# Boosting Few-Shot Learning with Adaptive Margin Loss

Aoxue Li[1*] Weiran Huang[2] Xu Lan[3] Jiashi Feng[4] Zhenguo Li[2] Liwei Wang[1]

[1]School of EECS, Peking University, China
[2]Huawei Noah's Ark Lab, China
[3]Queen Mary University of London, UK
[4]National University of Singapore, Singapore

lax@pku.edu.cn, weiran.huang@outlook.com, x.lan@qmul.ac.uk,
elefjia@nus.edu.sg, li.zhenguo@huawei.com, wanglw@cis.pku.edu.cn

## Abstract

*Few-shot learning (FSL) has attracted increasing attention in recent years but remains challenging, due to the intrinsic difficulty in learning to generalize from a few examples. This paper proposes an adaptive margin principle to improve the generalization ability of metric-based meta-learning approaches for few-shot learning problems. Specifically, we first develop a class-relevant additive margin loss, where semantic similarity between each pair of classes is considered to separate samples in the feature embedding space from similar classes. Further, we incorporate the semantic context among all classes in a sampled training task and develop a task-relevant additive margin loss to better distinguish samples from different classes. Our adaptive margin method can be easily extended to a more realistic generalized FSL setting. Extensive experiments demonstrate that the proposed method can boost the performance of current metric-based meta-learning approaches, under both the standard FSL and generalized FSL settings.*

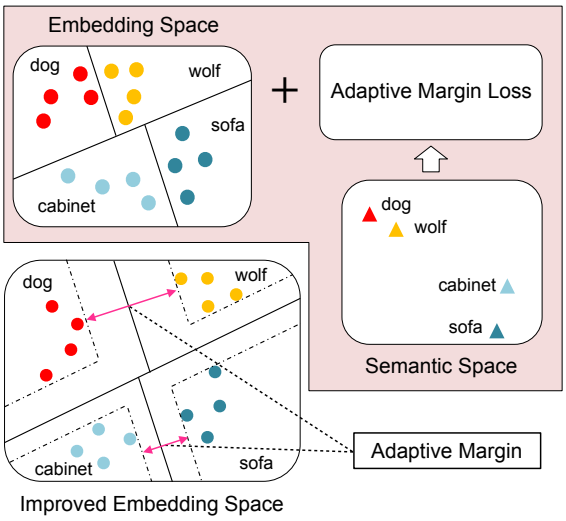


Figure 1. The illustration of the key insight of our adaptive margin loss. In our approach, semantic similarities between different classes (measured in the semantic space of classes) are leveraged to generate adaptive margin between classes. Then, the margin is integrated into the classification loss to make similar classes more separable in the embedding space, which benefits FSL.

## 1. Introduction

Deep learning has achieved great success in various computer vision tasks [10, 26]. However, with a large number of parameters, deep neural networks require large amounts of labeled data for model training. This severely limits their scalability – for many rare classes, it is infeasible to collect a large number of labeled samples. In contrast, humans can recognize an object after seeing it once. Inspired by the few-shot learning ability of humans, there has been an increasing interest in the few-shot learning (FSL) problem [6, 13, 25, 27]. Given a set of base classes with sufficient labeled samples, and a set of novel classes with only a few labeled samples, FSL aims to learn a classifier for the novel classes by learning a generic knowledge from the base classes.

Recently, metric-based meta-learning approaches [8, 14, 16, 27, 29] have shown the superior performance in solving the FSL problem, with attractive simplicity. These approaches usually learn a good embedding space, where samples from the same class are clustered together while samples from different classes are far away from each other. In this way, a new sample from the novel class can be rec-

*This work was done when the first author was an intern at Huawei Noah's Ark Lab.

ognized directly through a simple distance metric within the learned embedding space. The success of these metric-based approaches relies on learning a discriminative embedding space.

To further improve the performance, we introduce the adaptive margin in the embedding space, which helps to separate samples from different classes, especially for similar classes. The key insight of our approach is that the semantic similarity between different classes can be leveraged to generate adaptive margin between classes, *i.e*., the margin between similar classes should be larger than the one between dissimilar classes (as illustrated in Figure 1). By integrating the adaptive margin into the classification loss, our method learns a more discriminative embedding space with better generalization ability.

Specifically, we first propose a class-relevant margin generator which produces an adaptive margin for each pair of classes based on their semantic similarity in the semantic space. By combining the margin generated by class-relevant margin generator and the classification loss of FSL approaches, our class-relevant additive margin loss can effectively pull each class away from other classes. Considering the semantic context among a sampled training task in the FSL, we further develop a task-relevant margin generator. By comparing each class with the rest classes among the task in the semantic space, our task-relevant margin generator produces more suitable margin for each pair of classes. By involving these margin penalty, our task-relevant margin loss learns more discriminative embedding space, thus leads to stronger generalization ability to recognize novel class samples. Moreover, our approach can be easily extended to a more realistic yet more challenging FSL setting (*i.e*., the generalized FSL) where the label space of test data covers both base and novel classes. This is as opposed to the standard FSL setting where the test data contain novel class samples only. Experimental results on the two FSL benchmarks show that our approach significantly improves the performance of current metric-learning-based approaches on both of the two FSL settings.

In summary, our contributions are three folds: (1) To the best of our knowledge, this is the first work to propose an adaptive margin principle to improve the performance of current metric-based meta-learning approaches for FSL. (2) We propose a task-relevant adaptive margin loss to well distinguish samples from different classes in the embedding space according to their semantic similarity, and experimental results demonstrate that our method achieves the state-of-the-art results on the benchmark dataset. (3) Our approach can be easily extended to a more realistic yet more challenging generalized FSL setting, with superior performance obtained. This further validates the effectiveness of our method.

