Rethinking the Metric in Few-shot Learning: From an Adaptive Multi-Distance Perspective

Jinxiang Lai
Siqian Yang
jinxiangla@tencent.com
seasonsyang@tencent.com
Tencent Youtu Lab, China

Guannan Jiang
Xi Wang
jianggn@catl.com
wangx30@catl.com
CATL, China

Yuxi Li
Zihui Jia
lyxok1@sjtu.edu.cn
xibeijia@tencent.com
Tencent Youtu Lab, China

Xiaochen Chen
husonchen@tencent.com
Tencent Youtu Lab, China

Jun Liu
Bin-Bin Gao
junsenselee@gmail.com
csgaobb@gmail.com
Tencent Youtu Lab, China

Wei Zhang
zhangwei@catl.com
CATL, China

Yuan Xie†
yxie@cs.ecnu.edu.cn
School of Computer Science and Technology, East China Normal University, China

Chengjie Wang†
jasoncjwang@tencent.com
Tencent Youtu Lab, China

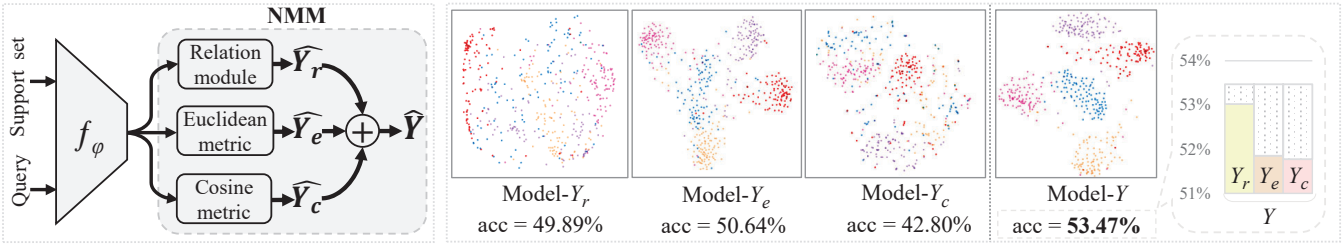


Figure 1: Effectiveness of different metrics and their combination. (a) The Model-Y consists of embedding f_φ (Conv4) and NMM which integrates three metrics by summing their predictions and more details are in Sec.4.1. (b) Comparison between different metrics on 5-way 1-shot classification on *mini*ImageNet. The Model-Y_r, Model-Y_e and Model-Y_c adopt Relation module, Euclidean and Cosine metrics in separated training way, respectively. In right corner, Model-Y achieves 53.47%, while the integrated three metrics obtain different accuracy in joint training way (detail accuracy are shown in the last column of Tab.1).

ABSTRACT

Few-shot learning problem focuses on recognizing unseen classes given a few labeled images. In recent effort, more attention is paid to fine-grained feature embedding, ignoring the relationship among different distance metrics. In this paper, for the first time, we investigate the contributions of different distance metrics, and propose an adaptive fusion scheme, bringing significant improvements in few-shot classification. We start from a naive baseline of confidence summation and demonstrate the necessity of exploiting the complementary property of different distance metrics. By finding the competition problem among them, built upon the baseline, we propose an *Adaptive Metrics Module* (AMM) to decouple metrics fusion into metric-prediction fusion and metric-losses fusion. The former encourages mutual complementary, while the latter alleviates metric competition via multi-task collaborative learning. Based on AMM, we design a few-shot classification framework AMTNet, including the AMM and the *Global Adaptive Loss* (GAL), to jointly optimize the few-shot task and auxiliary self-supervised task, making the embedding features more robust. In the experiment, the proposed AMM achieves 2% higher performance than the naive metrics fusion module, and our AMTNet outperforms the state-of-the-arts on multiple benchmark datasets.

CCS CONCEPTS

• **Computing methodologies** → **Artificial intelligence**; **Computer vision**; **Computer vision problems**;

KEYWORDS

Few-Shot Learning, Distance Metric, Metrics Fusion

ACM Reference Format:
Jinxiang Lai, Siqian Yang, Guannan Jiang, Xi Wang, Yuxi Li, Zihui Jia, Xiaochen Chen, Jun Liu, Bin-Bin Gao, Wei Zhang, Yuan Xie†, and Chengjie Wang†. 2022. Rethinking the Metric in Few-shot Learning: From an Adaptive Multi-Distance Perspective. In *Proceedings of the 30th ACM International*

†Corresponding Author.

MM '22, October 10–14, 2022, Lisboa, Portugal

ACM ISBN 978-1-4503-9203-7/22/10…$15.00
https://doi.org/10.1145/3503161.3547853

Conference on Multimedia (MM '22), October 10–14, 2022, Lisboa, Portugal. ACM, Lisbon, Portugal, 10 pages. https://doi.org/10.1145/3503161.3547853

1 INTRODUCTION

Few-Shot Learning (FSL) task inspires from the association ability of humans, which tries to learn a transferable classifier given a few samples of each class. General FSL methods consist of two parts: feature embedding and distance metrics. Recent works [25, 28] have demonstrated that well-trained embedding is helpful to identify samples of the same category, which benefits following distance measurement. However, the contribution of different distance metrics has not been uniformly studied so far.

The distance metrics in recent investigations can be divided into two categories: non-parametric fixed distance metric [11, 25, 29] (i.e., Euclidean or Cosine metrics), and flexible distance metric [27, 31] with learnable parameters. To the best of our knowledge, there is no uniform investigation to measure the effect of different distance metrics, therefore, we first conduct empirical analysis over different distance metrics. We start with some simple experiments where few-shot learner is trained with three classic distance metrics, including flexible measurement (Relation module [27]) and fixed metrics (Cosine and Euclidean distances). As illustrated in Fig. 1(b), t-SNE visualizations of the embedding features show that Cosine metric (Model-Y_c) and Euclidean metric (Model-Y_e) learn more discriminative embedding (with clear separation gap) under inter-class constraint provided by *fixed metrics*, while Relation module (Model-Y_r) pay more attention to learn a reasonable embedding structure under intra-instance restriction enforced by *flexible metric*.

In order to obtain a flexible metric while keeping discriminative embedding, we design a simple metric fusion solution, termed as *Naive Metrics Module* (NMM), which combines three metrics by directly adding their prediction results. In Fig. 1, the Model-Y adopts the NMM to achieve a comprehensive decision metric with a more reasonable intra-instance structure (the intra-class distribution tends to uniform). Meanwhile, the Model-Y learns a discriminative embedding with the constraints of Cosine and Euclidean metrics. Therefore, the Model-Y takes advantages of both learnable and fixed metrics to offset the weaknesses when each metric is individually applied.

The results in Fig. 1 reveals that the fixed learning metric and the flexible learning metric can be complementary. Furthermore, by comparing results of different combinations between any two metrics (shown in Tab. 1, the detailed analysis is provided in Sec. 4.1), we find there is competition between different distance metrics. For example, at the 7th row in Tab. 1, there are differences among the contributions of the cosine metric with different combinations. The large contrast is (51.77% − 43.67% = 8.1%). This means if we apply NMM, the three metrics cannot maximize their effectiveness, as illustrated in the right corner of Fig. 1(b).

To further explore complementary among different metrics while alleviate the *metrics competition problem*, we proposed the *Adaptive Metrics Module* (AMM) as illustrated in Fig. 2. The AMM inserts adaptive layers to automatically learn the weights of different metrics via considering them as multiple metric-learning tasks, instead of establishing a uniform standard for measurement. Specifically, the AMM decouples metrics fusion into metric-predictions fusion and metric-losses fusion: (i) The metric-predictions fusion utilizes an adaptive layer to re-weight the contributions of different metrics (i.e. $\{\hat{Y}_r, \hat{Y}_e, \hat{Y}_c\}$, which are the metric-predictions of {*Relation module, Euclidean metric, Cosine metric*} respectively); (ii) The metric-losses fusion guides the model to learn a generalized-well embedding via multi-task collaborative learning paradigm (i.e. $\mathcal{G}(\mathcal{L}_r, \mathcal{L}_e, \mathcal{L}_c)$). Besides, a KL regularization term $\mathcal{L}_{KL}$ is added to increase the consistency between predictions of each metric.

