LEARN TO LEARN CONSISTENTLY

Anonymous authors
Paper under double-blind review

ABSTRACT

In the few-shot learning problem, a model trained on a disjoint meta-train dataset is required to address novel tasks with limited novel examples. A key challenge in few-shot learning is the model's propensity to learn biased shortcut features(e.g., background, noise, shape, color), which are sufficient to distinguish the few examples during fast adaptation but lead to poor generalization. In our work, we observed when the model learns with higher consistency, the model tends to be less influenced by shortcut features, resulting in better generalization. Based on the observation, we propose a simple yet effective meta-learning method named Meta Self-Distillation. By maximizing the consistency of the learned knowledge during the meta-train phase, the model initialized by our method shows better generalization in the meta-test phase. Extensive experiments demonstrate that our method improves the model's generalization across various few-shot classification scenarios and enhances the model's ability to learn consistently.

1 INTRODUCTION

Few-shot learning aims to address novel tasks with a limited number of examples, typically through rapid adaptation of a model trained on a dataset with disjoint labels. Many approaches tackle this issue from the perspective of meta-learning(Finn et al., 2017; Lee et al., 2019; Ravi & Larochelle, 2016; Lake & Baroni, 2023). Methods such as Model-Agnostic meta-Learning (MAML) (Finn et al., 2017) and its variants (Raghu et al., 2019; Ye & Chao, 2021; Antoniou et al., 2018; Kao et al., 2021; Nichol et al., 2018) aim to learn initialized parameters for a model with prior knowledge for fast adaptation. Recent research has explored more challenging scenarios, such as cross-domain few-shot learning (Ullah et al., 2022; Triantafillou et al., 2019; Tseng et al., 2020; Guo et al., 2020), where the novel task belongs to a different domain and label set than the training dataset.

A key challenge in various few-shot learning problems is the model's tendency to learn biased shortcut features (e.g., background, noise, shape, color) from limited examples (Shah et al., 2020; Teney et al., 2022; Lyu et al., 2021; Le et al., 2021). These shortcut features may suffice to distinguish the few classes during rapid adaptation but result in poor generalization. Several solutions have been proposed to address these issues. Although these approaches partially mitigate the problem, they often require additional resources or learn generalized features only within the meta-train dataset (Le et al., 2021; Zhou et al., 2023; Liu et al., 2020; Dvornik et al., 2020; Snell et al., 2017). From the perspective of meta-learning, we ask the following question: *Can we make the initialized model more inclined to learn generalization features rather than shortcut features when addressing novel tasks?*

This problem is challenging to address directly, as identifying generalized versus shortcut features in the data is difficult. In our study, we generate different views of the same data through data augmentation, which makes these views have different shortcut features but similar generalized features. We use these views to update the model and observe that when model learning with better consistency tends to exhibit better generalization. This implies that when tasks are learned with higher consistency by the model, the model is less influenced by the shortcut features and reaches higher accuracy. At this point, if we can enhance the model's consistency of learning across all tasks, we can make the model less influenced by the shortcut feature and more inclined to learn generalized features.

Based on this observation and inspired by the idea of self-distillation (Caron et al., 2020; Chen & He, 2021; Caron et al., 2021), we proposed meta self-distillation, which aims to maximize the

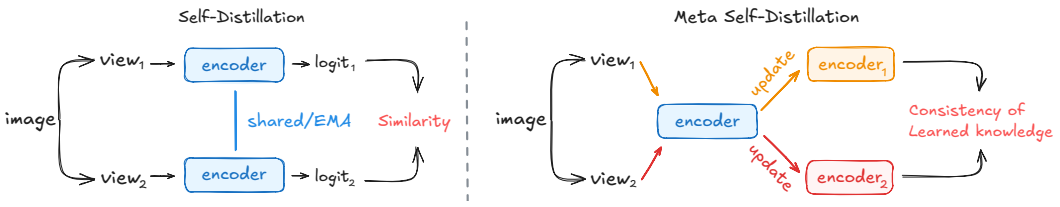

Figure 1: **The core idea between self-distillation and meta self-distillation.** Self-distillation aims to make the deep representation of different views closer, while meta self-distillation aims to learn consistent knowledge from the different views of the same image.

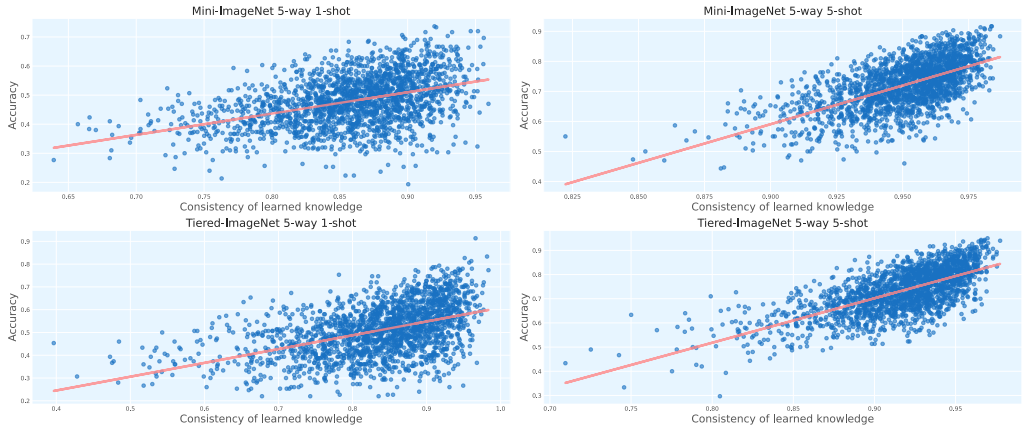

Figure 2: **The consistency versus accuracy of the model initialized by MAML across different tasks.** The results demonstrate a clear trend: as the model's consistency learned from the task increases, the average accuracy in predicting query data improves.

consistency of models updated from the initialized model by using augmented views of the same data. Specifically, in the inner loop, we augment the same tasks to update the initialized model independently. In the outer loop, we maximize the consistency of the outputs for the same query data produced by the differently updated models. This approach enhances the initialized model's ability to learn consistently, thereby improving the generalization of the initialized model. We evaluate our method in three different settings of few-shot learning, which has demonstrated the effectiveness of our method.