## 2. Related Work

### 2.1. Few-shot Learning

In recent years, few-shot object recognition has become topical. With the success of deep convolutional neural network (DCNN) based approaches in the data-rich setting [5, 10, 26, 30], there has been a great of interest in generalizing such deep learning approaches to the few-shot setting. Most of the recent approaches use a meta-learning strategy. With the meta-learning, these models extract transferable knowledge from a set of auxiliary tasks via episodic training. The knowledge then helps to learn the few-shot classifier trained for the novel classes.

Existing meta-learning based FSL approaches usually learn a model that, given a task (a set of few-shot labeled data and some test query data), produces a classifier that generalizes across all tasks [8]. A main group of gradient-based meta-learning models attempt to modify the classical gradient-based optimization to adapt to a new episodic task by producing efficient parameter updates [1, 7, 8, 23]. Recently, many meta-learning approaches attempt to learn an effective metric on the feature space. The intuition is that if a model can determine the similarity of two images, it can classify an unseen test image with a few labeled examples [27, 29]. To learn an effective metric, these methods make their prediction conditioned on distances to a few labeled examples during the training stage [2, 31]. These examples are sampled from base classes designed to simulate the few-shot scenario. In this paper, we propose a novel generic adaptive margin strategy which can be integrated in existing metric-based meta-learning approaches. Our method can force different classes far from each other in the embedding space. This makes it much easier to recognize novel class samples.

### 2.2. Margin Loss in Visual Recognition

Softmax loss has been widely used in training DCNNs for extracting discriminative visual features for object recognition tasks. By observing that the weights from the last fully connected layer of a classification DCNN trained on the softmax loss bear conceptual similarities with the centers of each class, the works in [4, 18, 33] proposed several margin losses to improve the discriminative power of the trained model. Liu *et al*. [18] introduced the important idea of angular margin. However, their loss function required a series of approximations in order to be computed, which resulted in an unstable training of the network. Wang *et al*. [32] and Wang *et al*. [33] directly add cosine margin to the target logits and achieve better results than [18]. Deng *et al*. [4] proposed an additive angular margin loss to further improve the discriminative power of feature embedding space. Although the aforementioned margin losses have achieved promising results on visual recog-

nition tasks, they are not designed for FSL, where limited samples are provided for novel classes. To learn more suitable margin for FSL, we thus propose an adaptive margin principle, where the semantic context among a sampled training task is considered. By training the FSL approach with our adaptive margin loss, the learned model generalizes better across all tasks and thus achieves better recognition results on novel classes.

# 3. Methodology

## 3.1. Preliminary: Metric-Based Meta-Learning

In the few-shot learning (FSL), we are given a base class set $C_{base}$ consisting of $n_{base}$ base classes, and for each base class, we have sufficient labeled samples. Meanwhile, we also have a novel class set $C_{novel}$ with $n_{novel}$ novel classes, each of which has only a few labeled samples (*e.g.*, less than 5 samples). The **goal** of FSL is to obtain a good classifier for the novel classes.

Meta-learning [7, 27, 31, 34, 35] is a common approach for the FSL. A standard meta-learning procedure involves two stages: meta-training and meta-test. In the meta-training stage, we train the model in an episodic manner. In each episode, a small classification task is constructed by sampling a small training set and a small test set from the whole base class dataset, and then it is used to update the model. In the meta-test stage, the learned model is used to recognize samples from novel classes. Recently, metric-based meta-learning approaches become popular [8, 35]. Most metric-based meta-learning approaches generally assume that there exists an embedding space in which samples cluster around a single representation for each class, and then these class representations are used as references to infer labels of test samples. In the following, we introduce the framework of metric-based meta-learning approaches.

**Meta-Training.** In each episode of meta-training, we sample a $n_t$-way $n_s$-shot classification task from the base class dataset. Specifically, we randomly choose $n_t$ classes from base class set $C_{base}$ for the episodic training, denoted as $C_t$. We randomly select $n_s$ samples from each episodic training class and combine them to form a small training set, which is called support set $S$. Moreover, we also randomly select some other samples from each episodic training class and combine them to form a small test set, which is called query set $Q$.

In the current episode, all samples from both query set and support set are embedded into the embedding space by using an embedding module $\mathcal{F}$. Then, the meta-learner generates class representations $r_1, r_2, \cdots, r_{n_t}$ by using the samples from support set $S$. For example, Prototypical Networks [27] generates class representations by averaging the embeddings of support samples by class. After that, the meta-learner uses a metric module $\mathcal{D}$ (*e.g.*, cosine similarity) to measure the similarity between every query point $(x, y) \in Q$ and the current class representations in the embedding space. Based on these similarities, the meta-learner incurs a classification loss for each point in the current query set. The meta-learner then back-propagates the gradient of the total loss of all query samples. The classification loss can be formulated as:

$$\mathcal{L}^{\text{cls}} = -\frac{1}{|Q|} \sum_{(x,y)\in Q} \log \frac{e^{\mathcal{D}(\mathcal{F}(x), r_y)}}{\sum\limits_{k\in C_t} e^{\mathcal{D}(\mathcal{F}(x), r_k)}}, \tag{1}$$

where $\mathcal{D}(\mathcal{F}(x), r_k)$ denotes the similarity between sample $x$ and the $k$-th class representation $r_k$ predicted by the meta-learner.

**Meta-Test.** In an episode of meta-test, a novel classification task is similar to a training base classification task. Specifically, the labeled few-shot sample set and unlabeled test examples are used to form the support set and query set, respectively. Then they are fed into the learned model with predicted classification results of query samples as outputs.

Different metric-based meta-learning approaches differ in the form of the class representation generation module and metric module, our work introduces different margin loss to improve current metric-based meta-learning approaches.