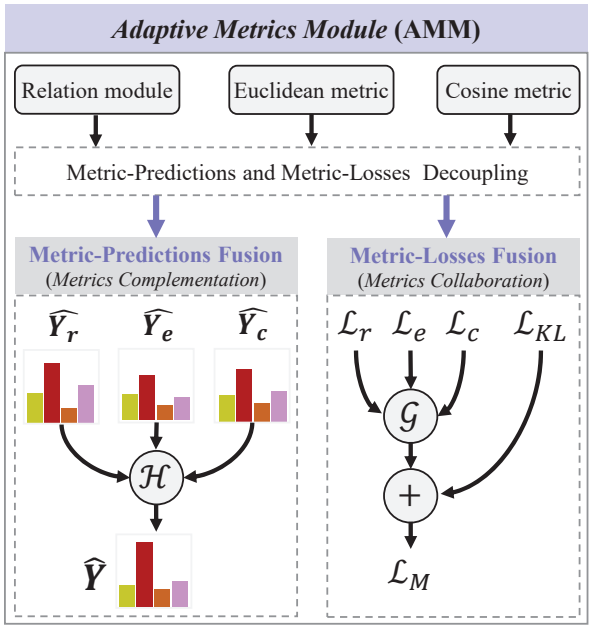


Figure 2: The proposed *Adaptive Metrics Module* (AMM) decouples metrics fusion into metric-predictions fusion and metric-losses fusion to realize metrics complementation and collaboration respectively, where $\mathcal{H}$ and $\mathcal{G}$ are the functions of predictions fusion and losses fusion respectively.

Based on the proposed AMM and inspired by [22, 28], we establish a framework, named *Adaptive Metrics and Tasks Network* (AMTNet), as illustrated in Fig. 3, to integrate auxiliary self-supervised tasks for FSL. To maximize the performance, a *Global Adaptive Loss* (GAL) is designed in the framework, which refs to Pareto Optimal multi-task collaborative learning [37], to merge the embedding from auxiliary tasks and main few-shot classification task. In addition, AMTNet utilizes the GAL to optimize the whole model in an end-to-end manner. To summarize, our main contributions are:

- For the first time, we rethink the role of different types of metric in FSL, and propose to boost the performance from an adaptive multi-distance perspective.
- By finding the complementary and competition among different metrics, a novel *Adaptive Metrics Module* (AMM) is proposed to integrate the flexible and fixed distance metrics to achieve mutual complementarity. Meanwhile, the collaboration between metrics can be ensured by considering them as multi-task learning.
- An effective few-shot classification framework AMTNet is designed based on AMM, which leverages a *Global Adaptive Loss* (GAL) to combine few-shot task with auxiliary self-supervised tasks, realizing an end-to-end training.

• AMTNet achieves the state-of-the-art results on multiple benchmark datasets, and the effectiveness of the proposed AMM and GAL is also demonstrated in the experiments.

2 RELATED WORK

Few-Shot Learning: FSL algorithms pre-train a base classifier with abundant samples, then learn to recognize novel classes with a few labeled samples. There are four representative directions of inductive FSL algorithms as briefly introduced in following.

Optimization-based methods [6, 17, 20] are able to perform rapid adaption with a few training samples for new classes. *Parameter-generating methods* [8, 18] focus on learning a parameter generating network. *Embedding-based methods* [16, 22, 28, 37, 38] aim to learn a generalize-well embedding with supervised or self-supervised learning tasks at first, then freeze this embedding and further train a linear classifier or design a metric classifier on novel classes.

Metric-learning based methods classify a new input image by computing the similarity compared with labeled instances [9]. To learn comparison models, metric-learning based methods make predictions conditioned on distance metrics to few labeled samples during the training stage. There are four popular distance metrics: Cosine similarity [11, 29, 30], Euclidean distance [25], CNN-based relation module [27], and Earth Mover's Distance (EMD) [36]. These methods design carefully on the embedding network to match their corresponding distance metrics. In this paper, we first investigate the relationships of different distance metrics, and prove that an adaptive fusion brings significant improvements in few-shot classification.

Auxiliary Task in FSL: Some recent works gain a performance improvement by training few-shot models with supervised and self-supervised auxiliary tasks. The supervised task for FSL simply performs global classification on the base dataset as in [11]. Recently, the effectiveness of self-supervised learning for FSL has been demonstrated in [3, 5, 7, 16, 22, 26]. In [3, 16], contrastive learning is employed to improve the generalization ability of embedding features. In [7, 26], an additional rotation prediction task was adopted as auxiliary task to learn more robust features.

In contrast to the existing FSL approaches applied supervised or self-supervised auxiliary tasks, we propose to jointly optimize the main few-shot task and auxiliary tasks in an end-to-end manner. Specifically, our approach adopts AMM based metric classification for main few-shot task, global classification [11] for supervised task, and the widely-used and powerful rotation classification for self-supervised task.

3 PRELIMINARY

3.1 Problem Definition

A few-shot classification usually adopts the N-way K-shot episode training strategy, which learns a classifier for N unseen classes with K labeled samples. It involves two mutually disjoint datasets X^{base} and X^{novel}, where the sufficient labeled base set X^{base} contains C^{base} categories, the few labeled novel set X^{novel} has C^{novel} categories, and $C^{base} \cap C^{novel} = \emptyset$. In few-shot testing, a set of episodes $\mathcal{T} = \{\mathcal{T}_i\}_{i=1}^{n_e}$ are sampled from X^{novel}, and the average accuracy over $\mathcal{T}$ are utilized to evaluate the performance of FSL algorithm. An episode $\mathcal{T}_i$ is considered as a N-way K-shot task, which contains N classes with K samples per class as the support set $\mathcal{S} = \left\{\left(x_i^s, y_i^s\right)\right\}_{i=1}^{n_s}$ ($n_s = N \times K$), and a fraction of the rest samples as the query set $\mathcal{Q} = \left\{\left(x_i^q, y_i^q\right)\right\}_{i=1}^{n_q}$ ($n_q = N \times T$). The support subset of the k-th class is denoted as $\mathcal{S}^k$. Following [11, 27, 29, 32], we adopt the episodic training strategy to mimic the few-shot testing setting. In particular, the episodic training iteratively samples the same sized episode from base set X^{base} to train a meta-learner (i.e., a few-shot classification model). After training on X^{base}, given N unseen classes with K labeled samples, the meta-learner aims to classifier $n_q = N \times T$ unlabeled samples into N categories correctly.

3.2 Metric Classifier

Metric classifier categorizes the query images into N novel classes based on similarity measurement. As shown in Fig. 1(a), firstly the embedding f_φ transfers the support set $\mathcal{S}$ and a query sample x^q into prototype feature map $P^k = \frac{1}{|\mathcal{S}^k|}\sum_{x_i^s \in \mathcal{S}^k} f_\varphi(x_i^s)$ and a query feature map $Q = f_\varphi(x^q) \in \mathbb{R}^{c \times h \times w}$, respectively. Then, each pair (P^k, Q) is fed into metric classifier to calculate the similarity for classification. We use $\{d_j\}_{j=r,e,c}$ to denote the corresponding Relation module [27], Euclidean and Cosine metrics, and define $\{\hat{y}_j, \hat{Y}_j, \mathcal{L}_j\}_{j=r,e,c}$ to represent the corresponding metric prediction probability, prediction distribution and loss, respectively. Formally, for metric d_j, the probability that Q belongs to the k-th class is:

$$\hat{y}_j^k(Q) = \hat{y}_j(y = k|Q) = \frac{\exp\left(-d_j\left(Q, P^k\right)\right)}{\sum_{i=1}^{N}\exp\left(-d_j\left(Q, P^i\right)\right)}. \tag{1}$$

The individual metric prediction distribution is expressed as:

$$\hat{Y}_j = [\hat{y}_j^1, \ldots, \hat{y}_j^k, \ldots, \hat{y}_j^N]. \tag{2}$$

According to the true N-way few-shot class label y^q, the individual metric classification loss for d_j is then defined as:

$$\mathcal{L}_j = CE(\hat{y}_j, y^q) = -\sum_{i=1}^{n_q}\log\hat{y}_j(y = y_i^q|Q_i), \tag{3}$$

where CE is the cross-entropy loss function.