In summary, our contributions are as follows:

- We observed that the consistency of model learning could serve as an indicator of the model's inclination towards learning shortcut features that lead to overfitting.
- We proposed a meta-learning method named meta self-distillation(MSD). By maximizing the consistency of learned knowledge, MSD improves the initialized model's ability to learn more consistently, thereby making the model more inclined to learn generalized features.
- Extensive experiments demonstrate that our method achieves remarkable performance across various few-shot scenarios and significantly enhances the model's ability to learn consistently in unseen tasks.

2 Related Work

2.1 Few-shot learning

Few-shot learning aims to address novel tasks with a limited number of examples, typically through the rapid adaptation of a model trained on base classes, which are disjoint from the classes in novel tasks. Solutions to few-shot learning are primarily categorized into meta-learning and transfer learn-

ing approaches. meta-learning (Antoniou et al., 2018; Finn et al., 2017; Ye & Chao, 2021) aims to train a model with prior knowledge that can fast adapt to novel tasks. Transfer learning (Tian et al., 2020; Mangla et al., 2020; Liu et al., 2021) focuses on developing a generalized feature extractor from base classes that can generalize to novel tasks. Traditionally, few-shot learning assumes that base and novel classes originate from the same domain but differ in categories. Recent studies have extended this to cross-domain few-shot learning, where base and novel classes belong to different domains (Ullah et al., 2022; Triantafillou et al., 2019; Tseng et al., 2020; Guo et al., 2020). A critical challenge in few-shot learning is that during fast adaptation, models tend to learn shortcut features, leading to overfitting on novel tasks (Shah et al., 2020; Teney et al., 2022; Lyu et al., 2021; Le et al., 2021). Various methods have been proposed to address this issue. For instance, Poodle (Le et al., 2021) suggests using additional data to penalize out-of-distribution samples, while LDP-net (Zhou et al., 2023) employs local and global knowledge distillation to enable the model to learn more diverse features from the meta-training dataset. Although these methods mitigate the problem to some extent, they often require additional data or parameters to adapt to unseen tasks and domains. Alternatively, some approaches train a powerful feature extractor solely on the meta-train dataset, which may limit the model's ability to recognize unseen features in novel tasks (Le et al., 2021; Zhou et al., 2023; Liu et al., 2020; Dvornik et al., 2020; Snell et al., 2017). Our method aims to make the initialized model inclined to learn generalized features, thereby avoiding such limitations.

2.2 META-LEARNING

Meta-learning, also known as learning to learn, aims to learn initialized parameters with prior knowledge for fast adaptation. It is mainly divided into metric-based meta-learning, represented by ProtoNet (Snell et al., 2017), and optimize-based meta-learning, represented by MAML (Finn et al., 2017). Metric-based meta-learning improves model representation by bringing the representation between the support data and the query data that belong to the same category closer, typically not requiring fine-tuning during the meta-test phase. Optimize-based meta-learning aims to provide the initial parameters with prior knowledge, offering better generalization performance when fine-tuning on novel category samples. This category includes algorithms like MAML (Finn et al., 2017) and its variants, such as (Ye & Chao, 2021), which utilizes a single vector to replace the network's classification head weight, thus preventing the permutation in the meta-test phase. MAML++ (Antoniou et al., 2018) enhances MAML's performance by addressing multiple optimization issues encountered by MAML, while ANIL (Raghu et al., 2019) improves MAML's performance by freezing the backbone during the inner loop. In our work, we mainly focus on the optimize-based meta-learning. From the perspective of meta-learning, our goal is to train initialized parameters that incline to learn generalized features rather than shortcut features, thereby enhancing accuracy in the few-shot learning problems.

2.3 SELF-DISTILLATION

Self-distillation is a variant of contrastive learning (Caron et al., 2020; Chen & He, 2021), which is trained by bringing the representations of positive instance pairs closer without using negative pairs. BYOL (Grill et al., 2020)utilizes the exponential moving average of the network to produce the target of an online network. SimSiam (Chen & He, 2021) further explored how self-distillation avoids collapse in a self-supervised setting. (Allen-Zhu & Li, 2020) suggests that self-distillation can serve as an implicit ensemble distillation, allowing the model to distinguish more view features. Self-distillation is an effective method to enhance the model's feature extraction capabilities and can be combined with meta-learning (Li et al., 2022; Ni et al., 2021). Typically, self-distillation aims to maximize the similarity of the representations across different views. Different from the typical self-distillation that directly aligns the representation, we propose to use meta self-distillation to maximize the consistency of the different updated models' outputs. In this way, we can make the initialized model less influenced by shortcut features when addressing a new task.

3 PRELIMINARY

Here, we provide an overview of the fundamental setting and problem for few-shot learning classification, along with an introduction to model-agnostic meta-learning (MAML).

