} \sum_{k'\neq y} \left\{ \alpha_{\boldsymbol{x},y} - \min_{k\neq y} \left((\boldsymbol{w}_y - \boldsymbol{w}_k)^\top \boldsymbol{z}_{y,k'}^{\text{adv}} \right) \right\}_+ \right].$$

5 Experiments

We conduct experiments to evaluate the few-shot classification performance of our MABAS approach. We first present experimental setup (Section 5.1) and discuss the quantitative and qualitative results (Sections 5.2 and 5.3). Please refer to Appendix for additional experimental results including evaluation on tieredImageNet [37].

5.1 Experimental Setup

Datasets. We use three benchmark datasets for evaluation of few-shot classification. (1) miniImageNet [45] consists of 100 classes each of which has 600 images with a size of $84 \times 84 \times 3$. We adopt the same class split used by [36, 22]: 64, 16 and 20 classes for training, validation and the test, respectively. (2) CIFAR-FS [3] splits all of the classes in CIFAR100 [20] into 64 training, 16 validation and 20 test sets, respectively. (3) FC100 [34] is another CIFAR100-based dataset. Classes are split into 60, 20 and 20 for training, validation and test, respectively. This class split is designed to minimize the overlap of information between all three subsets, to be more challenging than CIFAR-FS for few-shot learning.

Embedding functions. We employ ResNet-12, which is one of the popular choices for few-shot learning research [29, 34, 22]. For the experiments of MetaOptNet [22] and STL [6, 42], we use the same architecture of ResNet-12 as [22]. The only architectural difference for STL from MetaOptNet is that an average pooling is applied to the last residual block. For FSwF experiments, we use the architecture in [29] following [16].

Meta-training and meta-validation phase. For the MetaOptNet, FSwF and STL models, we follow the training and validation protocol in the original papers [22, 16, 6] with some minor modifications as follows. For MetaOptNet-Proto, we use a learning decay rate of 0.1 with a decay period of 15 epochs for simplicity. For the STL models, we train for 100 epochs with a batch size of 256 and a learning rate of 0.001 using the cosine annealing decay [18].

Meta-test phase. We test all the models on the 5-way 5-shot and 5-way 1-shot classification tasks. For a fair comparison, all of the meta-test results are obtained using the same setup with the previous works [16, 42, 12]. Each meta-test run consists of 600 tasks sampled from $\mathcal{D}^{\text{test}}$, and a single task contains 15 query samples per each of the 5 classes.

Boundary-adversarial fine-tuning. For all the four base methods, we fine-tune only the last (*i.e.* the fourth) block of the ResNet-12 embedding function, since it preserves most of the representation power of the embedding function. We use Adam [19] optimizer for fine-tuning, and maintain a single set of hyper-parameters per base method across all datasets. In all experiments, we update the embedding function for 150 steps and use the step-based learning rate decay with a decay rate of 0.8 and a period of 5. For MetaOptNet(-Proto), we use $\eta = 0.0005$ as its regularization coefficient and use an initial learning rate of 0.000025. We set the adversarial step size δ to fix the change of margin to $\lambda = 1$, as in Section 4.1. Since FSwF uses the cosine similarity, which is a measure

Table 1. Meta-test accuracies (%) of the four base methods before and after applying our MABAS method with the 95% confidence interval. We also report the accuracy gains (%p) by MABAS. We obtain all results using the source codes provided by the original authors to precisely measure the accuracy improvements by our method.