3.3 Metrics Fusion Methodology

In this paper, we first propose metric fusion to construct a compound distance metric for FSL via merging the contributions of different distance metrics. The challenge is that, there is no uniform criterion for different metrics. To deal with the problem, we decouple metrics fusion step into two aspects (as shown in Fig.2): predictions fusion and losses fusion. The former obtains a comprehensive metric-based predictor by inserting an adaptive re-weighting layer, and the latter guides the model to learn generalized embedding via multi-task collaborative learning paradigm.

Formally, the predictions fusion is defined as:

$$\hat{Y} = \mathcal{H}(\hat{Y}_r, \hat{Y}_e, \hat{Y}_c), \tag{4}$$

where $\hat{Y}$ is the overall metric prediction distribution, and $\mathcal{H}$ is the predictions fusion function which aims to tackle the problem of metric criterion discordance. In $\hat{Y}$, the corresponding predicted probability for the sample x_q can be represented as $\hat{y}$, and its classification loss $\mathcal{L}_y$ is calculated by Eq. 3 as:

$$\mathcal{L}_y = CE(\hat{y}, y^q). \tag{5}$$

Then, the generic losses fusion is defined as:

$$\mathcal{L}_M = \mathcal{F}\left(\mathcal{G}\left(\mathcal{L}_r, \mathcal{L}_e, \mathcal{L}_c\right), \mathcal{L}_y\right), \tag{6}$$

where, $\mathcal{L}_M$ is the overall metric classification loss, $\mathcal{F}$ and $\mathcal{G}$ are losses fusion functions. By treating the losses fusion as a multi-task collaborative learning paradigm, our method avoids the problem of metric criterion discordance, which is helpful for embedding generalization.

4 METRICS FUSION MODEL

In this section, we first introduce a Naive Metrics Module (NMM) to demonstrate the advantages of metrics fusion. Then, a well-designed metrics fusion approach, named Adaptive Metrics Module (AMM) is proposed based on NMM to further improve the performance.

4.1 Naive Metrics Module

Naive Metrics Module is a naive metric fusion method, which consists of learnable and fixed distance metrics including Relation module, Euclidean and Cosine metrics (illustrated in Fig. 1(a)). Formally, NMM is expressed as:

$$\hat{Y} = \hat{Y}_r + \hat{Y}_e + \hat{Y}_c, \tag{7}$$

$$\mathcal{L}_M = \mathcal{L}_r + \mathcal{L}_e + \mathcal{L}_c. \tag{8}$$

Empirically, the advantage of this simple metric fusion scheme can be reflected in two aspects. First, the metric diversity is enhanced, thus benefit the simiarity measurement. Second, the fusion process naturally exploit the complementarity between learnable and fixed distance metrics.

To evaluate the effectiveness of NMM, we compare the t-SNE of Model-Y, Model-Y_r, Model-Y_e and Model-Y_c, as illustrated in Fig. 1. We observe that: (i) The embedding feature of the Model-Y is still discriminative even in the fixed metrics combination. (ii) The Model-Y achieves a substantial accuracy improvement compared with the independent counterpart, indicating that integrating the learnable and the fixed metrics achieves mutual complementarity.

To demonstrate the observations of t-SNE visualizations, we conducted more experiments about the different combinations of multiple metrics, and the results shown in Tab. 1 indicate that: (i) With the increase of the metric diversity, the Merge Acc keeps going up. (ii) The Merge Acc is better than the corresponding Individual Acc, which indicates that integrating multiple metrics can achieve a more comprehensive decision maker. For example, in column 8 of Tab. 1, the Merge Acc of Model-Y achieves 53.47% which is higher than the corresponding three Individual Acc (53.02%, 51.85% and 51.77% respectively). (iii) The Individual Acc of the integrating model is superior to the corresponding independent model, and the advantages are more obvious along with the increase of the metric diversity. Specifically, comparing column 8 with columns 2, 3 and 4 of Tab. 1, all the Individual Acc of Model-Y are superior than the corresponding independent Model-Y_c, Model-Y_e and Model-Y_r. Their Individual Acc increase from 42.80%, 50.64% and 49.89% to 51.77%, 51.85% and 53.02%, which indicates that the Model-Y obtains a more discriminative embedding than these corresponding independent models.

Though NMM is demonstrated to be effective for FSL, the results in Tab. 1 still reveals the problem of competition among different distance metrics, i.e. the contributions of metrics are different and one metric may bring negative effect to other contributors. In the training stage, different distance metrics have different quantities of their metric losses, which leads to inconsistent gradients in the process of back propagation. Consequently, competition occurs while training with different distance metrics. With the help of the descriptions in section 3.2, the problem can be subdivided into metric criterion discordance in predictions fusion (Eq. 7), and tasks competition in losses fusion (Eq. 8).

Table 1: Comparison between different combinations of multiple metrics on 5-way 1-shot classification on *mini*ImageNet. The embedding backbone f_φ is Conv4. For each column, the model is optimized by $\mathcal{L}_M$ and inferences with different metrics, i.e. each column has the same learned embedding f_φ for different metrics. In Individual Acc, each row has the same metric while with different learned embedding.

Column Index		1	2	3	4	5	6	7	8
Metric num			1	1	1	2	2	2	**3**
$\mathcal{L}_M$	$\mathcal{L}_r$	✓	-	-	-	✓	✓	✓	
	$\mathcal{L}_e$	-	✓	-	✓	-	✓	✓	
	$\mathcal{L}_c$	-	-	✓	✓	✓	-	✓	
Individual Acc	$\hat{Y}_r$	49.89	-	-	-	51.07	50.88	**53.02**	
	$\hat{Y}_e$	-	50.64	-	51.06	-	49.63	**51.85**	
	$\hat{Y}_c$	-	-	42.80	51.01	43.67	-	**51.77**	
Merge Acc	$\hat{Y}$	49.89	50.64	42.80	51.69	51.48	52.02	**53.47**	

4.2 Adaptive Metrics Module

To handle the metrics competition problem, we propose an *Adaptive Metrics Module* (AMM) on basis of NMM. The AMM decouples metrics fusion into metric-predictions fusion and metric-losses fusion to realize metrics complementation and collaboration respectively.

Metric-Predictions Fusion: To deal with the problem of metric criterion discordance in *predictions fusion*, an adaptive layer is inserted in AMM to automatically learn the weights of different metrics:

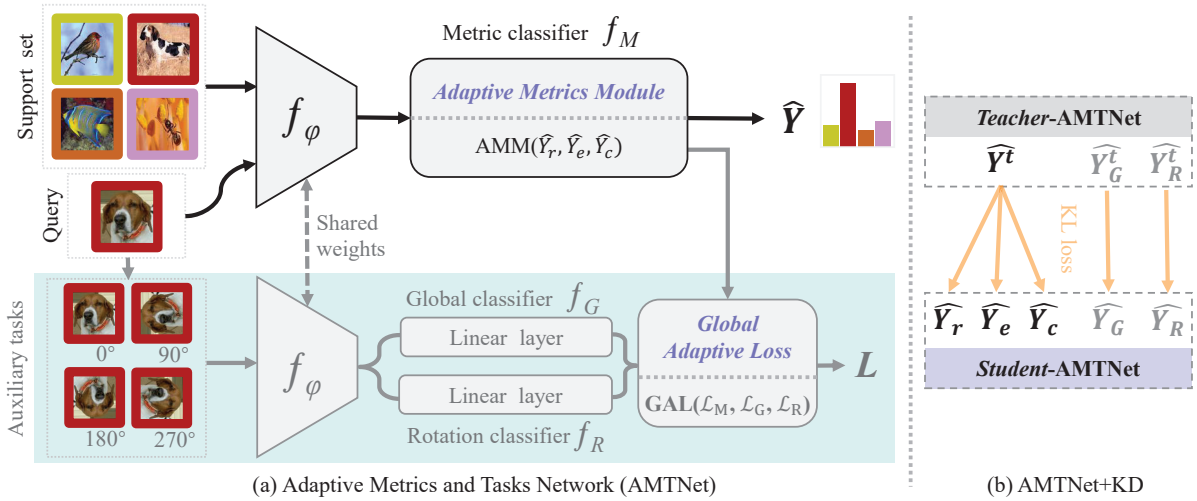
$$\hat{Y} = \sum_{j=r,e,c} (1 + u_j)\hat{Y}_j, \tag{9}$$

where u_j is a learnable variable which is used as a scaling factor reflecting the contribution of $\hat{Y}_j$, and the residual weighted (i.e., $1 + u_j$) strategy is applied to ensure learning stability. Then the metric classification loss $\mathcal{L}_y$ for $\hat{Y}$ can be calculated by Eq. 5.