3.1 PROBLEM DEFINITION FOR FEW-SHOT CLASSIFICATION

Following (Vinyals et al., 2016; Chen et al., 2019; Wang et al., 2020), We define the few-shot classification problem(FSL) as an $\mathcal{N}$-way $\mathcal{K}$-shot task, where there are $\mathcal{N}$ classes, each containing $\mathcal{K}$-labeled support samples. Typically, $\mathcal{K}$ is small, such as 1 or 5. The data used to attempt to update the model is defined as the support data $\mathcal{S} = \{x_s, y_s\}$, where each x_s represents the model's input, and y_s denotes the corresponding label for x_s. The data used to evaluate the effectiveness of the model updates is defined as the query data $\mathcal{Q} = \{x_q, y_q\}$, which has the same class as the support data, but the samples contained in the query set are different from those in the support set. The FSL task is defined as the problem of learning to correctly classify the query data $\mathcal{Q}$ with the support data $\mathcal{S}$, which can be written as follows:

$$\arg\min_{\theta} \mathbb{E}_{\mathcal{S},\mathcal{Q}\sim\mathcal{D}_{\text{meta-test}}} \left[\mathcal{L}_{FSL}(\theta, \mathcal{S}, \mathcal{Q})\right] \tag{1}$$

If the model is randomly initialized and directly fine-tuned on the limited support data, the model will overfit. To address that, we need to transfer knowledge from seen data to the unseen data. The seen data used in FSL is referred to as the meta-train set, and the unseen data is referred to as the meta-test set. The labels in the two sets are disjoint, and in the cross-domain few-shot learning, the domains of the two sets are also different. The goal of FSL is to pretrain or initialize the parameters by using the meta-train set and generalize to the unseen task sampled from the meta-test set.

3.2 MODEL-AGNOSTIC META-LEARNING

Model Agnostic Meta-Learning (MAML) (Finn et al., 2017) is a meta-learning framework. The objective of MAML is to learn initialized parameters θ with prior knowledge, such that after a few steps of standard training on the support data, the model can generalize well on the query data. The objective can be as follows:

$$\arg\min_{\theta} \mathbb{E}_{\mathcal{S},\mathcal{Q}\sim\mathcal{D}_{\text{meta-train}}} \left[\mathcal{L}(\mathcal{U}^k(\theta, \mathcal{S}), \mathcal{Q})\right] \tag{2}$$

Where $\mathcal{U}^k$ denotes k updates of the parameter θ using tasks sampled from the task distribution, which corresponds to adding a sequence of gradient vectors to the initialized parameters:

$$\mathcal{U}^k(\theta, \mathcal{S}) = \theta - \sum_{i=1}^{k} \alpha \cdot \frac{\partial \mathcal{L}(\mathcal{U}^{i-1}(\theta, \mathcal{S}), \mathcal{S})}{\partial \theta}, \quad \mathcal{U}^0(\theta, \mathcal{S}) = \theta \tag{3}$$

The process of updating the parameters with support data is referred to as *inner loop process*, where α is the stepsize of the inner loop. Subsequently, the query data $\mathcal{Q}$ is used to evaluate $\mathcal{U}^k(\theta, \mathcal{S})$, and directly updating the initial parameters θ, which known as the *outer loop process*. The outer loop commonly employs SGD for updates, and the update process can be computed as follows:

$$\theta' = \theta - \beta \cdot \frac{\partial \mathcal{L}(\mathcal{U}^k(\theta, \mathcal{Q}), \mathcal{S})}{\partial \theta} \tag{4}$$

Where β is the learning rate of the outer loop. By minimizing the loss across sampled tasks, MAML enables the parameters to learn prior knowledge from the meta-train set.

4 LEARN TO LEARN CONSISTENTLY

4.1 WHY LEARN CONSISTENTLY IN FSL

Previous studies have indicated that in few-shot learning (FSL) scenarios, models tend to learn shortcut features (e.g., background, noise, shape, color) from limited examples (Shah et al., 2020; Teney et al., 2022; Lyu et al., 2021; Le et al., 2021). These shortcut features may suffice to distinguish the few classes during rapid adaptation but often lead to poor generalization. From the perspective

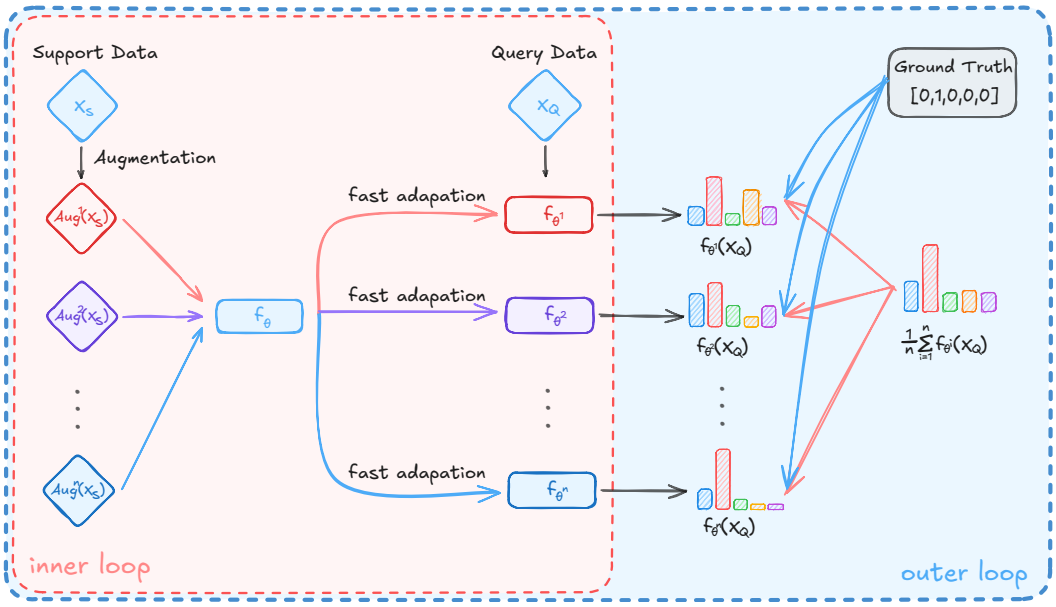

Figure 3: **An overview of the proposed MSD.** In the inner loop, MSD first uses different augmented support data to update the f_θ. In the outer loop, then maximizes the consistency among the outputs of the same query data with different update versions of the initial model