## 3.2. Naive Additive Margin Loss

An intuitive idea to learn a discriminative embedding space is to add a margin between the predicted results of different classes. This helps to increase the inter-class distance in the embedding space and make it easier to recognize test novel samples. To achieve this, we propose a naive additive margin loss (NAML), which can be formulated as:

$$\mathcal{L}^{\text{na}} = -\frac{1}{|Q|} \sum_{(x,y)\in Q} \log p^{\text{na}}(y|x, S), \tag{2}$$

where

$$p^{\text{na}}(y|x, S) = \frac{e^{\mathcal{D}(\mathcal{F}(x), r_y)}}{e^{\mathcal{D}(\mathcal{F}(x), r_y)} + \sum\limits_{k\in C_t\setminus\{y\}} e^{\mathcal{D}(\mathcal{F}(x), r_k)+m}}.$$

The above naive additive margin loss assumes all classes should be equally far away from each other and thus add a fixed margin among all classes. In this way, this loss forces the embedding module $\mathcal{F}$ to extract more separable visual features for samples from different classes, which benefits the FSL. However, the fixed additive margin may lead to mistakes on test samples of similar classes, especially for the FSL where very limited number of labelled samples are provided in the novel classes.

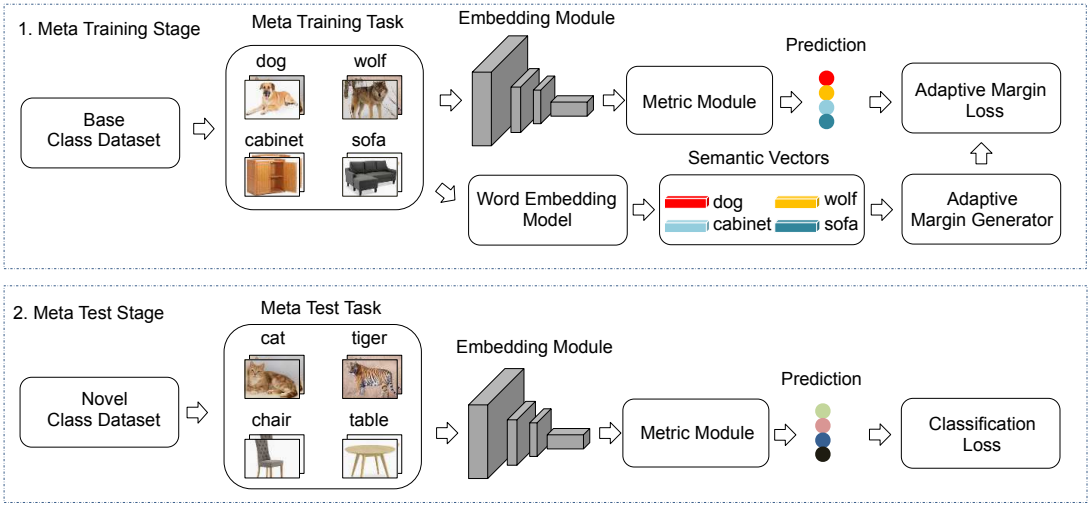


Figure 2. The overview of the proposed approach. Our approach consists of two stages: 1) In each episode of the meta-training stage, we first sample a meta-training task from the base class dataset. Then, the names of classes in the meta-training task are fed into a word embedding model to extract semantic vectors for classes. After that, we propose an adaptive margin generator to produce margin penalty for each pair of classes (*e.g.*, the class relevant margin generator proposed in Section 3.3 or the task relevant margin generator proposed in Section 3.4). Finally, we integrate the margin penalty into the classification loss and thus obtain an adaptive margin loss. A meta-learner consisting of an embedding module and a metric module is trained by minimizing the adaptive margin loss. 2) In the meta-test stage, with the embedding module and metric module learned in the meta-training stage, we use a simple softmax (without any margin) to predict the labels of test novel samples.

### 3.3. Class-Relevant Additive Margin Loss

To better separate similar classes in the feature embedding space, the margin between two classes should be adaptive, *i.e.*, the margin should be larger for similar classes than dissimilar classes. To achieve such adaptive margin in a principled manner, we design a class-relevant additive margin loss (CRAML), where semantic similarities between classes are introduced to adjust the margin.

Before introducing the class-relevant additive margin loss, we first describe how to measure the semantic similarity between classes in a semantic space. Specifically, we represent each class name using a semantic vector extracted by a word embedding model (*e.g.*, Glove [21]). As illustrated in Figure 2, we feed a class name, such as wolf or dog, into the word embedding model, and it will embed the class name into the semantic space and return a semantic word vector. Then, we construct a class-relevant margin generator $\mathcal{M}$. For each pair of classes, class $i$ and class $j$, $\mathcal{M}$ uses their semantic word vectors $e_i$ and $e_j$ as inputs and generates their margin $m^{\text{cr}}_{i,j}$ as follows:

$$m^{\text{cr}}_{i,j} := \mathcal{M}(e_i, e_j) = \alpha \cdot \text{sim}(e_i, e_j) + \beta, \tag{3}$$

where sim denotes a metric (*e.g.*, cosine similarity) to measure the semantic similarity between classes. We use $\alpha$ and $\beta$ to denote the scale and bias parameters for the class-relevant margin generator, respectively.

By introducing the class-relevant margin generator into the classification loss, we obtain a class-relevant additive margin loss as follows.

$$\mathcal{L}^{\text{cr}} = -\frac{1}{|Q|} \sum_{(x,y)\in Q} \log p^{\text{cr}}(y|x, S), \tag{4}$$

where

$$p^{\text{cr}}(y|x, S) = \frac{e^{\mathcal{D}(\mathcal{F}(x), r_y)}}{e^{\mathcal{D}(\mathcal{F}(x), r_y)} + \sum\limits_{k\in C_t\setminus\{y\}} e^{\mathcal{D}(\mathcal{F}(x), r_k)) + m^{\text{cr}}_{y,k}}}.$$

By exploiting the semantic similarity between classes properly, our class-relevant margin loss makes the samples from similar classes to be more separable in the embedding space. The more discriminative embedding space will help better recognize test novel class samples.