Metric-Losses Fusion: To tackle the competition problem among different losses, AMM learns the metric loss weights for the corresponding metric losses $\{\mathcal{L}_r, \mathcal{L}_e, \mathcal{L}_c\}$. Inspired by [1], we use task-dependent uncertainty as a basis to modulate multi-task losses:

$$\mathcal{G}\left(\mathcal{L}_r, \mathcal{L}_e, \mathcal{L}_c\right) = -\sum_{i=1}^{n_q} \log(\hat{y}_r \cdot \hat{y}_e \cdot \hat{y}_c) \approx \sum_{j=r,e,c} \left(\frac{1}{\theta_j^2}\mathcal{L}_j + log\theta_j^2\right), \tag{10}$$

where θ_j is a learnable variable for metric d_j, and the detailed formula derivation is presented in the APPENDIX. According to Eq. 10, large scale value θ_j will decrease the contribution of $\mathcal{L}_j$, whereas small scale θ_j will increase its contribution. The loss $\mathcal{L}_M$



(a) Adaptive Metrics and Tasks Network (AMTNet) (b) AMTNet+KD

Figure 3: (a) The framework of the proposed AMTNet. (b) AMTNet+KD applies knowledge distillation.

is penalized when setting θ_j too small, therefore, it can prevent trivial solutions where some loss terms are degraded to zero.

Moreover, a KL regularization term $\mathcal{L}_{KL}$ is added to increase consistencies between different metric distributions. Formally, the overall metric-losses fusion of AMM is expressed as:

$$\begin{aligned}\mathcal{L}_M &= \mathcal{G}(\mathcal{L}_r, \mathcal{L}_e, \mathcal{L}_c) + \alpha \mathcal{L}_{KL} \\ &= \sum_{j=r,e,c} \left(\frac{1}{\theta_j^2}\mathcal{L}_j + log\theta_j^2\right) + \alpha \sum_{j=r,e,c} KL\left(\hat{Y}_j, \|\hat{Y}\|\right),\end{aligned} \tag{11}$$

where α is a hyper-parameter, $KL(\cdot,\cdot)$ is Kullback–Leibler divergence function. In detail, the KL regularization $\mathcal{L}_{KL}$ considers the fused prediction $\hat{Y}$ as the teacher-metric and the individual metric predictions $\hat{Y}_j$ as the student-metric in Eq. 11, which increases consistencies between different metric distributions to alleviate the metric criterion discordance problem.

Algorithm 1: AMTNet model training

Input: X^{base}; training epochs E
Model: Backbone f_φ; Metric classifier (i.e. AMM) f_M; Global classifier f_G; Rotation classifier f_R
Output: f_φ; f_M

1 **begin**
2 Randomly initialize $\{f_\varphi, f_M, f_G, f_R\}$;
3 **for** *i from 1 to E* **do**
4 Sample training data $(\mathcal{S}, \mathcal{Q}) \in X^{base}$;
5 Compute $\mathcal{L}$ by Eq. 13 involved $\{\mathcal{L}_M, \mathcal{L}_G, \mathcal{L}_R\}$;
6 Optimize $\{f_\varphi, f_M, f_G, f_R\}$ with SGD;
7 Freeze all params except $\{u_r, u_e, u_c\}$ in Eq. 9;
8 Compute loss $\mathcal{L}_y$ by Eq. 5;
9 Optimize $\{u_r, u_e, u_c\}$ with SGD;
10 **end**
11 **return** learned f_φ and f_M.
12 **end**

Finally, we optimize $\mathcal{L}_M$ and $\mathcal{L}_y$ separately, of which the algorithm is shown in Alg. 1. Especially, we do not optimize the learnable variables u (Line 7 of Alg. 1) and network weights f (Line 6 of Alg. 1) simultaneously, because when f changes, the competition of different metrics changes as well. Consequently, AMM is able to learn a generalized embedding via optimizing $\mathcal{L}_M$ based on multi-task collaborative learning paradigm, and obtain a comprehensive similarity measurement through optimizing $\mathcal{L}_y$.

5 ADAPTIVE METRICS AND TASKS NETWORK

On the basis of AMM, we design a novel training framework for FSL, named as *Adaptive Metrics and Tasks Network* (AMTNet). The structure of the framework is illustrated in Fig. 3(a), which consists of the main few-shot branch (see the top in Fig. 3(a)) and the auxiliary self-supervision branch (see the bottom in Fig. 3(a)). In training stage, a *Global Adaptive Loss* (GAL) is proposed to coordinate the relationship between different losses generated from the above two branches. It also adopts the methodology of multi-task learning to optimize the model in an end-to-end manner, whose pseudo code is shown as in Algorithm 1. In inductive inference, for a task with novel data, the pre-trained embedding is directly utilized to extract the features of the support classes and query samples. Then the overall prediction $\hat{Y}$ for a query is predicted by the AMM based metric classifier via Eq. 9.

5.1 Model Training via Optimization

As shown in Fig. 3, firstly each query sample x^q is rotated under four angles $[0°, 90°, 180°, 270°]$, thus the query set $\mathcal{Q} = \{\left(x_i^q, y_i^q\right)\}_{i=1}^{n_q}$ is transformed into a rotated query set $\bar{\mathcal{Q}} = \{\left(\bar{x}_i^q, \bar{y}_i^q\right)\}_{i=1}^{n_q \times 4}$. Then, the embedding f_φ transfers the support set $\mathcal{S}$ and a rotated query sample $\bar{x}^q$ into the class prototype feature map $P^k = \frac{1}{|\mathcal{S}^k|}\sum_{x_i^s \in \mathcal{S}^k} f_\varphi(x_i^s)$ and a query feature map $Q = f_\varphi(\bar{x}^q)$, respectively. Finally, AMTNet is optimized by minimizing the overall classification loss $\mathcal{L}$ contributing from the fused Metric $\mathcal{L}_M$ (defined in Eq. 11), the Global classifier ($\mathcal{L}_G$) and a Rotation classifier ($\mathcal{L}_R$), where the latter two are defined below.

The Metric classifier f_M (see Fig. 3(a)) categorizes the query images into N support classes based on the proposed AMM similarity measurement. The Global classifier f_G categorizes the query

samples into all available classes of training set, and its loss is $\mathcal{L}_G = CE(\hat{y}_G, C^q)$, where $\hat{y}_G$ is the prediction results of the Global classifier, and C^q is the true global class of $\bar{x}^q$ with total of C classes. Similarly, the loss of Rotation classifier f_R is computed as $\mathcal{L}_R = CE(\hat{y}_R, B^q)$, where $\hat{y}_R$ is the predictions, and B^q is the true rotation class of $\bar{x}^q$ in four kinds of angle.