Algorithm 1: Evaluate the consistency and accuracy of model initialized by MAML

Given the learned initialization θ by MAML
for $t \in \{1, \cdots, T\}$ **do**
  Sample task $\mathcal{T} = (\mathcal{S}, \mathcal{Q}) \sim \mathcal{D}_{\text{meta-test}}$
  for $i \in \{1, \cdots, n\}$ **do**
    Random Augmented the support data: Get $\mathcal{S}_i = \mathcal{A}ug(\mathcal{S})$
    Update θ by augmented support data $\mathcal{S}_i$: Get $\theta_i = \mathcal{U}^k(\theta, \mathcal{S}_i)$
    Get the output of the query data x_q by θ_i: Get $v_i = f_{\theta_i}(x_q)$
  end
  Record the consistency and average accuracy of the output $\{v_i\}$:
    $\mathcal{C}[t] = \frac{1}{n}\sum_{i=1}^{n} \mathcal{F}_{\text{sim}}\left(v_i, \frac{1}{n}\sum_{i=1}^{n} v_i\right), \mathcal{A}[t] = \frac{1}{n}\sum_{i=1}^{n} \mathcal{A}cc\left(v_i\right)$
end
Return $\mathcal{C}, \mathcal{A}$

of meta-learning, we aim for the initialized model to learn more generalized features, avoiding the reliance on shortcut features. However, it's hard to distinguish these features directly in practice. To solve that, We proposed that one can make the initialized model less influenced by the shortcut features by enhancing the model's consistency in learning.

To validate the point, we evaluate the consistency and accuracy of the parameters initialized by MAML. During the meta-test phase, we sample one task and augment its support data, updating the initialized parameters respectively. To evaluate the knowledge acquired from these support data, we tested the updated models using the same query data and recorded the average prediction accuracy and output consistency across the tasks. For each task group, the support data are augmented from the same data, thus containing different shortcut features and similar generalized features. Therefore, the inconsistency of the differently updated models is mainly caused by different shortcut features. When the model is more inclined to learn generalized features, its outputs for the same query data should be similar. In contrast, if the model tends to learn shortcut features, which results in overfitting, the augmented inconsistencies in these features lead to greater output variance for the same query data. The results are illustrated in Figure 2: lower consistency corresponds to lower average prediction accuracy. Additionally, as the amount of support data increases, the model is less influenced by shortcut features, and both consistency and accuracy are improved consistently. Consistency and accuracy exhibit a high degree of alignment In various settings. Therefore, we demonstrate that model consistency reflects the tendency to learn shortcut features. Based on the

observation, We propose to enhance the consistency of learning across all tasks to make the initialized model less influenced by the shortcut feature during fast adaptation. Therefore, we define the objective of "learn to learn consistently" as follows:

$$\arg\max_{\theta} \mathbb{E}_{\mathcal{S},\mathcal{Q}\sim\mathcal{D}_{\text{meta-test}}} \left[\mathcal{F}_{\text{sim}}\left(v_i, \bar{v}\right)\right] \tag{5}$$

Where $\mathcal{F}_{\text{sim}}$ is the similar function to evaluate consistency, v_i is the output of sampled query data by different updated models. When $\mathcal{F}_{\text{sim}}$ is the negative mean squared error, the objective is to minimize the variance of the output among the different updated models.

4.2 Meta Self-Distillation

Based on the objective proposed in Eq.5, we propose a meta-learning method named meta self-distillation to enhance the model's ability to learn consistently.

Meta-Train Phase. Specifically, we sample tasks from the meta-train set to obtain support and query data. Unlike MAML, which samples multiple tasks, we sample a single task and create multiple augmented versions as substitutes. Only the support data is augmented in the different augmented tasks, and the tasks share the same query data. The rationale behind this is to have the same standard when assessing the knowledge learned by the model. Let the tasks be denoted as $\mathcal{T} = \{\mathcal{S}_i, \mathcal{Q}\}$, where $\mathcal{S}_i$ represent the i-th augmented view of support data. In the inner loop, we update the model with different augmented views of the support data to obtain varied models:

$$\theta_i = \mathcal{U}^k(\theta, \mathcal{S}_i) = \theta - \sum_{i=1}^{k} \alpha \cdot \frac{\partial \mathcal{L}(\mathcal{U}^{k-1}(\theta, \mathcal{S}_i), \mathcal{S}_i)}{\partial \theta}, \quad \mathcal{U}^0(\theta, \mathcal{S}_i) = \theta \tag{6}$$

In the outer loop, we test the query with different updated versions of the parameters. Since we desire the model to extract the same knowledge from different augmented views of support data, we measure the consistency of their query outputs to assess if the knowledge learned is identical:

$$\mathcal{L}_{\text{CK}} = -\frac{1}{n}\sum_{i=1}^{n} \mathcal{F}_{\text{sim}}\left(f_{\theta_i}(x_q), \frac{1}{n}\sum_{i=1}^{n} f_{\theta_i}(x_q)\right) \tag{7}$$

Where $\mathcal{F}_{\text{sim}}$ is the similarity function. Following (Chen & He, 2021), we use cosine similarity as the similarity function in practice. Furthermore, to ensure the model fully utilizes label information and learns precise classification, we also compute the classification loss for each updated parameter by query data. The model's total loss is expressed as:

$$\mathcal{L}_{\text{total}} = \mathcal{L}_{\text{cls}} + \gamma \cdot \mathcal{L}_{\text{CK}} \tag{8}$$

Where γ represents the coefficient of consistency loss. The process of updating the initial parameters is as follows:

$$\theta' = \theta - \beta \cdot \nabla_{\theta} \mathcal{L}_{\text{total}} \tag{9}$$

Where β represents the learning rate in the outer loop. The specific process has been shown in Algorithm 2 in the Appendix.