Method	miniImageNet, 5-way		CIFAR-FS, 5-way		FC100, 5-way	
	1-shot	5-shot	1-shot	5-shot	1-shot	5-shot
STL [6, 42]	55.98 ± 0.78	76.19 ± 0.60	64.32 ± 0.92	83.62 ± 0.61	38.83 ± 0.70	54.54 ± 0.72
+ **Ours**	60.19 ± 0.79	79.34 ± 0.57	67.41 ± 0.91	84.29 ± 0.65	40.76 ± 0.68	58.16 ± 0.78
Gain (%p)	4.21	3.15	3.09	0.67	1.93	3.62
FSwF [16]	55.64 ± 0.82	69.94 ± 0.68	69.23 ± 0.90	82.52 ± 0.68	37.91 ± 0.77	49.75 ± 0.72
+ **Ours**	60.45 ± 0.82	78.28 ± 0.61	70.71 ± 0.89	85.25 ± 0.65	40.63 ± 0.75	54.95 ± 0.75
Gain (%p)	4.81	8.34	1.48	2.73	2.72	5.20
MetaOptNet [22]	62.25 ± 0.82	78.55 ± 0.58	72.11 ± 0.96	84.32 ± 0.65	40.15 ± 0.71	54.92 ± 0.75
+ **Ours**	64.21 ± 0.82	81.01 ± 0.57	73.24 ± 0.95	85.65 ± 0.65	41.74 ± 0.73	57.11 ± 0.75
Gain (%p)	1.96	2.46	1.13	1.33	1.59	2.19
MetaOptNet-Proto	61.68 ± 0.85	78.36 ± 0.59	72.46 ± 0.91	84.02 ± 0.67	40.60 ± 0.75	55.04 ± 0.75
+ **Ours**	65.08 ± 0.86	82.70 ± 0.54	73.51 ± 0.92	85.49 ± 0.68	42.31 ± 0.75	57.56 ± 0.78
Gain (%p)	3.40	4.34	1.05	1.47	1.71	2.52

invariant to scaling of inputs, we set $\eta = 0$. Its initial learning rate is 0.0003 and the adversarial step size is $\delta = 10$. In the STL experiments, we perform the fine-tuning with an initial learning rate of 0.0001, $\eta = 0.01$ and $\delta = 10$.

5.2 Quantitative Evaluation

Table 1 reports the test-time accuracies of the four base methods with or without MABAS on the six tasks. We also summarize the accuracy gains by our method. Table 2 compares our method with the state-of-the-art models from the original papers. Here are some important observations to emphasize:

1. Our method achieves the new state-of-the-art performance on four of the six tasks (miniImageNet 5-shot, CIFAR-FS 1-shot and 5-shot, and FC100 5-shot settings) if we exclude the methods with WRN-28-10 [50], which consumes about three times more parameters than ResNet-12 [22].
2. Our method improves the accuracy in every experiment of the four base methods on three datasets in Table 1, where only a single set of fine-tuning hyperparameters is used for all experiments per base method.
3. The largest accuracy improvement brought by our method is 8.34%p, and the average improvement is $\approx 2.80\%p$.
4. These results demonstrate that our approach is effective for solving the labeled data scarcity problem of few-shot learning. Also, our method is universal enough to improve various base methods with different properties.

Effects of embedding function architectures. FSwF [16] is tested with three ConvNet and one ResNet-12 embedding functions in the original paper. In its experiments on 5-way miniImageNet tasks, the accuracies on $\mathcal{D}^{\text{train}}$ are the highest with ResNet-12, while its accuracies on $\mathcal{D}^{\text{test}}$ are worse than those of the ConvNet architectures. Inspired by this observation, we experiment the fine-tuning accuracy of our approach with different embedding functions.

Table 2. Comparison with the state-of-the-art few-shot learning methods. We present the average meta-test accuracy with its 95% confidence interval. ‡ denotes the results from [36, 22, 42], while all the other baseline scores are referred to their original papers. The numbered superscripts denote the architecture of f_ϕ: [1]Conv-4, [2]Conv-4+MetaNet, [3]ResNet-12, [4]WRN-28-10, [5]Conv4+MetaGAN. Note that WRN-28-10 involves three times more parameters than ResNet-12.