Global Adaptive Loss: Then, the generic *Global Adaptive Loss* (GAL) with three inputs $\{\mathcal{L}_M, \mathcal{L}_G, \mathcal{L}_R\}$ is defined as follows:

$$\mathcal{L} = w_M \mathcal{L}_M + w_G \mathcal{L}_G + w_R \mathcal{L}_R, \tag{12}$$

where $\{w_M, w_G, w_R\}$ are used to re-weight the losses of different tasks in optimizing. According to [11], it sets $w_M = \frac{1}{2}$, and recommends that the weight of auxiliary loss should be larger than metric loss (i.e. $\{w_G, w_R\} > w_M$). Further inspired by the multi-task learning defined in Eq. 10, we derive:

$$\mathcal{L} = \frac{1}{2}\mathcal{L}_M + \sum_{z=G,R}\left(w_z \mathcal{L}_z + log\frac{1}{w_z}\right), \ where \quad w_z = \frac{1}{\theta_z^2} + \lambda, \tag{13}$$

where, $\mathcal{L}_M$ is defined in Eq. 11, learnable variables include $\{\theta_r, \theta_e, \theta_c\}$ and $\{\theta_G, \theta_R\}$, and λ is a hyper-parameter (empirical value is within [0.5, 2.0] to balance the effects of losses of few-shot and auxiliary tasks, as illustrated in Tab. 6). In Eq. 13, in order to approximate the condition of $\{w_G, w_R\} > w_M$, we propose to introduce a parameter λ as the base bias for auxiliary loss. The results in Tab. 6 demonstrate that the proposed λ plays an important role in final performance.

5.2 Discussion

The Benefits of GAL: In previous works [7, 26, 37], self-supervised rotation task is treated as an independent rotation prediction task, which only benefits embedding with rotation aware property. The usage in our framework is obviously different as follows: (i) Our approach exploits both rotation aware and rotation invariant properties, which results in a more semantic and robust embedding. Concretely, as illustrated in Fig. 3(a), Rotation classifier is rotation aware, and Global classifier and Metric classifier are rotation invariant. (ii) Different from these two stage embedding-based methods [22, 37], our model is training in an end-to-end one stage manner (i.e. optimizes few-shot task and auxiliary tasks simultaneously).

AMTNet with Knowledge Distillation: Our AMTNet is optimized via the proposed GAL based on multi-task paradigm, which is helpful to prevent overfitting and learn robust embedding. With the usage of GAL, our AMTNet is able to achieve better few-shot classification performance with a larger backbone f_φ such as WRN-28 instead of the ResNet-12. As shown in Fig. 3(b), with knowledge distillation [10], AMTNet+KD uses a strong teacher-AMTNet (WRN-28) model to train a student-AMTNet (ResNet-12). The teacher-AMTNet is first trained via Algorithm 1, then the student-AMTNet is optimized by the overall loss:

$$\begin{aligned}
\mathcal{L} &= \mathcal{L}_{GAL} + \beta \mathcal{L}_{KD}, \quad where, \\
\mathcal{L}_{GAL} &= \frac{1}{2}\sum_{j=r,e,c}\left(\frac{1}{\theta_j^2}\mathcal{L}_j + log\theta_j^2\right) \\
&\quad + \sum_{z=G,R}\left(\left(\frac{1}{\theta_z^2} + \lambda\right)\mathcal{L}_z + log\frac{1}{\left(\frac{1}{\theta_z^2} + \lambda\right)}\right), \\
\mathcal{L}_{KD} &= \sum_{j=r,e,c} KL\left(\hat{Y}_j, \|\hat{Y}^t\|\right) + \sum_{z=G,R} KL\left(\hat{Y}_z, \hat{Y}_z^t\right),
\end{aligned} \tag{14}$$

where β is a hyper-parameter, $\hat{Y}^t$ and $\hat{Y}_z^t$ are the output predictions of teacher-AMTNet.

6 EXPERIMENT

6.1 Experimental Setup

Datasets: ***mini*ImageNet** dataset is a subset of ImageNet [14], which consists of 100 classes with image size of 84×84 pixels. We split the 100 classes following the setting in [32], i.e. 64, 16 and 20 classes for training, validation and testing respectively. ***tiered*ImageNet** dataset [21] is also a subcollection of ImageNet. It contains 608 classes with image size of 84×84 pixels, which are separated into 351 classes for training, 97 for validation and 160 for testing. **CIFAR-FS** dataset is constructed by randomly splitting the 100 classes of the CIFAR-100 dataset into 64, 16, and 20 train, validation, and test splits.

Evaluation: We conduct experiments on 5-way 1-shot and 5-shot inductive settings. We report the *average accuracy* and 95% *confidence interval* over 2000 episodes sampled from the test set. Here, we do not adopt transductive setting [2, 13, 19, 34].

Implementation details: Following [32], horizontal flipping, random cropping, random erasing and color jittering are employed for data augmentation in training. According to the ablation study results, the hyperparameter λ in Eq. 13 is set to 0.5 and 1.5 under ResNet-12 and WRN-28 respectively, and β in Eq. 14 is set to 0.75 for AMTNet+KD. In line with the setting of [11], SGD with 5e-4 weight decay is used as the optimizer. The detailed info of learning-rate and training-epochs are referred to our public source code.

6.2 Comparison with State-of-the-arts

Tab. 2 and Tab. 3 compare our method with existing few-shot methods on *mini*ImageNet, *tiered*ImageNet and CIFAR-FS, which indicates that the proposed AMTNet outperforms the existing SOTAs with a large margin under WRN-28 backbone as well as is very competitive under Conv4 and ResNet-12.

On *mini*ImageNet, our AMTNet performs better than the best parameter-generating method wDAE [8] with an improvement up to 7.09%. Comparing to Pareto Optimal multi-task collaborative learning based PSST [37] approach, our GAL based AMTNet achieves 5.89% higher performance. Many existing metric-based methods [25, 27, 29, 36] focus on designing different distance metrics for few-shot classification, and the strongest competitor is DeepEMD [36] with Earth Mover's Distance. Our AMTNet is 3.57% higher than DeepEMD, which demonstrates the superiority of the proposed adaptive metrics module. Some metric-based methods [11, 32] apply cross attention strategy to get more discriminative features before metric classification. Even without any feature attention, our method still outperforms the DANet [32] with an improvement up to 1.72%.

Besides, COSOC [33] method adopts a complicated multi-stage framework, including pre-training backbone by contrastive learning, data clustering, generating cropped data, training backbone by FSL algorithm and inference with image-cropping. Furthermore, COSOC also takes an extra cost at inference step, because it needs to crop the source image several times for similarity calculation. Comparing to COSOC, our AMTNet is an end-to-end framework, which achieves new state-of-the-art results with WRN-28.

Table 2: Comparison with SOTAs on 5-way classification on *mini*ImageNet and *tiered*ImageNet datasets. * indicates methods evaluated using *multi-cropping*. The best two results under different backbones are highlighted in boldtype and *italics*.

Model	Backbone	Venue	*mini*ImageNet		*tiered*ImageNet	
			1-shot	5-shot	1-shot	5-shot
MatchingNet [29]	Conv4	NeurIPS'2016	43.44 ± 0.77	60.60 ± 0.71	-	-
ProtoNet [25]	Conv4	NeurIPS'2017	49.42 ± 0.78	*68.20 ± 0.66*	53.31 ± 0.89	*72.69 ± 0.74*
RelationNet [27]	Conv4	CVPR'2018	*50.44 ± 0.82*	65.32 ± 0.70	*54.48 ± 0.93*	71.32 ± 0.78
Our AMTNet	Conv4	Ours	**54.91 ± 0.47**	**71.01 ± 0.37**	**57.33 ± 0.50**	**73.11 ± 0.36**
CAN [11]	ResNet-12	NeurIPS'2019	63.85 ± 0.48	79.44 ± 0.34	69.89 ± 0.51	84.23 ± 0.37
DeepEMD [36]	ResNet-12	CVPR'2020	65.91 ± 0.82	82.41 ± 0.56	71.16 ± 0.87	86.03 ± 0.58
IENet [22]	ResNet-12	CVPR'2021	66.82 ± 0.80	*84.35 ± 0.51*	71.87 ± 0.89	86.82 ± 0.58
infoPatch [16]	ResNet-12	AAAI'2021	67.67 ± 0.45	82.44 ± 0.31	71.51 ± 0.52	85.44 ± 0.35
RFS [28]	ResNet-12	ECCV'2020	67.73 ± 0.63	83.35 ± 0.41	72.55 ± 0.69	86.72 ± 0.49
DANet [32]	ResNet-12	CVPR'2021	67.76 ± 0.46	82.71 ± 0.31	71.89 ± 0.52	85.96 ± 0.35
COSOC* [33]	ResNet-12	NeurIPS'2021	*69.28 ± 0.49*	**85.16 ± 0.42**	**73.57 ± 0.43**	**87.57 ± 0.10**
Our AMTNet	ResNet-12	Ours	69.17 ± 0.46	83.88 ± 0.31	72.63 ± 0.49	86.54 ± 0.36
Our AMTNet+KD	ResNet-12	Ours	**69.48 ± 0.47**	84.22 ± 0.31	*73.02 ± 0.50*	*86.98 ± 0.36*
wDAE-GNN [8]	WRN-28	CVPR'2019	61.07 ± 0.15	76.75 ± 0.11	68.18 ± 0.16	83.09 ± 0.12
LEO [23]	WRN-28	ICLR'2019	61.76 ± 0.08	77.59 ± 0.12	66.33 ± 0.05	81.44 ± 0.09
wDAE [8]	WRN-28	CVPR'2019	62.96 ± 0.15	78.85 ± 0.10	68.18 ± 0.16	83.09 ± 0.12
PSST [37]	WRN-28	CVPR'2021	64.16 ± 0.44	80.64 ± 0.32	-	-
FEAT [35]	WRN-28	CVPR'2020	65.10 ± 0.20	81.11 ± 0.14	70.41 ± 0.23	84.38 ± 0.16
CAN [11]	WRN-28	NeurIPS'2019	66.12 ± 0.47	80.43 ± 0.33	71.04 ± 0.53	84.92 ± 0.37
DANet [32]	WRN-28	CVPR'2021	*67.84 ± 0.46*	*82.74 ± 0.31*	*72.18 ± 0.52*	*86.26 ± 0.35*
Our AMTNet	WRN-28	Ours	**70.05 ± 0.46**	**84.55 ± 0.29**	**73.86 ± 0.50**	**87.62 ± 0.33**