Meta-Test Phase. During the meta-test phase, MSD is consistent with MAML. We perform fast adaptation on the input support data using SGD and classify the query data directly with the updated model.

5 Experiment

5.1 Experiment Setting

Datasets. For standard and augmented FSL evaluation, Our method was primarily evaluated on two benchmark datasets: Mini-ImageNet (Vinyals et al., 2016) and Tiered-ImageNet (Ren et al., 2018), both widely used for few-shot learning assessments. For cross-domain FSL evaluation, we use Mini-ImageNet as the source domain and use another eight datasets as the target domain, i.e., CUB, Cars, Places, Plantae, ChestX, ISIC, EuroSAT and CropDisease.

Table 1: **5way-1shot and 5way-5shot classification accuracy in standard few-shot classification task** and 95% confidence interval on Mini-ImageNet and Tiered-ImageNet (over 2000 tasks), using ResNet-12 as the backbone. NIW-Meta used ResNet-18 as the backbone.

	Mini-ImageNet		Tiered-ImageNet	
Methods	1-Shot	5-Shot	1-Shot	5-Shot
ProtoNet (Snell et al., 2017)	62.39 ± 0.20	80.53 ± 0.20	68.23 ± 0.23	84.03 ± 0.16
MAML (Finn et al., 2017)	64.42 ± 0.20	83.44 ± 0.14	65.72 ± 0.20	84.37 ± 0.16
MetaOptNet (Lee et al., 2019)	62.64 ± 0.35	78.63 ± 0.68	65.99 ± 0.72	81.56 ± 0.53
ProtoMAML (Triantafillou et al., 2019)	64.12 ± 0.20	81.24 ± 0.20	68.46 ± 0.23	84.67 ± 0.16
DSN-MR (Simon et al., 2020)	64.60 ± 0.72	79.51 ± 0.50	67.39 ± 0.82	82.85 ± 0.56
Meta-AdaM (Sun & Gao, 2024)	59.89 ± 0.49	77.92 ± 0.43	65.31 ± 0.48	85.24 ± 0.35
LA-PID (Yu et al., 2024)	63.29 ± 0.48	79.18 ± 0.43	64.77 ± 0.47	82.59 ± 0.37
NIW-Meta† (Kim & Hospedales, 2024)	65.49 ± 0.56	81.71 ± 0.17	70.52 ± 0.19	85.83 ± 0.17
MSD	$\mathbf{65.41 \pm 0.47}$	$\mathbf{84.88 \pm 0.29}$	$\mathbf{68.51 \pm 0.53}$	$\mathbf{86.87 \pm 0.34}$

Table 2: **5way-5shot classification accuracy in cross-domain few-shot classification task** (over 2000 tasks), using ResNet-12 as the backbone. Only the meta-train set of Mini-ImageNet is used during the meta-train phase.

	CUB	Cars	Places	Plantae	Euro	ISIC	CropD	ChestX
GNN (Garcia & Bruna, 2017)	62.87	43.70	70.91	48.51	78.69	42.54	83.12	23.87
GNN+FT (Tseng et al., 2020)	64.97	46.19	70.70	49.66	78.02	40.87	87.07	24.28
TPN+ATA (Wang & Deng, 2021)	70.14	55.23	73.87	59.02	85.47	49.83	93.56	24.74
GNN+ATA (Wang & Deng, 2021)	66.22	49.14	75.48	52.69	83.75	44.91	90.59	24.32
MatchingNet+AFA (Hu & Ma, 2022)	59.46	46.13	68.87	52.43	69.63	39.88	80.07	23.18
GNN+AFA (Hu & Ma, 2022)	68.25	49.28	**76.21**	54.26	85.58	46.01	88.06	25.02
LDP-net (Zhou et al., 2023)	**70.39**	52.84	72.90	58.49	82.01	48.06	89.40	26.67
GNN +FAP (Zhang et al., 2024)	67.66	50.20	74.98	54.54	82.52	47.60	91.79	25.31
RFS+MLP (Bai et al., 2024)	-	-	-	-	83.14	46.02	66.87	**29.09**
MSD	<u>70.22</u>	**58.55**	<u>75.59</u>	**60.81**	**85.65**	**51.54**	**95.12**	<u>28.26</u>

The Mini-ImageNet dataset comprises 100 classes, each containing 600 samples. Following prior work, we divided the 100 classes into training, validation, and test sets, containing 64, 16, and 20 classes, respectively. The Tiered-ImageNet dataset encompasses 608 fine-grained classes, which are categorized into 34 higher-level classes. In alignment with previous studies, we divided these higher-level classes into training, validation, and test sets, comprising 20, 6, and 8 higher-level classes, respectively. Tiered-ImageNet is designed to consider class similarity when segmenting the dataset, ensuring a significant distributional difference between training and test data. CUB, Cars, Places, and Plantae proposed in (Tseng et al., 2020) contain natural images of different properties. ChestX, ISIC, EuroSAT and CropDisease proposed in (Guo et al., 2020) are cross-domain datasets from the domain of medicine, agriculture, and remote sensing, which have significant domain shifts. All the images are resized to 84×84 pixels following common practice.