Method	miniImageNet, 5-way		CIFAR-FS, 5-way		FC100, 5-way	
	1-shot	5-shot	1-shot	5-shot	1-shot	5-shot
Meta-LSTM[1] [36]	43.44 ± 0.77	60.60 ± 0.71	-	-	-	-
MatchingNets[1‡] [45]	43.56 ± 0.84	55.31 ± 0.73	-	-	-	-
MAML[1‡] [12]	48.70 ± 1.84	63.11 ± 0.92	58.9 ± 1.9	71.5 ± 1.0	38.1 ± 1.7	50.4 ± 1.0
ProtoNets[1‡] [40]	49.42 ± 0.78	68.20 ± 0.66	55.5 ± 0.7	72.0 ± 0.6	35.3 ± 0.6	48.6 ± 0.6
RelationNets[1‡] [43]	50.44 ± 0.82	65.32 ± 0.70	55.0 ± 1.0	69.3 ± 0.8	-	-
R2D2[1] [3]	51.20 ± 0.60	68.80 ± 0.10	65.30 ± 0.20	79.40 ± 0.10	-	-
FSwF[1] [16]	56.20 ± 0.86	73.00 ± 0.64	-	-	-	-
FSwF[3] [16]	55.45 ± 0.89	70.13 ± 0.68	-	-	-	-
Bilevel Program[3] [13]	50.54 ± 0.85	64.53 ± 0.68	-	-	-	-
MetaGAN[5] [52]	52.71 ± 0.64	68.63 ± 0.67	-	-	-	-
SNAIL[3] [29]	55.71 ± 0.99	68.88 ± 0.92	-	-	-	-
AdaResNet[3] [33]	56.88 ± 0.62	71.94 ± 0.57	-	-	-	-
TADAM[3] [34]	58.5 ± 0.3	76.7 ± 0.3	-	-	40.1 ± 0.4	56.1 ± 0.4
ProtoNets[3‡] [40]	59.25 ± 0.64	75.60 ± 0.48	72.2 ± 0.7	83.5 ± 0.5	37.5 ± 0.6	52.5 ± 0.6
MTL[3] [42]	61.2 ± 1.8	75.5 ± 0.8	-	-	**45.1 ± 1.8**	57.6 ± 0.9
TapNet[3] [49]	61.65 ± 0.15	76.36 ± 0.10	-	-	-	-
MetaOptNet[3] [22]	62.64 ± 0.61	78.63 ± 0.46	72.0 ± 0.7	84.2 ± 0.5	41.1 ± 0.6	55.5 ± 0.6
LEO[4] [38]	61.76 ± 0.08	77.59 ± 0.12	-	-	-	-
CC+rot[4] [15]	62.93 ± 0.45	79.87 ± 0.33	**76.09 ± 0.30**	**87.83 ± 0.21**	-	-
$S2M2_R$[4] [26]	64.93 ± 0.18	**83.18 ± 0.11**	74.81 ± 0.19	87.47 ± 0.13	-	-
LGM-Net[2] [23]	**69.13 ± 0.35**	71.18 ± 0.68	-	-	-	-
STL + **Ours**[3]	60.19 ± 0.79	79.34 ± 0.57	67.41 ± 0.91	84.29 ± 0.65	40.76 ± 0.68	**58.16 ± 0.78**
MetaOptNet +**Ours**[3]	64.21 ± 0.82	81.01 ± 0.57	73.24 ± 0.95	**85.65 ± 0.65**	41.74 ± 0.73	57.11 ± 0.75
–Proto + **Ours**[3]	65.08 ± 0.86	**82.70 ± 0.54**	**73.51 ± 0.92**	85.49 ± 0.68	42.31 ± 0.75	57.56 ± 0.78

Table 3. Meta-test accuracy of FSwF with different embedding functions with or without MABAS. All ConvNets share the same hyperparameters.

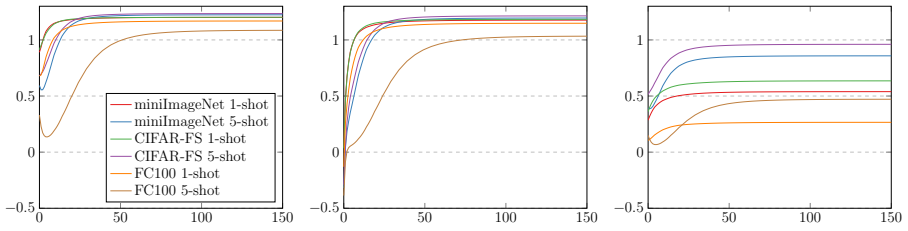
f_ϕ	FSwF	+ MABAS
Conv32	70.01 ± 0.66	70.05 ± 0.65
Conv64	71.89 ± 0.67	72.15 ± 0.66
Conv128	72.59 ± 0.65	72.79 ± 0.63
ResNet-12	69.94 ± 0.68	78.28 ± 0.61

Table 4. Comparison of meta-test accuracy between naive fine-tuning and MABAS using FSwF on the miniImageNet dataset.