### 3.4. Task-Relevant Additive Margin Loss

So far, we assume that the margin is task-irrelevant. A dynamic task-relevant margin generator, which considers the semantic context among all classes in a meta-training task, should generate more suitable margin between different classes. By comparing each class with other classes among a meta-training task, our task-relevant margin generator can measure the relatively semantic similarity between classes. Thus, the generator will add larger margin for relatively similar classes and smaller margin for relatively dissimilar classes. Therefore, we incorporate the generator into

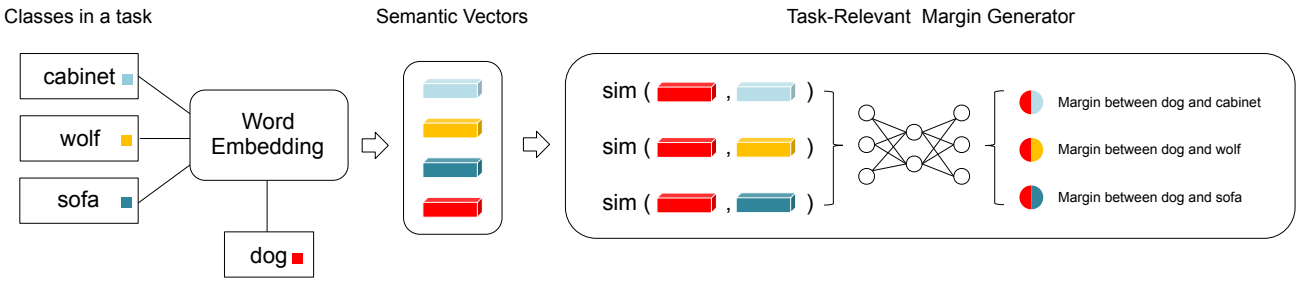


Figure 3. The illustration of the architecture of our task-relevant margin generator.

the classification loss and obtain the task-relevant additive margin loss (TRAML).

Specifically, given a class $y \in C_t$ in a meta-training task, the generator will produce the margins between class $y$ and the other classes $C_t \setminus \{y\}$ in the task according to their semantic similarities, namely,

$$\{m^{\mathrm{tr}}_{y,k}\}_{k \in C_t \setminus \{y\}} = \mathcal{G}\left(\{\mathrm{sim}(e_y, e_k)\}_{k \in C_t \setminus \{y\}}\right), \quad (5)$$

where $m^{\mathrm{tr}}_{y,k}$ denotes the task-relevant margin between class $y$ and class $k$, and $\mathcal{G}$ denotes the task-relevant margin generator, whose architecture is illustrated in Figure 3. As shown in this figure, for a query sample (*e.g.*, a dog image) with label $y \in C_t$, we first compute the similarities between its semantic vector $e_y$ and the semantic vectors of the other classes in the task (*e.g.*, class wolf, sofa and cabinet), respectively. Then, these semantic similarities[1] are fed into the a fully-connected network to generate task-relevant margin for each class pair. By considering the context among all the classes in a meta-training task, our task-relevant margin generator can better measure the similarity among classes, thus generate more suitable margin for each class pair.

By integrating our task-relevant margin generator into the classification loss, we can obtain a task-relevant additive margin loss given in Equation 6 and the outline of computing task-relevant additive margin loss for a training episode is given in Algorithm 1.

$$\mathcal{L}^{\mathrm{tr}} = -\frac{1}{|Q|} \sum_{(x,y) \in Q} \log p^{\mathrm{tr}}(y|x, S), \quad (6)$$

where

$$p^{\mathrm{tr}}(y|x, S) = \frac{e^{\mathcal{D}(\mathcal{F}(x), r_y)}}{e^{\mathcal{D}(\mathcal{F}(x), r_y)} + \sum\limits_{k \in C_t \setminus \{y\}} e^{\mathcal{D}(\mathcal{F}(x), r_k) + m^{\mathrm{tr}}_{y,k}}}.$$

In a test episode, with the learned embedding module and metric module, we use the simple softmax function (without any margin) to predict the label of unlabeled data, *i.e.*, we don't need to use semantic vectors of novel classes during the test stage, which makes our model flexible for any novel class.

[1]The order of input similarities has little impact on the performance.

**Algorithm 1** Task-relevant additive margin loss computation for a training episode in few-shot learning

**Input:** Base class set $C_{base}$, task-relevant generator $\mathcal{G}$.
**Output:** Task-relevant additive margin loss $\mathcal{L}^{\mathrm{tr}}$.

1: Randomly sample $n_t$ classes from base class set $C_{base}$ to form an episodic training class set $C_t$;
2: Randomly sample $n_s$ images per class in $C_t$ to form a support set $S$;
3: Randomly sample $n_q$ images per class in $C_t$ to form a query set $Q$;
4: Obtain the semantic vector for each class in $C_t$ by feeding its class name into a word embedding model;
5: For each query sample, compute the task-relevant margins between its class $y$ and the classes in $C_t \setminus \{y\}$ by using task-relevant margin generator $\mathcal{G}$ according to Equation 5;
6: Compute the task-relevant additive margin loss $\mathcal{L}^{\mathrm{tr}}$ according to Equation 6.