Backbone Model. For our model evaluation, following (Lee et al., 2019), we employed a ResNet-12 (He et al., 2016) architecture, noted for its broader widths and Dropblock modules as introduced by (Ghiasi et al., 2018). This backbone is broadly used across numerous few-shot learning algorithms. Additionally, we follow the original MAML approach, utilizing a 4-layer convolutional neural network(Conv4) (Vinyals et al., 2016). Following the recent practice (Ye et al., 2020; Qiao et al., 2018; Rusu et al., 2018), The models' weights are pre-trained on the meta-train set to initialize.

Experiment Details. The other details are listed in the Appendix A.1

5.2 Results

We evaluate our method under three settings: standard few-shot learning problems, cross-domain few-shot learning problems, and augmented few-shot learning problems.

Table 3: **5way-1shot and 5way-5shot classification accuracy** in augmented few-shot classification task and 95% confidence interval on Mini-ImageNet and Tiered-ImageNet (over 2000 tasks), using Conv4 as the backbone.the terms "strong" and "weak" denote the varying levels of augmentation applied to the support data in the meta-test phase.

		Mini-ImageNet (Strong)		Mini-ImageNet (Weak)	
Methods	Backbone	1-Shot	5-Shot	1-Shot	5-Shot
MAML	Conv4	28.13 ± 0.29	37.77 ± 0.31	35.89 ± 0.35	49.54 ± 0.36
MSD + MAML	Conv4	$\mathbf{30.64 \pm 0.30}$	$\mathbf{40.79 \pm 0.33}$	$\mathbf{37.11 \pm 0.37}$	$\mathbf{50.38 \pm 0.37}$
Unicorn-MAML	Conv4	29.26 ± 0.30	40.58 ± 0.33	36.07 ± 0.36	51.43 ± 0.37
MSD + Unicorn-MAML	Conv4	$\mathbf{31.37 \pm 0.32}$	$\mathbf{42.59 \pm 0.33}$	$\mathbf{38.94 \pm 0.38}$	$\mathbf{54.11 \pm 0.37}$

Table 4: **5way-1shot and 5way-5shot classification accuracy** in strongly augmented few-shot classification task and 95% confidence interval on Mini-ImageNet and Tiered-ImageNet (over 2000 tasks), using ResNet-12 as the backbone.

		Mini-ImageNet		Tiered-ImageNet	
Methods	Backbone	1-Shot	5-Shot	1-Shot	5-Shot
MAML	ResNet-12	49.94 ± 0.43	73.46 ± 0.36	51.87 ± 0.48	75.11 ± 0.39
MSD + MAML	ResNet-12	$\mathbf{57.31 \pm 0.44}$	$\mathbf{78.32 \pm 0.33}$	$\mathbf{55.79 \pm 0.49}$	$\mathbf{76.49 \pm 0.39}$
Unicorn-MAML	ResNet-12	50.57 ± 0.43	73.68 ± 0.35	53.01 ± 0.49	76.08 ± 0.40
MSD + Unicorn-MAML	ResNet-12	$\mathbf{57.75 \pm 0.44}$	$\mathbf{77.25 \pm 0.33}$	$\mathbf{56.39 \pm 0.47}$	$\mathbf{78.11 \pm 0.38}$

5.2.1 Standard Few-shot Learning Problems.

The results in Table.10 demonstrate the performance of MSD and several mainstream few-shot algorithms on few-shot tasks. MSD exhibits a significant improvement over MAML in standard few-shot tasks. The results of maml are produced by(Ye & Chao, 2021), which uses more inner steps for maml to reach better performance. On Mini-ImageNet, our method achieved an increase of 0.99% in 5way-1shot and 1.44% in 5way-5shot tasks compared with maml, respectively. On Tiered-ImageNet, the improvements for 5way-1shot and 5way-5shot tasks were 2.79% and 2.50% compared with MAML, respectively. MSD shows excellent effectiveness in few-shot tasks, with better performance compared to the recent meta-learning algorithms and MAML's variants.

5.2.2 Cross domain Few-shot Learning Problems.

To explore the performance when there is a large domain gap between the meta-train set and the meta-test set, we also evaluated the performance of MSD under the cross-domain dataset setting. The results are shown in Table 2. Experimental results demonstrate that our method achieves significant outcomes across different domains. We achieved optimal performance on five datasets and second-best performance on three additional datasets. Notably, our approach demonstrated a strong lead on the Cars, EuroSAT, ISIC, and CropDisease datasets. This suggests that MSD also demonstrates strong generalization in cross-domain few-shot problems, reducing the impact of shortcut features during the fast adaptation phase.

5.2.3 Augmented Few-shot Learning Problems.

To further explore the enhancement of the model's learning capabilities initialized by MSD, we employed augmented tasks for testing. Specifically, during the meta-test phase, we augmented the support data for model fine-tuning and then classified the query data using the updated model. We report both the classification accuracy and the consistency of knowledge learned across different methods. Conv4 and ResNet12 were utilized to validate the generalization capabilities of MSD across varying scales.

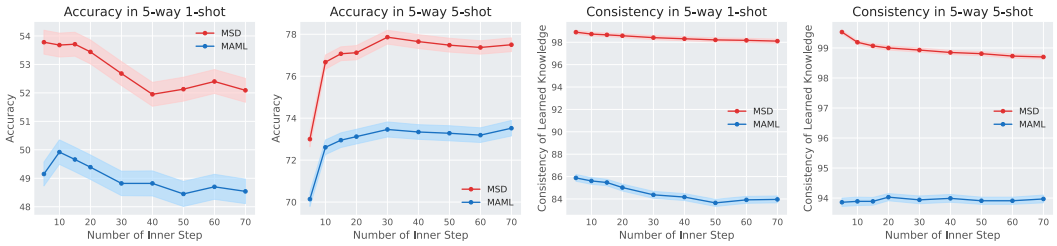

Figure 4: **The 5way-1shot and 5way-5shot classification accuracy and the consistency of learned knowledge with different numbers of inner steps** with 95% confidence interval, averaged over 2000 tasks

Table 5: **5way-1shot and 5way-5shot consistency of learned knowledge** in strong augmented few-shot classification task on Mini-ImageNet and Tiered-ImageNet (over 2000 tasks), using ResNet-12 as the backbone.