Method	miniImageNet, 5-way	
	1-shot	5-shot
FSwF	55.64 ± 0.82	69.94 ± 0.68
+ Naive FT	55.73 ± 0.82	70.14 ± 0.67
+ Ours	60.45 ± 0.82	78.28 ± 0.61

Table 3 shows that our method increases the fine-tuning accuracy of ResNet-12 much larger than ConvNet architectures. The accuracy gains with ConvNets are less than 1%p, while the gain becomes 8.34%p with the ResNet-12 on the 5-way 5-shot miniImageNet task. This result hints that ResNet-12 has high capacity and thus suffers from overfitting, more seriously in few-shot tasks. Our method helps each embedding function to maximize its representation ability even with only a few examples of novel classes.

Comparison with naive fine-tuning. To highlight the effectiveness of MABAS, we compare MABAS with the naive fine-tuning of the embedding function, which is fine-tuning using support samples only (*i.e.* no adversarial sample).



(a) Support (b) Adversarial samples (c) Query

Fig. 2. Average classification margin with FSwF + Ours for support (*left*), adversarial (*middle*) and query (*right*) samples on 5-way tasks. The *x-axis* and *y-axis* denote the number of fine-tuning updates and the average classification margin, respectively. The margin values are averaged over all the meta-test tasks.

Table 4 shows that the naive fine-tuning provides only small performance gains whereas MABAS brings significant improvements.

Evolution of support, adversarial, and query sample margins. Figure 2 shows the evolution of average classification margin for support, adversarial and query samples with FSwF on 5-way meta-test tasks. In each fine-tuning update step, MABAS generates adversarial samples and update the embedding space by increasing the classification margins for those samples. As the fine-tuning progresses, the embedding space becomes denser around the support samples and the margins for novel query samples increase as in (c), due to the changes in the embedding space. It indicates that their classification confidences increase too, which results in accuracy improvement.

5.3 Qualitative Evaluation

Figure 3 illustrates how the embeddings of the support and query change according to fine-tuning updates. Using t-SNE [24], we visualize the embeddings computed by MetaOptNet-Proto and FSwF for the 5-way 5-shot and 1-shot miniImageNet problems. As FSwF uses the cosine similarity in its score function, its ℓ_2-normalized embeddings are taken as input to t-SNE. Before applying our method, the embeddings from different classes are distributed in a mixed way with no clear class separation. As the boundary-adversarial fine-tuning proceeds, not only the support embeddings but also most query embeddings are condensed, and samples from distinct classes become distant.

6 Conclusion

We presented MABAS, a novel model-agnostic approach to generating adversarial samples in the embedding space at test time for few-shot generalization.