Table 3: Comparison on 5-way task on CIFAR-FS dataset.

Model	Backbone	CIFAR-FS	
		1-shot	5-shot
RFS [28]	ResNet-12	71.50 ± 0.80	86.00 ± 0.50
MetaOpt [15]	ResNet-12	72.60 ± 0.70	84.30 ± 0.50
MABAS [12]	ResNet-12	73.51 ± 0.92	85.49 ± 0.68
DSN-MR [24]	ResNet-12	75.60 ± 0.90	86.20 ± 0.60
IENet [22]	ResNet-12	76.83 ± 0.82	89.26 ± 0.58
Our AMTNet	ResNet-12	78.94 ± 0.48	89.28 ± 0.33
Our AMTNet+KD	ResNet-12	**79.52 ± 0.48**	**89.60 ± 0.33**
Boosting [7]	WRN-28	73.60 ± 0.30	86.00 ± 0.20
Fine-tuning [4]	WRN-28	76.58 ± 0.68	85.79 ± 0.50
Our AMTNet	WRN-28	**80.38 ± 0.48**	**89.89 ± 0.32**

6.3 Model Analysis

Effectiveness of Adaptive Metrics Module: In Tab. 4, different metric methods are compared without rotation task using Res-12 backbone. The proposed AMM is 4.02% and 4.03% higher than the best individual Cosine metric on 1-shot and 5-shot tasks, respectively. Comparing to any individual metric method, NMM achieves impressive accuracy gains which indicates increasing the metric diversity boosts the performance. And AMM gains a further accuracy improvement around 2% upon NMM via alleviating the metrics competition problem. Due to length constraints, the results of backbone WRN-28 is shown in the APPENDIX, which indicates the similar performance compared with ResNet-12.

Specifically, the results in Tab. 4 demonstrate the effectiveness of our approaches: (i) Comparing NMM to Coupled: decoupling metrics fusion into predictions fusion and losses fusion is helpful;

Table 4: The results on 5-way classification with different metric methods under ResNet-12 backbone. The Metric and Global loss weights are set to 0.5 and 1.0 respectively, and the Rotation classifier is not applied.

Metric	$\hat{Y}$	$\mathcal{L}_M$	Param	*mini*ImageNet	
				1-shot	5-shot
Relation	$\hat{Y}_r$	$\mathcal{L}_r$	8.50M	61.84 ± 0.48	77.48 ± 0.35
Euclidean	$\hat{Y}_e$	$\mathcal{L}_e$	7.75M	62.90 ± 0.48	78.03 ± 0.35
Cosine	$\hat{Y}_c$	$\mathcal{L}_c$	7.75M	64.45 ± 0.47	79.20 ± 0.34
Coupled	Eq. 7	$\mathcal{L}_y$	8.50M	66.04 ± 0.47	80.03 ± 0.34
NMM	Eq. 7	Eq. 8	8.50M	66.48 ± 0.47	80.26 ± 0.34
AMM-V1	Eq. 7	Eq. 10	8.50M	67.50 ± 0.47	81.18 ± 0.33
AMM-V2	Eq. 9	Eq. 10	8.50M	68.02 ± 0.46	82.64 ± 0.32
AMM	Eq. 9	Eq. 11	8.50M	**68.47 ± 0.46**	**83.23 ± 0.31**

Table 5: The results on 5-way classification about the influence of the hyper-parameter α as introduced in Eq. 11. The setting is consistent with Tab. 4.

Metric	$\hat{Y}$	$\mathcal{L}_M$	α	*mini*ImageNet	
				1-shot	5-shot
AMM-V2	Eq. 9	Eq. 10	-	68.02 ± 0.46	82.64 ± 0.32
AMM	Eq. 9	Eq. 11	1.0	68.06 ± 0.47	83.11 ± 0.32
AMM	Eq. 9	Eq. 11	0.5	68.18 ± 0.46	**83.23 ± 0.31**
AMM	Eq. 9	Eq. 11	0.1	**68.47 ± 0.46**	82.89 ± 0.32

(ii) Comparing AMM-V1 to NMM: multi-metrics loss fusion is able to learn a more robust embedding; (iii) Comparing AMM-V2 to AMM-V1: the adaptive layer controls the contributions of different metrics to alleviate metrics competition; (iv) Comparing AMM to

Table 6: The results on 5-way classification on *mini*ImageNet about the influence of Global Adaptive Loss (GAL) employed in AMTNet under ResNet-12 and WRN-28 backbones. There are two experimental groups: Group1, showing the influence of Global and Rotation tasks. Group2, searching the experimental optimal hyper-parameter λ of GAL as introduced in Eq. 13.

Exp. group	λ	Loss weights			ResNet-12		WRN-28	
		Metric	Global	Rotation	1-shot	5-shot	1-shot	5-shot
Group1	-	0.5	-	-	62.43 ± 0.50	80.61 ± 0.33	61.15 ± 0.50	76.90 ± 0.38
	-	0.5	-	1.0	65.33 ± 0.50	80.43 ± 0.35	63.84 ± 0.50	78.50 ± 0.38
	-	0.5	1.0	-	68.47 ± 0.46	83.23 ± 0.31	67.07 ± 0.48	82.59 ± 0.31
	-	0.5	1.0	1.0	**68.80 ± 0.47**	**83.77 ± 0.30**	**69.07 ± 0.48**	**84.36 ± 0.29**
Group2	0.0	0.5	w_G	w_R	67.91 ± 0.48	83.15 ± 0.31	67.32 ± 0.50	82.59 ± 0.32
	0.5	0.5	w_G	w_R	**69.17 ± 0.46**	**83.88 ± 0.31**	68.61 ± 0.48	84.01 ± 0.30
	1.0	0.5	w_G	w_R	68.94 ± 0.46	83.76 ± 0.31	69.32 ± 0.47	84.14 ± 0.31
	1.5	0.5	w_G	w_R	68.53 ± 0.46	82.97 ± 0.31	**70.05 ± 0.46**	**84.55 ± 0.29**
	2.0	0.5	w_G	w_R	67.95 ± 0.46	82.76 ± 0.31	69.85 ± 0.46	84.52 ± 0.29
	3.0	0.5	w_G	w_R	66.94 ± 0.46	82.11 ± 0.32	69.93 ± 0.46	84.51 ± 0.29
	4.0	0.5	w_G	w_R	66.31 ± 0.46	81.39 ± 0.32	69.25 ± 0.46	84.33 ± 0.30
	6.0	0.5	w_G	w_R	61.45 ± 0.47	81.02 ± 0.33	69.62 ± 0.46	83.63 ± 0.30

Table 7: The results of AMTNet+KD on 5-way classification. The hyper-parameter β is introduced in Eq. 14.