	Mini-ImageNet		Tiered-ImageNet	
Methods	1-Shot	5-Shot	1-Shot	5-Shot
MAML	85.88	94.03	84.93	93.87
MSD + MAML	**98.58**	**99.00**	**99.70**	**99.80**
Unicorn-MAML	87.55	94.60	86.67	95.41
MSD + Unicorn-MAML	**99.91**	**99.92**	**99.94**	**99.96**

Table 6: **Ablation study on Mini-ImageNet.** All models are trained on the meta-train set of Mini-ImageNet.

Aug	$\mathcal{L}_{CK}$	Mini-ImageNet 1-shot	Mini-ImageNet 5-shot
✗	✗	64.43± 0.46	83.90 ± 0.29
✓	✗	64.31± 0.48	84.14 ± 0.28
✓	✓	**65.41**± **0.47**	**84.88** ± **0.29**

Augmented few-shot accuracy. Table.3 presents the performance of Conv4 on the Mini-ImageNet dataset under varying levels of augmentation. MSD has an approximate 2% increase in classification accuracy on query data, irrespective of whether the perturbations are weak or strong. Table.4 demonstrates the performance of ResNet-12 under strong augmentation on both Mini-ImageNet and Tiered ImageNet datasets. It is evident that MSD confers greater improvements on models with larger capacities and contributes to a significant increase in accuracy for various tasks. This has further demonstrated the generalization of MSD.

Consitency of learned knowledge. Table.5 presents the consistency of knowledge acquired by the model variants for the same support data, as quantified by the similarity among the outputs of different model versions for the same query data, as shown in Eq.5. It is observed that both MAML and its variant, MAML-Unicorn, tend to learn inconsistency knowledge in both 5way-1shot and 5way-5shot scenarios. This implies that the model initialized by MAML and Unicorn-MAML is easily influenced by the different shortcut features produced by different augmentations, while our method achieves around 99% consistency in knowledge across both datasets for 5way-1shot and 5way-5shot problems. The result shows that our method significantly enhances the model's ability to learn consistently.

5.3 Ablation Study

To further explore the effectiveness of MSD, we conducted some ablation studies on MSD. We focus on the affection of data augmentation and the number of inner steps.

The impact of data augmentation and $\mathcal{L}_{CK}$ Table 5 illustrates the impact of data augmentation and $\mathcal{L}_{CK}$. The first row presents the results of MSD without data augmentation and $\mathcal{L}_{CK}$, which is equivalent to MAML. The second row shows the results of MSD without $\mathcal{L}_{CK}$, which is equivalent to MAML with augmentation. The third row displays the results of MSD. The result indicate that augmentation is not the primary factor in MSD's improvement. The main improvement is attributed to $\mathcal{L}_{CK}$, which enables the initialized model to learn consistently. This result further underscores the motivation to learn consistently.

The impact of the inner step. We further investigated the impact of different inner steps during the meta-test phase on the model's few-shot classification accuracy and precise learning capabilities.Fig.4 illustrates the impact of the number of inner steps during the meta-test phase on the

performance of the MSD algorithm. The results indicate that for any given number of inner steps, the models trained using MSD consistently outperformed those trained with MAML. Specifically, in the 5way-1shot and 5way-5shot tasks, MSD achieved an accuracy of approximately 7% and 4% higher than MAML, respectively. Concerning the consistency of the knowledge learned, there was a trend of decreasing consistency for both MAML and MSD as the number of inner steps increased. This suggests that an excessive number of inner steps during the meta-test phase may lead to the model learning shortcut features. However, MSD still maintained approximately 99% consistency in different settings of the inner step, which shows the robustness and generalization of MSD.

5.4 Further Analysis

Compute consumption. Compared to MAML, MSD achieves parity in algorithmic complexity by substituting different tasks with varied versions of the same task. Consequently, the computational overhead of MSD aligns with that of MAML.

Visualization. To further analyze the MSD on the learning capabilities of models, we visualized the models updated by augmented data as shown in Appendix Fig.5. Specifically, during the meta-test phase, we visualized models trained with Model-Agnostic Meta-Learning (MAML) and MSD. The model was first fine-tuned using augmented support data, with the number of inner steps set to 20. Then, query data was employed as the visualized data. Grad-CAM++ (Chattopadhay et al., 2018) was utilized to visualize the critical regions that the models focused on for understanding the query data. The visualizations reveal that the model trained with MAML tends to allocate more attention to the surrounding environment, potentially prioritizing it over the classified objects, while the model trained with MSD focuses more on the objects used for classification.

6 Conclusion

The tendency to learn shortcut features is the key challenge to few-shot learning. In our work, we observe that the model learned with higher consistency tends to be less influenced by the shortcut features. Building on this foundation, we introduce a meta-learning method named meta self-distillation(MSD). MSD updates the model respectively by utilizing different augmented views of support data in the inner loop, then maximizing the consistency of the outputs of the same query produced by different updated models. We evaluate MSD across three few-shot learning problems. MSD significantly enhances the performance of algorithms across various settings.