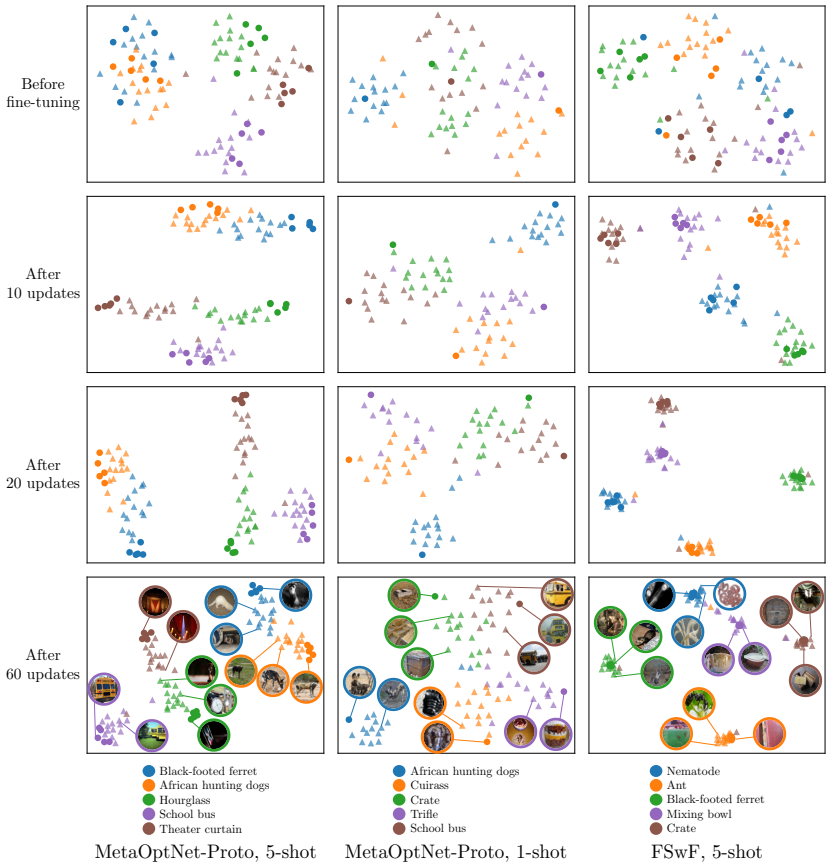


Fig. 3. t-SNE [24] visualization of the support embeddings (*circles*) and query embeddings (*triangles*) obtained by MetaOptNet-Proto and FSwF before and after fine-tuning updates at meta-test time for 5-way miniImageNet tasks.

MABAS is a practical method that works with no additional training and is integrable with any few-shot learning methods. Our results on three few-shot benchmark datasets – miniImageNet, CIFAR-FS, and FC100 – showed that MABAS significantly enhanced the performance of the base methods with various characteristics, and consequently achieved the state-of-the-art performance in several tasks. We believe this work provides a low-effort add-on method for performance enhancement with existing and future few-shot learning methods.

Acknowledgements. This work was supported by Samsung Research Funding Center of Samsung Electronics (No. SRFC-IT1502-51) and Institute of Information & communications Technology Planning & Evaluation (IITP) grant funded by the Korea government (MSIT) (No. 2017-0-01772, Video Turing Test). Jaekyeom Kim was supported by Hyundai Motor Chung Mong-Koo Foundation. Gunhee Kim is the corresponding author.

References

1. Antoniou, A., Storkey, A., Edwards, H.: Data augmentation generative adversarial networks. arXiv preprint arXiv:1711.04340 (2017)
2. Azadi, S., Fisher, M., Kim, V.G., Wang, Z., Shechtman, E., Darrell, T.: Multi-content gan for few-shot font style transfer. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 7564–7573 (2018)
3. Bertinetto, L., Henriques, J.F., Torr, P.H., Vedaldi, A.: Meta-learning with differentiable closed-form solvers. In: Proceedings of the 7th International Conference on Learning Representations (ICLR) (2019)
4. Brendel, W., Rauber, J., Bethge, M.: Decision-based adversarial attacks: Reliable attacks against black-box machine learning models. In: Proceedings of the 6th International Conference on Learning Representations (ICLR) (2018)
5. Carlini, N., Wagner, D.: Towards evaluating the robustness of neural networks. In: 2017 IEEE Symposium on Security and Privacy (SP). pp. 39–57. IEEE (2017)
6. Chen, W.Y., Liu, Y.C., Kira, Z., Wang, Y.C.F., Huang, J.B.: A closer look at few-shot classification. In: Proceedings of the 7th International Conference on Learning Representations (ICLR) (2019)
7. Chen, Z., Fu, Y., Wang, Y.X., Ma, L., Liu, W., Hebert, M.: Image deformation meta-networks for one-shot learning. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 8680–8689 (2019)
8. Choe, J., Park, S., Kim, K., Hyun Park, J., Kim, D., Shim, H.: Face generation for low-shot learning using generative adversarial networks. In: Proceedings of the IEEE International Conference on Computer Vision. pp. 1940–1948 (2017)
9. Crammer, K., Singer, Y.: On the algorithmic implementation of multiclass kernel-based vector machines. Journal of machine learning research **2**(Dec), 265–292 (2001)
10. Dong, N., Xing, E.P.: Domain adaption in one-shot learning. In: Joint European Conference on Machine Learning and Knowledge Discovery in Databases. pp. 573–588. Springer (2018)
11. Fei-Fei, L., Fergus, R., Perona, P.: One-shot learning of object categories. IEEE transactions on pattern analysis and machine intelligence **28**(4), 594–611 (2006)
12. Finn, C., Abbeel, P., Levine, S.: Model-agnostic meta-learning for fast adaptation of deep networks. In: Proceedings of the 34th International Conference on Machine Learning-Volume 70. pp. 1126–1135. JMLR. org (2017)
13. Franceschi, L., Frasconi, P., Salzo, S., Grazzi, R., Pontil, M.: Bilevel programming for hyperparameter optimization and meta-learning. In: International Conference on Machine Learning. pp. 1563–1572 (2018)
14. Gao, H., Shou, Z., Zareian, A., Zhang, H., Chang, S.F.: Low-shot learning via covariance-preserving adversarial augmentation networks. In: Advances in Neural Information Processing Systems. pp. 975–985 (2018)
15. Gidaris, S., Bursuc, A., Komodakis, N., Perez, P., Cord, M.: Boosting few-shot visual learning with self-supervision. In: The IEEE International Conference on Computer Vision (ICCV) (October 2019)
16. Gidaris, S., Komodakis, N.: Dynamic few-shot visual learning without forgetting. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 4367–4375 (2018)
17. Hariharan, B., Girshick, R.: Low-shot visual recognition by shrinking and hallucinating features. In: Proceedings of the IEEE International Conference on Computer Vision. pp. 3018–3027 (2017)