AMTNet	Backbone	β	λ	*mini*ImageNet	
				1-shot	5-shot
Teacher	ResNet-12	-	0.5	69.17 ± 0.46	83.88 ± 0.31
Student	ResNet-12	1.0	0.5	69.06 ± 0.47	83.61 ± 0.31
		0.75	0.5	68.96 ± 0.47	83.86 ± 0.31
		0.5	0.5	**69.28 ± 0.47**	**83.99 ± 0.31**
		0.25	0.5	69.22 ± 0.47	83.71 ± 0.31
		0.1	0.5	69.16 ± 0.47	83.57 ± 0.31
Teacher	WRN-28	-	1.5	70.05 ± 0.46	84.55 ± 0.29
Student	ResNet-12	1.0	0.5	68.59 ± 0.47	83.72 ± 0.31
		0.75	0.5	**69.48 ± 0.47**	**84.22 ± 0.31**
		0.5	0.5	69.26 ± 0.46	83.84 ± 0.31
		0.25	0.5	69.13 ± 0.47	83.87 ± 0.31
		0.1	0.5	69.27 ± 0.46	83.84 ± 0.31

AMM-V2: the KL regularization $\mathcal{L}_{KL}$ increases consistencies between different metric distributions to alleviate the metric criterion discordance problem.

Tab. 5 shows the influence of the hyper-parameter α as introduced in Eq. 11. The KL regularization $\mathcal{L}_{KL}$ considers AMM $\hat{Y}$ as the teacher-metric and individual metric $\hat{Y}_j$ as the student-metric. The optimal values of α are 0.1 on 1-shot task and 0.5 on 5-shot task respectively, which indicates that AMM obtains a more stable teacher-metric on 5-shot task than on 1-shot task.

Influence of Global Adaptive Loss: As illustrated in Tab. 6, our AMTNet obtains its best results as setting λ to 0.5 and 1.5 under ResNet-12 and WRN-28 backbones respectively. Based on the proposed AMM and GAL modules, AMTNet achieves large accuracy improvements (maximum up-to 8.9%) on 1-shot and 5-shot tasks comparing to the model without GAL (first row in Tab. 6). As shown in **Group2**, the recommended setting is $\lambda \in [0.5, 2.0]$, and our method under WRN-28 gets competitive performance on a large range of $\lambda \in [1.0, 4.0]$. In **Group1**, the results indicate that the auxiliary tasks (i.e. Global classification and Rotation classification) are useful for training a more robust embedding leading to an accuracy improvement.

Effectiveness of Knowledge Distillation: As illustrated in Tab. 7, we obtain a further accuracy improvements with the usage of knowledge distillation. The Student-AMTNet (ResNet-12) achieves its best results as setting β to 0.75 with Teacher-AMTNet (WRN-28). Thus, our AMTNet+KD approach encourages adopting a large backbone (WRN-28) to achieve better accuracy performance, then using knowledge distillation technology to improve the accuracy on a smaller (ResNet-12) backbone.

7 CONCLUSION

In this paper, we investigate the contributions of different distance metrics, and propose an effective few-shot classification framework, named AMTNet, which consists of two novel structures: *Adaptive Metrics Module* (AMM) and the *Global Adaptive Loss* (GAL). Specifically, AMM integrates both flexible and fixed distance metrics to achieve mutual complementarity in embedding learning and metric-decision making. To deal with the metrics competition problem, the proposed AMM decouples metrics fusion into predictions fusion and losses fusion, and further utilizes KL regularization to increase consistencies between different metric distributions. Moreover, GAL further riches the embedding by providing more supervised information from multiple tasks which gains a significant performance improvement. Extensive experiments show that our method is effective for few-shot classification, and achieves new state-of-the-art results on *mini*ImageNet and *tiered*ImageNet benchmark datasets.

ACKNOWLEDGEMENT

This work was supported by the National Key Research and Development Program of China (2021ZD0111000), National Natural Science Foundation of China No. 62176092, Shanghai Science and Technology Commission No.21511100700, Natural Science Foundation of Shanghai (20ZR1417700).

REFERENCES

[1] Kendall Alex, Gal Yarin, and Cipolla Roberto. 2018. Multi-Task Learning Using Uncertainty to Weigh Losses for Scene Geometry and Semantics. In *CVPR*.

[2] Zhang Baoquan, Li Xutao, Ye Yunming, Huang Zhichao, and Zhang Lisai. 2021. Prototype Completion with Primitive Knowledge for Few-Shot Learning. In *CVPR*.

[3] Medina Carlos, Devos Arnout, and Grossglauser Matthias. 2020. Self-supervised prototypical transfer learning for few-shot classification. In *arXiv preprint arXiv:2006.11325*.

[4] Guneet S Dhillon, Pratik Chaudhari, Avinash Ravichandran, and Stefano Soatto. 2019. A baseline for few-shot image classification. In *arXiv preprint arXiv:1909.02729*.

[5] Carl Doersch, Ankush Gupta, and Andrew Zisserman. 2020. CrossTransformers: spatially-aware few-shot transfer. In *NeurIPS*.

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

[7] Spyros Gidaris, Andrei Bursuc, Nikos Komodakis, Patrick Pérez, and Matthieu Cord. 2019. Boosting few-shot visual learning with self-supervision. In *ICCV*.

[8] Spyros Gidaris and Nikos Komodakis. 2019. Generating classification weights with gnn denoising autoencoders for few-shot learning. In *CVPR*.

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

[10] Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. 2015. Distilling the knowledge in a neural network. In *arXiv preprint arXiv:1503.02531*.

[11] Ruibing Hou, Hong Chang, MA Bingpeng, Shiguang Shan, and Xilin Chen. 2019. Cross Attention Network for Few-shot Classification. In *NeurIPS*.

[12] Kim Jaekyeom, Kim Hyoungseok, and Kim Gunhee. 2020. Model-agnostic boundary-adversarial sampling for test-time generalization in few-shot learning. In *ECCV*.

[13] Hong Jie, Fang Pengfei, Li Weihao, Zhang Tong, Simon Christian, Harandi Mehrtash, and Petersson Lars. 2021. Reinforced Attention for Few-Shot Learning and Beyond. In *CVPR*.

[14] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. 2012. Imagenet classification with deep convolutional neural networks. In *NeurIPS*.

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

[16] Chen Liu, Yanwei Fu, Chengming Xu, Siqian Yang, Jilin Li, Chengjie Wang, and Li Zhang. 2021. Learning a Few-shot Embedding Model with Contrastive Learning. In *AAAI*.

[17] Andrychowicz Marcin, Denil Misha, Gomez Sergio, W. Hoffman Matthew, Pfau David, Schaul Tom, Shillingford Brendan, and de Freitas Nando. 2018. Learning to learn by gradient descent by gradient descent. In *NeurIPS*.

[18] Tsendsuren Munkhdalai, Xingdi Yuan, Soroush Mehri, and Adam Trischler. 2018. Rapid adaptation with conditionally shifted neurons. In *ICML*.

[19] Rodríguez Pau, Laradji Issam, Drouin Alexandre, and Lacoste Alexandre. 2020. Embedding Propagation: Smoother Manifold for Few-Shot Classification. In *ECCV*.

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

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

[22] Mamshad Nayeem Rizve, Salman Khan, Fahad Shahbaz Khan, and Mubarak Shah. 2021. Exploring Complementary Strengths of Invariant and Equivariant Representations for Few-Shot Learning. In *CVPR*.

[23] Andrei A Rusu, Dushyant Rao, Jakub Sygnowski, Oriol Vinyals, Razvan Pascanu, Simon Osindero, and Raia Hadsell. 2019. Meta-learning with latent embedding optimization. In *ICLR*.

[24] Christian Simon, Piotr Koniusz, Richard Nock, and Mehrtash Harandi. 2020. Adaptive Subspaces for Few-Shot Learning. In *CVPR*.