### 3.5. Extension to Generalized Few-Shot Learning

Although the proposed approach is originally designed for the standard FSL, it can be easily extended to the generalized FSL: simply including test data from both base and novel classes, and their labels are predicted from all classes in both base and novel class set in the test stage. This setting is much more challenging and realistic than the standard FSL, where test data are from only novel classes. Note that, our adaptive margin loss is flexible for the generalized FSL: the embedding module and the metric module trained by the adaptive loss with all training samples from base classes can be directly used for label inference of test samples from the disjoint space of both base and novel classes. Experimental results show that our method can improve the state-of-the-art alternative and create a new state-of-the-art for metric-based meta-learning approaches.

## 4. Experiments and Discussions

In this section, we evaluate our approach by conducting three groups of experiments: 1) standard FSL setting where

| Model | Backbone | Type | Test Accuracy | |
|---|---|---|---|---|
| | | | 5-way 1-shot | 5-way 5-shot |
| Matching Networks [31] | 4Conv | Metric | 43.56 ± 0.84 | 55.31 ± 0.73 |
| Prototypical Network [27] | 4Conv | Metric | 49.42 ± 0.78 | 68.20 ± 0.66 |
| Relation Networks [27] | 4Conv | Metric | 50.44 ± 0.82 | 65.32 ± 0.70 |
| GCR [15] | 4Conv | Metric | 53.21 ± 0.40 | 72.34 ± 0.32 |
| Memory Matching Network [3] | 4Conv | Metric | 53.37 ± 0.48 | 66.97 ± 0.35 |
| Dynamic FSL [8] | 4Conv | Metric | 56.20 ± 0.86 | 73.00 ± 0.64 |
| Prototypical Network [27] | ResNet12 | Metric | 56.52 ± 0.45 | 74.28 ± 0.20 |
| TADAM [20] | ResNet12 | Metric | 58.50 ± 0.30 | 76.70 ± 0.38 |
| DC [17] | ResNet12 | Metric | 62.53 ± 0.19 | 78.95 ± 0.13 |
| TapNet [36] | ResNet12 | Metric | 61.65 ± 0.15 | 76.36 ± 0.10 |
| ECMSFMT [24] | ResNet12 | Metric | 59.00 | 77.46 |
| AM3 (Prototypical Network) [35] | ResNet12 | Metric | 65.21±0.49 | 75.20 ± 0.36 |
| MAML [7] | 4Conv | Gradient | 48.70 ± 1.84 | 63.11 ± 0.92 |
| MAML++ [1] | 4Conv | Gradient | 52.15 ± 0.26 | 68.32 ± 0.44 |
| iMAML [22] | 4Conv | Gradient | 49.30 ± 1.88 | - |
| LCC [19] | 4Conv | Gradient | 54.6 ± 0.4 | 71.1 ± 0.4 |
| CAML [11] | ResNet12 | Gradient | 59.23 ± 0.99 | 72.35 ± 0.18 |
| MTL [28] | ResNet12 | Gradient | 61.20 ± 1.80 | 75.50 ± 0.80 |
| MetaOptNet-SVM [12] | ResNet12 | Gradient | 62.64 ± 0.61 | 78.63 ± 0.46 |
| Prototypical Network + TRAML (OURS) | ResNet12 | Metric | 60.31 ± 0.48 | 77.94 ± 0.57 |
| AM3 (Prototypical Network) + TRAML (OURS) | ResNet12 | Metric | **67.10** ± 0.52 | **79.54** ± 0.60 |

Table 1. Comparative results for FSL on the miniImageNet dataset. The averaged accuracy (%) on 600 test episodes is given followed by the 95% confidence intervals (%). Notations: '4Conv' – feature embedding module as in [27], *i.e.*, four stacked convolutions layers of 64 filters; 'ResNet12' – the feature embedding module as in [20], *i.e.*, ResNet12 architecture containing four residual blocks of three stacked $3 \times 3$ convolutional layers; 'Metric' – metric-based meta-learning approaches for FSL; 'Gradient' – gradient-based meta-learning approaches for FSL.

the label space of test data is restricted to a few novel classes at each test episode, 2) generalized FSL setting where the label space of test data is extended to both base classes and novel classes, and 3) further evaluation including ablation study and comparison with other margin losses.

## 4.1. Standard Few-Shot Learning

### 4.1.1 Datasets and Settings

Under the standard FSL setting [27, 31], we evaluate our approach on the most popular benchmark, *i.e.*, miniImageNet. It contains 100 classes randomly selected from ImageNet [26] and each class contains 600 images with resolution of $84 \times 84$. Following the widely used setting in prior works [27, 31], we take 64 classes for training, 16 for validation and 20 for testing. In the training stage, the 64 training classes and 16 validation classes are respectively regarded as base classes and novel classes to decide the model hyperparameters. Following the standard setting adopted by most existing few-shot learning works [3, 8, 27, 29, 31], we conduct 5-way 1-shot/5-shot classification on the miniImageNet dataset. In 1-shot and 5-shot scenarios, each query set has 15 images per class, while each support set contains 1 and 5 image(s) per class, respectively. For a training episode, images in the support sets and query sets are randomly selected from the base class set. In a test episode, images in the support sets and the query sets are randomly selected from the novel class set. The evaluation metric for the miniImageNet dataset is defined as the top-1 classification accuracy on randomly selected 600 test episodes. We test our task-relevant additive margin loss with two backbone metric-based meta learning approaches: Prototypical Networks [27] and its most recent improvement AM3 (Prototypical Networks) [35] which are the state-of-the-art metric-based meta learning methods for FSL.