18. He, T., Zhang, Z., Zhang, H., Zhang, Z., Xie, J., Li, M.: Bag of tricks for image classification with convolutional neural networks. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 558–567 (2019)
19. Kingma, D.P., Ba, J.L.: Adam: A method for stochastic optimization. In: Proceedings of the 3rd International Conference on Learning Representations (ICLR) (2015)
20. Krizhevsky, A., Hinton, G., et al.: Learning multiple layers of features from tiny images. Tech. rep., Citeseer (2009)
21. Kurakin, A., Goodfellow, I., Bengio, S.: Adversarial machine learning at scale. In: Proceedings of the 5th International Conference on Learning Representations (ICLR) (2017)
22. Lee, K., Maji, S., Ravichandran, A., Soatto, S.: Meta-learning with differentiable convex optimization. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 10657–10665 (2019)
23. Li, H., Dong, W., Mei, X., Ma, C., Huang, F., Hu, B.G.: Lgm-net: Learning to generate matching networks for few-shot learning. In: International Conference on Machine Learning. pp. 3825–3834 (2019)
24. Maaten, L.v.d., Hinton, G.: Visualizing data using t-sne. Journal of machine learning research **9**(Nov), 2579–2605 (2008)
25. Madry, A., Makelov, A., Schmidt, L., Tsipras, D., Vladu, A.: Towards deep learning models resistant to adversarial attacks. In: Proceedings of the 6rd International Conference on Learning Representations (ICLR) (2018)
26. Mangla, P., Kumari, N., Sinha, A., Singh, M., Krishnamurthy, B., Balasubramanian, V.N.: Charting the right manifold: Manifold mixup for few-shot learning. In: Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV) (March 2020)
27. Mehrotra, A., Dukkipati, A.: Generative adversarial residual pairwise networks for one shot learning. arXiv preprint arXiv:1703.08033 (2017)
28. Miller, E.G., Matsakis, N.E., Viola, P.A.: Learning from one example through shared densities on transforms. In: Proceedings IEEE Conference on Computer Vision and Pattern Recognition. CVPR 2000 (Cat. No. PR00662). vol. 1, pp. 464–471. IEEE (2000)
29. Mishra, N., Rohaninejad, M., Chen, X., Abbeel, P.: A simple neural attentive meta-learner. In: Proceedings of the 6th International Conference on Learning Representations (ICLR) (2018)
30. Mondal, A.K., Dolz, J., Desrosiers, C.: Few-shot 3d multi-modal medical image segmentation using generative adversarial learning. arXiv preprint arXiv:1810.12241 (2018)
31. Moosavi-Dezfooli, S.M., Fawzi, A., Fawzi, O., Frossard, P.: Universal adversarial perturbations. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 1765–1773 (2017)
32. Motiian, S., Jones, Q., Iranmanesh, S., Doretto, G.: Few-shot adversarial domain adaptation. In: Advances in Neural Information Processing Systems. pp. 6670–6680 (2017)
33. Munkhdalai, T., Yuan, X., Mehri, S., Trischler, A.: Rapid adaptation with conditionally shifted neurons. In: International Conference on Machine Learning. pp. 3661–3670 (2018)
34. Oreshkin, B., López, P.R., Lacoste, A.: Tadam: Task dependent adaptive metric for improved few-shot learning. In: Advances in Neural Information Processing Systems. pp. 721–731 (2018)