[25] Jake Snell, Kevin Swersky, and Richard Zemel. 2017. Prototypical networks for few-shot learning. In *NeurIPS*.

[26] Jong-Chyi Su, Subhransu Maji, and Bharath Hariharan. 2020. When does self-supervision improve few-shot learning?. In *ECCV*.

[27] Flood Sung, Yongxin Yang, Li Zhang, Tao Xiang, Philip HS Torr, and Timothy M Hospedales. 2018. Learning to compare: Relation network for few-shot learning. In *CVPR*.

[28] Yonglong Tian, Yue Wang, Dilip Krishnan, Joshua B Tenenbaum, and Phillip Isola. 2020. Rethinking Few-Shot Image Classification: a Good Embedding Is All You Need?. In *ECCV*.

[29] Oriol Vinyals, Charles Blundell, Timothy Lillicrap, Daan Wierstra, et al. 2016. Matching networks for one shot learning. In *NeurIPS*.

[30] Zeyuan Wang, Yifan Zhao, Jia Li, and Yonghong Tian. 2020. Cooperative bi-path metric for few-shot learning. In *ACMMM*.

[31] Ziyang Wu, Yuwei Li, Lihua Guo, and Kui Jia. 2019. PARN: Position-Aware Relation Networks for Few-Shot Learning. In *ICCV*.

[32] Chengming Xu, Yanwei Fu, Chen Liu, Chengjie Wang, Jilin Li, Feiyue Huang, Li Zhang, and Xiangyang Xue. 2021. Learning Dynamic Alignment via Meta-filter for Few-shot Learning. In *CVPR*.

[33] Luo Xu, Wei Longhui, Wen Liangjian, Yang Jinrong, Xie Lingxi, Xu Zenglin, and Tian Qi. 2021. Rectifying the Shortcut Learning of Background for Few-Shot Learning. In *NeurIPS*.

[34] Liu Yanbin, Lee Juho, Park Minseop, Kim Saehoon, Yang Eunho, Ju Hwang Sung, and Yang Yi. 2018. Learning to propagate labels: Transductive propagation network for few-shot learning. In *ICLR*.

[35] Han-Jia Ye, Hexiang Hu, De-Chuan Zhan, and Fei Sha. 2020. Few-shot learning via embedding adaptation with set-to-set functions. In *CVPR*.

[36] Chi Zhang, Yujun Cai, Guosheng Lin, and Chunhua Shen. 2020. DeepEMD: Few-Shot Image Classification with Differentiable Earth Mover's Distance and Structured Classifiers. In *CVPR*.

[37] Chen Zhengyu, Ge Jixie, Zhan Heshen, Huang Siteng, and Wang Donglin. 2021. Pareto Self-Supervised Training for Few-Shot Learning. In *CVPR*.

[38] Shen Zhiqiang, Liu Zechun, Qin Jie, Savvides Marios, and Cheng Kwang-Ting. 2021. Partial Is Better Than All:Revisiting Fine-tuning Strategy for Few-shot Learning. In *AAAI*.

A APPENDIX

This section provides more details of our proposed method and experimental results, which are omitted in the main paper due to space limitation.

A.1 Structure of the Applied Relation Module

The detail structure of the variant Relation Module adopted in AMTNet is shown in Fig. 4.

A.2 Derivation of Losses Fusion in AMM

The detailed formula derivation of losses fusion as in Eq.10 for AMM, is described as follow. To realize task-dependent uncertainty in multi-task loss function [1], the classification likelihood originally calculated by Eq.1 is now modified into a *scaled* version:

$$\hat{y}_j(y=k|Q;\theta_j) = \frac{\exp\left(-\frac{1}{\theta_j^2}\cdot d_j\left(Q,P^k\right)\right)}{\sum_{i=1}^{N}\exp\left(-\frac{1}{\theta_j^2}\cdot d_j\left(Q,P^i\right)\right)}, \tag{15}$$

where θ_j is a positive scalar of which the parameter's magnitude determines how 'uniform' (flat) the discrete classification distribution is. This relates to its uncertainty, as measured in entropy. It can be interpreted as a Boltzmann distribution where the input is scaled by θ_j^2 (often referred as *temperature*). Then the classification

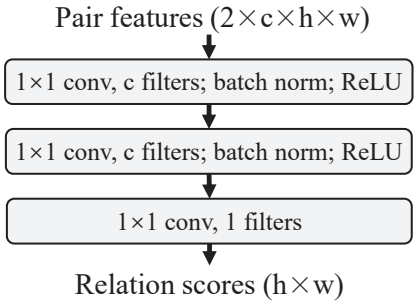


Figure 4: The variant Relation module applied in AMM.

cross-entropy loss $\mathcal{L}_j(\hat{y}_j;\theta_j)$ for this output can be written as:

$$
\begin{aligned}
&= -\log \hat{y}_j(y = y_i^q | Q_i;\theta_j) \\
&= -\log \frac{\exp\left(-\frac{1}{\theta_j^2}\cdot d_j\left(Q,P^k\right)\right)}{\sum_{i=1}^{N}\exp\left(-\frac{1}{\theta_j^2}\cdot d_j\left(Q,P^i\right)\right)} \\
&= \frac{1}{\theta_j^2}\cdot d_j\left(Q,P^k\right) + \log\sum_{i=1}^{N}\exp\left(-\frac{1}{\theta_j^2}\cdot d_j\left(Q,P^i\right)\right) \\
&= -\frac{1}{\theta_j^2}\log\frac{\exp\left(-d_j\left(Q,P^k\right)\right)}{\sum_{i=1}^{N}\exp\left(-d_j\left(Q,P^i\right)\right)} \\
&\quad + \left(\log\theta_j^2\cdot\frac{\sum_{i=1}^{N}\exp\left(-\frac{1}{\theta_j^2}\cdot d_j\left(Q,P^i\right)\right)}{\sum_{i=1}^{N}\exp\left(-d_j\left(Q,P^i\right)\right)}\right) \\
&\approx \frac{1}{\theta_j^2}\mathcal{L}_j + log\theta_j^2.
\end{aligned}
\tag{16}
$$

Therefore, based on Eq.16, we obtain the multi-task loss for metric losses fusion as expressed in Eq.10.

A.3 AMTNet Training with Patch-wise Strategy

Supplement to Sec.5.1, the detailed objective functions for Metric, Global and Rotation classifiers are illustrated as follow.

To produce precise embeddings, each spatial position of the query features are constrained to be independently classified which is called as the patch-wise classification strategy as referred in [11]. For **Metric classifier**, each local feature Q_n in the n^{th} position of Q, is classified into N support classes. Formally, the probability of predicting Q_n as k^{th} class is:

$$
\hat{y}_j(y = k|Q_n) = \frac{\exp\left(-d_j\left(Q_n, GAP(P^k)\right)\right)}{\sum_{i=1}^{N}\exp\left(-d_j\left(Q_n, GAP(P^i)\right)\right)}, \tag{17}
$$

where GAP represents global average pooling to obtain class feature. According to the true N-way few-shot class label $\bar{y}^q$, each individual metric classification loss is then defined as:

$$
\mathcal{L}_j = -\sum_{i=1}^{n_q}\sum_{n=1}^{h\times w}\log\hat{y}_j(y = \bar{y}_i^q|(Q_n)_i). \tag{18}
$$

The **Global classifier** categorizes the query into all available classes of training set, and its loss is:

$$
\begin{aligned}
\mathcal{L}_G &= PCE(Q, C^q) \\
&= -\sum_{i=1}^{n_q}\sum_{n=1}^{h\times w} C_i^q \log\left(softmax(W(Q_n)_i)\right).
\end{aligned}
\tag{19}
$$

where, PCE is defined as the patch-wise cross-entropy function, and W is the $Linear$ layer. Similarly, the loss of **Rotation classifier** is computed as $\mathcal{L}_R = PCE(Q, B^q)$.

Table 8: The results on 5-way 1-shot classification with different metric methods based on AMTNet framework with WRN-28 backbone.

Metric	AMM	NMM	Relation	Euclidean	Cosine
Param	36.25M	36.25M	36.25M	35M	35M
1-shot	70.05%	68.46%	65.12%	66.45%	67.67%