### 4.1.2 Implementation Details

Our feature embedding module mirrors the ResNet12 architecture used by [20], which consists of four residual blocks. Each block comprises three stacked $3 \times 3$ convolutional layers. Each block is followed by max pooling. We use the same feature extractor on images in both the support set and query set. The fully-connected network in the relation module consists of two fully-connected layers, each followed by a batch normalization layer and a ReLU nonlinearity layer. The word embedding model we used in this paper is Glove [21].

| Model | Novel | | | | | All | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | $n_s$=1 | 2 | 5 | 10 | 20 | $n_s$=1 | 2 | 5 | 10 | 20 |
| Logistic regression (from [34]) | 38.4 | 51.1 | 64.8 | 71.6 | 76.6 | 40.8 | 49.9 | 64.2 | 71.9 | 76.9 |
| Logistic regression w/H (from [9]) | 40.7 | 50.8 | 62.0 | 69.3 | 76.5 | 52.2 | 59.4 | 67.6 | 72.8 | 76.9 |
| Prototypical Network [27] (from [34]) | 39.3 | 54.4 | 66.3 | 71.2 | 73.9 | 49.5 | 61.0 | 69.7 | 72.9 | 74.6 |
| Matching Networks [31] (from [34]) | 43.6 | 54.0 | 66.0 | 72.5 | 76.9 | 54.4 | 61.0 | 69.0 | 73.7 | 76.5 |
| Squared Gradient Magnitude w/H [9] | - | - | - | - | - | 54.3 | 62.1 | 71.3 | 75.8 | 78.1 |
| Batch Squared Gradient Magnitude [9] | - | - | - | - | - | 49.3 | 60.5 | 71.4 | 75.8 | 78.5 |
| Prototype Matching Nets [34] | 43.3 | 55.7 | 68.4 | 74.0 | 77.0 | 55.8 | 63.1 | 71.1 | 75.0 | 77.1 |
| Prototype Matching Nets w/H [34] | 45.8 | 57.8 | 69.0 | 74.3 | 77.4 | 57.6 | 64.7 | 71.9 | 75.2 | 77.5 |
| Dynamic FSL [8] | 46.0 | 57.5 | 69.2 | 74.8 | 78.1 | 58.2 | 65.2 | 72.2 | 76.5 | 78.7 |
| Dynamic FSL + TRAML (OURS) | **48.1** | **59.2** | **70.3** | **76.4** | **79.4** | **59.2** | **66.2** | **73.6** | **77.3** | **80.2** |

Table 2. Comparative results for generalized FSL on the ImageNet2012 dataset. The top-5 accuracies (%) on the novel classes and on all classes are used as the evaluation metrics for this dataset. Methods with "w/ H" use mechanisms that hallucinate extra training examples for the novel classes.

#### 4.1.3 Experimental Results

Table 1 provides comparative results for FSL on the mini-ImageNet dataset. We can observe that: 1) our approach significantly improve the performance of baseline models (*i.e.*, Prototypical Network [27] and AM3 (Prototypical Networks [35]). This indicates that the proposed task-relevant additive margin loss can boost performance of metric-based meta-learning approaches very effectively. 2) Our approach clearly outperforms the state-of-the-art FSL model on both 5-way 1-shot and 5-way 5-shot settings, thanks to the discriminative feature embedding learned by the proposed task-relevant additive margin loss.

### 4.2. Generalized Few-Shot Learning

#### 4.2.1 Dataset and Settings

To further evaluate the effectiveness of our approach, we test our approach in a more challenging yet practical generalized FSL setting, where the label space of test data is extended to both base and novel classes. Following [8, 9, 34], we conduct experiment on the large-scale ImageNet2012 dataset. This benchmark splits the 1000 ImageNet classes into 389 base classes and 611 novel classes; 193 of the base classes and 300 of the novel classes are used for cross validation and the remaining 196 base classes and 311 novel classes are used for the final evaluation (for more details we refer to [9, 34]).

As in [8], the embedding module we used is ResNet10 network that gets as input images of 224 $\times$ 224 resolution. We compare our model with several generalized FSL alternatives: Matching Networks [31], Prototypical Networks [27], Logistic Regression [34], Batch Squared Gradient Magnitude [9], Squared Gradient Magnitude With Hallucination [9], Prototype Matching Nets [34], and Dynamic FSL [8].

We implement our task-relevant additive margin loss on the state-of-the-art model (*i.e.*, Dynamic FSL [8]). Following [34], we first train the embedding module (*i.e.*, ResNet10) by using our task-relevant additive margin loss with all base classes. Then we extract features for all training samples with the learned embedding module and save them to disk. The weight generator in Dynamic FSL [8] will use these pre-computed features as inputs. Finally, we train the weight generator by replacing the original classification loss with our task-relevant additive margin loss. The evaluation metric is the top-5 accuracy on the novel classes and on all classes. We repeat the above experiment 5 times (sampling each time a different set of training images for the novel classes) and report the mean accuracy.

#### 4.2.2 Results

Table 2 provides the comparative results of generalized FSL on the large-scale ImageNet2012 dataset. We can observe that: 1) our approach achieves the best results on all evaluation metrics. This indicates that, with the discriminative embedding space learned by our task-relevant additive margin loss, our approach has the strongest generalization ability under this more challenging setting. 2) Our approach yields consist performance improvement over the state-of-the-art generalized FSL model (*i.e.*, Dynamic FSL [8]) on the 1-shot, 2-shot, 5-shot, 10-shot, and 20-shot settings. This further validates the effectiveness of our approach.