35. Pahde, F., Ostapenko, O., Hnichen, P.J., Klein, T., Nabi, M.: Self-paced adversarial training for multimodal few-shot learning. In: 2019 IEEE Winter Conference on Applications of Computer Vision (WACV). pp. 218–226. IEEE (2019)
36. Ravi, S., Larochelle, H.: Optimization as a model for few-shot learning. In: Proceedings of the 5th International Conference on Learning Representations (ICLR) (2017)
37. Ren, M., Triantafillou, E., Ravi, S., Snell, J., Swersky, K., Tenenbaum, J.B., Larochelle, H., Zemel, R.S.: Meta-learning for semi-supervised few-shot classification. In: Proceedings of the 6th International Conference on Learning Representations (ICLR) (2018)
38. Rusu, A.A., Rao, D., Sygnowski, J., Vinyals, O., Pascanu, R., Osindero, S., Hadsell, R.: Meta-learning with latent embedding optimization. In: Proceedings of the 7th International Conference on Learning Representations (ICLR) (2019)
39. Schwartz, E., Karlinsky, L., Shtok, J., Harary, S., Marder, M., Kumar, A., Feris, R., Giryes, R., Bronstein, A.: Delta-encoder: an effective sample synthesis method for few-shot object recognition. In: Advances in Neural Information Processing Systems. pp. 2845–2855 (2018)
40. Snell, J., Swersky, K., Zemel, R.: Prototypical networks for few-shot learning. In: Advances in Neural Information Processing Systems. pp. 4077–4087 (2017)
41. Su, J., Vargas, D.V., Sakurai, K.: One pixel attack for fooling deep neural networks. IEEE Transactions on Evolutionary Computation (2019)
42. Sun, Q., Liu, Y., Chua, T.S., Schiele, B.: Meta-transfer learning for few-shot learning. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 403–412 (2019)
43. Sung, F., Yang, Y., Zhang, L., Xiang, T., Torr, P.H., Hospedales, T.M.: Learning to compare: Relation network for few-shot learning. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 1199–1208 (2018)
44. Szegedy, C., Zaremba, W., Sutskever, I., Bruna, J., Erhan, D., Goodfellow, I., Fergus, R.: Intriguing properties of neural networks. In: Proceedings of the 2nd International Conference on Learning Representations (ICLR) (2014)
45. Vinyals, O., Blundell, C., Lillicrap, T., Wierstra, D., et al.: Matching networks for one shot learning. In: Advances in neural information processing systems. pp. 3630–3638 (2016)
46. Wang, Y.X., Girshick, R., Hebert, M., Hariharan, B.: Low-shot learning from imaginary data. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 7278–7286 (2018)
47. Wu, L., Wang, Y., Yin, H., Wang, M., Shao, L., Lovell, B.C.: Few-shot deep adversarial learning for video-based person re-identification. arXiv preprint arXiv:1903.12395 (2019)
48. Xiao, C., Li, B., Zhu, J.Y., He, W., Liu, M., Song, D.: Generating adversarial examples with adversarial networks. In: 27th International Joint Conference on Artificial Intelligence, IJCAI 2018. pp. 3905–3911. International Joint Conferences on Artificial Intelligence (2018)
49. Yoon, S.W., Seo, J., Moon, J.: Tapnet: Neural network augmented with task-adaptive projection for few-shot learning. In: International Conference on Machine Learning. pp. 7115–7123 (2019)
50. Zagoruyko, S., Komodakis, N.: Wide residual networks. arXiv preprint arXiv:1605.07146 (2016)
51. Zhang, H., Zhang, J., Koniusz, P.: Few-shot learning via saliency-guided hallucination of samples. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 2770–2779 (2019)

52. Zhang, R., Che, T., Ghahramani, Z., Bengio, Y., Song, Y.: Metagan: An adversarial approach to few-shot learning. In: Advances in Neural Information Processing Systems. pp. 2365–2374 (2018)
53. Zou, H., Zhou, Y., Yang, J., Liu, H., Das, H.P., Spanos, C.J.: Consensus adversarial domain adaptation. In: Proceedings of the AAAI Conference on Artificial Intelligence. vol. 33, pp. 5997–6004 (2019)