### 4.3. Further Evaluation

#### 4.3.1 Ablation Study on Key Components

We compare our full model with a number of stripped down versions to evaluate the effectiveness of the key components of our approach. Specifically, three of such loss are compared, each of which uses the AM3 (Prototypical Networks) [35] as the baseline model and differs only in which loss is used to train the model: 'Original Classification Loss'

| Model | Test Accuracy | |
|---|---|---|
| | 5-way 1-shot | 5-way 5-shot |
| Original Classification Loss | 65.21 ± 0.49 | 75.20 ± 0.36 |
| Naive Additive Margin Loss | 65.42 ± 0.25 | 75.48 ± 0.34 |
| Class-Relevant Additive Margin Loss | 66.36 ± 0.57 | 77.21 ± 0.48 |
| Our Full Model | **67.10** ± 0.52 | **79.54** ± 0.60 |

Table 3. Ablation study for FSL on the miniImageNet dataset under the standard FSL setting. The evaluation metric is the same as in Table 1.

– model training using the softmax loss provided in [35]; 'Naive Additive Margin Loss' – model training by the loss proposed in Section 3.2; 'Class-Relevant Additive Margin Loss' – model training by the loss proposed in Section 3.3.

Table 3 presents the comparative results of the above losses on the miniImageNet dataset under the standard FSL setting. It can be observed that: 1) Training metric-based meta-learning approaches with our adaptive margin loss leads to significant improvements (see Our Full Model vs. Original Classification Loss). This provides strong supports for our main contribution on embedding learning for FSL. 2) The model trained by the proposed naive additive margin loss shows slight performance improvement over the model trained by original classification loss. This means that simply adding a fixed margin into the classification loss has limited effectiveness in FSL. 3) Thanks to the adaptive margin produced by the class-relevant margin generator, our class-relevant margin additive loss is shown to benefit the embedding learning for FSL (see Class-Relevant Additive Margin Loss vs. Naive Additive Margin Loss). 4) By considering the semantic context among classes in a meta-training task, our task-relevant additive margin loss yields better results than the class-relevant margin loss. Moreover, we observe that the learned coefficient $\alpha$ in Eq. (3) is positive, which verifies our intuition that the margin between similar classes should be larger than the one between dissimilar classes.

#### 4.3.2 Comparison with Other Margin Losses

To validate the effectiveness of the proposed adaptive margin loss, we compare our approach with two margin losses which are widely used in face recognition. Each of them uses the AM3 (Prototypical Networks) [35] as the baseline model and differs only in which loss is used to train the model. The two margin losses are: 1) Additive angular margin loss [4], which add an additive angular margin to the angle between the weight vector and feature embeddings. 2) Additive cosine margin loss [33], which directly adds a cosine margin to the target logits. Note that, both of these two methods add margin penalty to the target logits computed by the dot product between feature embeddings and weight vectors. This is different from Prototypical Network and its variants, which use the opposite of the euclidean distances between class representations and feature embedding as the logits. For fair comparison, we replace the opposite of euclidean metric used in AM3 (Prototypical Network) [35] with the cosine distance, and train the AM3 model with our task-relevant margin loss (the model is denoted by 'Our Full Model (cosine)' in Table 4) .

| Model | Test Accuracy | |
|---|---|---|
| | 5-way 1-shot | 5-way 5-shot |
| Additive angular margin loss [4] | 66.21±0.46 | 77.30 ±0.71 |
| Additive cosine margin loss [33] | 65.96 ±0.56 | 76.93 ±0.49 |
| Our Full Model (cosine) | 66.92 ± 0.43 | 79.08 ± 0.52 |
| Our Full Model (euclidean) | **67.10** ± 0.52 | **79.54** ± 0.60 |

Table 4. Comparative classification accuracies (%) of two other margin losses on the miniImageNet dataset under the standard FSL setting. Notations: 'Our Full Model (cosine)' – implementing our task-relevant additive margin loss on AM3 (Prototypical Network) [35] with cosine distance as metric in the embedding space; 'Our Full Model (euclidean)' – implementing our task-relevant additive margin loss on AM3 (Prototypical Network) [35] with euclidean distance as metric in the embedding space.

Table 4 presents the comparative results of the two margin losses and our losses on the miniImageNet dataset under the standard FSL setting. We can observe that our method is shown to be more effective than the two competitors. It can be expected that, our method is designed for the FSL problem. That is, our method involves semantic similarity among classes in meta-training task to learn a more suitable margin penalty, compared with a fixed one generated by [4, 33]. The suitable margin of each pair of classes helps to learn more discriminative embedding space and thus better distinguish samples from different novel classes.

## 5. Conclusion

In this paper, we propose an adaptive margin principle, which can effectively enhance the discriminative power of embedding space for few-shot image recognition. We first develop a class-relevant additive margin loss which combines the standard classification loss with an adaptive margin generator based semantic similarity between classes. Then, by considering the semantic context among classes in a meta-training task, a task-relevant additive margin loss is further proposed to learn more discriminiative embbeding space for FSL. Furthermore, we also extend the proposed model to the more realistic generalized FSL setting. Experimental results demonstrate that our method is effective under both of the two FSL settings.

**Acknowledgment.** This work is supported by National Key R&D Program of China (2018YFB1402600), BJNSF (L172037) and Beijing Acedemy of Artificial Intelligence.

## References


[1] Antreas Antoniou, Harrison Edwards, and Amos Storkey. How to train your maml. In *ICLR*, 2018.

[2] Luca Bertinetto, Joao F Henriques, Philip HS Torr, and Andrea Vedaldi. Meta-learning with differentiable closed-form solvers. In *ICLR*, 2019.

[3] Qi Cai, Yingwei Pan, Ting Yao, Chenggang Yan, and Tao Mei. Memory matching networks for one-shot image recognition. In *CVPR*, pages 4080–4088, 2018.

[4] Jiankang Deng, Jia Guo, Niannan Xue, and Stefanos Zafeiriou. Arcface: Additive angular margin loss for deep face recognition. In *CVPR*, pages 4690–4699, 2019.

[5] Jeff Donahue, Yangqing Jia, Oriol Vinyals, Judy Hoffman, Ning Zhang, Eric Tzeng, and Trevor Darrell. DeCAF: A deep convolutional activation feature for generic visual recognition. In *Proc. International Conference on Machine Learning*, pages 647–655, 2014.

[6] Matthijs Douze, Arthur Szlam, Bharath Hariharan, and Hervé Jégou. Low-shot learning with large-scale diffusion. In *CVPR*, pages 7229–7238, 2018.

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

[8] Spyros Gidaris and Nikos Komodakis. Dynamic few-shot visual learning without forgetting. In *CVPR*, pages 4367–4375, 2018.

[9] Bharath Hariharan and Ross B. Girshick. Low-shot visual recognition by shrinking and hallucinating features. In *ICCV*, pages 3037–3046, 2017.

[10] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, pages 770–778, 2016.

[11] Xiang Jiang, Mohammad Havaei, Farshid Varno, Gabriel Chartrand, Nicolas Chapados, and Stan Matwin. Learning to learn with conditional class dependencies. In *ICLR*, 2019.

[12] Kwonjoon Lee, Subhransu Maji, Avinash Ravichandran, and Stefano Soatto. Meta-learning with differentiable convex optimization. In *CVPR*, pages 10657–10665, 2019.

[13] Aoxue Li, Tiange Luo, Zhiwu Lu, Tao Xiang, and Liwei Wang. Large-scale few-shot learning: Knowledge transfer-with class hierarchy. In *CVPR*, pages 7212–7220, 2019.

[14] Aoxue Li, Tiange Luo, Tao Xiang, Weiran Huang, and Liwei Wang. Few-shot learning with global class representations. In *ICCV*, pages 9715–9724, 2019.

[15] Aoxue Li, Tiange Luo, Tao Xiang, Weiran Huang, and Liwei Wang. Few-shot learning with global class representations. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 9715–9724, 2019.

[16] Zhenguo Li, Fengwei Zhou, Fei Chen, and Hang Li. Meta-sgd: Learning to learn quickly for few shot learning. *arXiv preprint arXiv:1707.09833*, 2017.

[17] Yann Lifchitz, Yannis Avrithis, Sylvaine Picard, and Andrei Bursuc. Dense classification and implanting for few-shot learning. In *CVPR*, pages 9258–9267, 2019.

[18] Weiyang Liu, Yandong Wen, Zhiding Yu, Ming Li, Bhiksha Raj, and Le Song. Sphereface: Deep hypersphere embedding for face recognition. In *CVPR*, pages 6738–6746, 2017.

[19] Yaoyao Liu, Qianru Sun, An-An Liu, Yuting Su, Bernt Schiele, and Tat-Seng Chua. Lcc: Learning to customize and combine neural networks for few-shot learning. *arXiv preprint arXiv:1904.08479*, 2019.

[20] Boris N. Oreshkin, Pau Rodríguez López, and Alexandre Lacoste. Tadam: Task dependent adaptive metric for improved few-shot learning. In *NeurIPS*, pages 719–729, 2018.

[21] Jeffrey Pennington, Richard Socher, and Christopher Manning. Glove: Global vectors for word representation. In *EMNLP*, 2014.

[22] Aravind Rajeswaran, Chelsea Finn, Sham M. Kakade, and Sergey Levine. Meta-learning with implicit gradients. In *NeurIPS*, 2019.

[23] Sachin Ravi and Hugo Larochelle:. Optimization as a model for few-shot learning. In *ICLR*, 2016.

[24] Avinash Ravichandran, Rahul Bhotika, and Stefano Soatto. Few-shot learning with embedded class models and shot-free meta training. In *ICCV*, pages 331–339, 2019.

[25] Mengye Ren, Sachin Ravi Eleni Triantafillou, Jake Snell, Kevin Swersky, Joshua B. Tenenbaum, Hugo Larochelle, and Richard S. Zemel. Meta-learning for semi-supervised few-shot classification. In *ICLR*, 2018.

[26] 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. *International Journal of Computer Vision*, 115(3):211–252, 2015.

[27] Swersky Kevin Snell, Jake and Richard S. Zemel. Prototypical networks for few-shot learning. In *NeurIPS*, pages 4080–4090, 2017.

[28] Qianru Sun, Yaoyao Liu, Tat-Seng Chua, and Bernt Schiele. Meta-transfer learning for few-shot learning. In *CVPR*, pages 403–412, 2019.

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

[30] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. Going deeper with convolutions. In *CVPR*, pages 1–9, 2015.

[31] Oriol Vinyals, Charles Blundell, Tim Lillicrap, Koray Kavukcuoglu, and Daan Wierstra. Matching networks for one shot learning. In *NeurIPS*, pages 3630–3638, 2016.

[32] Feng Wang, Jian Cheng, Weiyang Liu, and Haijun Liu. Additive margin softmax for face verification. In *ICLR Workshop*, 2018.

[33] Hao Wang, Yitong Wang, Zheng Zhou, Xing Ji, Dihong Gong, Jingchao Zhou, Zhifeng Li, and Wei Liu. Cosface: Large margin cosine loss for deep face recognition. In *CVPR*, pages 5265–5274, 2018.

[34] Yu-Xiong Wang, Ross Girshick1, Martial Hebert, and Bharath Hariharan. Low-shot learning from imaginary data. In *CVPR*, pages 7229–7238, 2018.

[35] Chen Xing, Negar Rostamzadeh, Boris N. Oreshkin, and Pedro O. Pinheiro. Adaptive cross-modal few-shot learning. In *NeurIPS*, pages 4367–4375, 2019.

[36] Sung Whan Yoon, Jun Seo, and Jaekyun Moon:. Tapnet: Neural network augmented with task-adaptive projection for few-shot learning. In *ICML*, pages 7115–7123, 2019.