Learning to learn consistently is a new perspective for meta-learning. We believe our proposed algorithm represents a step forward in enhancing models' learning ability. Future research could extend such a framework to the domain of self-supervised learning and apply it to larger-scale models.

Reproducibility statement

The details of datasets, model architectures, hyper-parameters, and evaluation metrics are described in subsection 5.1 and Appendix A.1. Our code is attached to the Supplementary Material.

References

Zeyuan Allen-Zhu and Yuanzhi Li. Towards understanding ensemble, knowledge distillation and self-distillation in deep learning. *arXiv preprint arXiv:2012.09816*, 2020.

Antreas Antoniou, Harrison Edwards, and Amos Storkey. How to train your maml. In *International conference on learning representations*, 2018.

Shuanghao Bai, Wanqi Zhou, Zhirong Luan, Donglin Wang, and Badong Chen. Improving cross-domain few-shot classification with multilayer perceptron. In *ICASSP 2024-2024 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 5250–5254. IEEE, 2024.

Mathilde Caron, Ishan Misra, Julien Mairal, Priya Goyal, Piotr Bojanowski, and Armand Joulin. Unsupervised learning of visual features by contrasting cluster assignments. *Advances in neural information processing systems*, 33:9912–9924, 2020.

Mathilde Caron, Hugo Touvron, Ishan Misra, Hervé Jégou, Julien Mairal, Piotr Bojanowski, and Armand Joulin. Emerging properties in self-supervised vision transformers. In *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 9650–9660, 2021.

Aditya Chattopadhay, Anirban Sarkar, Prantik Howlader, and Vineeth N Balasubramanian. Grad-cam++: Generalized gradient-based visual explanations for deep convolutional networks. In *2018 IEEE winter conference on applications of computer vision (WACV)*, pp. 839–847. IEEE, 2018.

Wei-Yu Chen, Yen-Cheng Liu, Zsolt Kira, Yu-Chiang Frank Wang, and Jia-Bin Huang. A closer look at few-shot classification. *arXiv preprint arXiv:1904.04232*, 2019.

Xinlei Chen and Kaiming He. Exploring simple siamese representation learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 15750–15758, 2021.

Nikita Dvornik, Cordelia Schmid, and Julien Mairal. Selecting relevant features from a multi-domain representation for few-shot classification. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part X 16*, pp. 769–786. Springer, 2020.

Chelsea Finn, Pieter Abbeel, and Sergey Levine. Model-agnostic meta-learning for fast adaptation of deep networks. In *International conference on machine learning*, pp. 1126–1135. PMLR, 2017.

Victor Garcia and Joan Bruna. Few-shot learning with graph neural networks. *arXiv preprint arXiv:1711.04043*, 2017.

Golnaz Ghiasi, Tsung-Yi Lin, and Quoc V Le. Dropblock: A regularization method for convolutional networks. *Advances in neural information processing systems*, 31, 2018.

Jean-Bastien Grill, Florian Strub, Florent Altché, Corentin Tallec, Pierre Richemond, Elena Buchatskaya, Carl Doersch, Bernardo Avila Pires, Zhaohan Guo, Mohammad Gheshlaghi Azar, et al. Bootstrap your own latent-a new approach to self-supervised learning. *Advances in neural information processing systems*, 33:21271–21284, 2020.

Yunhui Guo, Noel C Codella, Leonid Karlinsky, James V Codella, John R Smith, Kate Saenko, Tajana Rosing, and Rogerio Feris. A broader study of cross-domain few-shot learning. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XXVII 16*, pp. 124–141. Springer, 2020.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770–778, 2016.

Yanxu Hu and Andy J Ma. Adversarial feature augmentation for cross-domain few-shot classification. In *European conference on computer vision*, pp. 20–37. Springer, 2022.

Chia-Hsiang Kao, Wei-Chen Chiu, and Pin-Yu Chen. Maml is a noisy contrastive learner in classification. *arXiv preprint arXiv:2106.15367*, 2021.

Minyoung Kim and Timothy Hospedales. A hierarchical bayesian model for few-shot meta learning. In *The Twelfth International Conference on Learning Representations*, 2024.

Brenden M Lake and Marco Baroni. Human-like systematic generalization through a meta-learning neural network. *Nature*, 623(7985):115–121, 2023.

Duong Le, Khoi Duc Nguyen, Khoi Nguyen, Quoc-Huy Tran, Rang Nguyen, and Binh-Son Hua. Poodle: Improving few-shot learning via penalizing out-of-distribution samples. *Advances in Neural Information Processing Systems*, 34:23942–23955, 2021.

Kwonjoon Lee, Subhransu Maji, Avinash Ravichandran, and Stefano Soatto. Meta-learning with differentiable convex optimization. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 10657–10665, 2019.

Jiangmeng Li, Wenwen Qiang, Changwen Zheng, Bing Su, and Hui Xiong. Metaug: Contrastive learning via meta feature augmentation. In *International Conference on Machine Learning*, pp. 12964–12978. PMLR, 2022.

Chen Liu, Yanwei Fu, Chengming Xu, Siqian Yang, Jilin Li, Chengjie Wang, and Li Zhang. Learning a few-shot embedding model with contrastive learning. In *Proceedings of the AAAI conference on artificial intelligence*, volume 35, pp. 8635–8643, 2021.

Lu Liu, William Hamilton, Guodong Long, Jing Jiang, and Hugo Larochelle. A universal representation transformer layer for few-shot image classification. *arXiv preprint arXiv:2006.11702*, 2020.

Kaifeng Lyu, Zhiyuan Li, Runzhe Wang, and Sanjeev Arora. Gradient descent on two-layer nets: Margin maximization and simplicity bias. *Advances in Neural Information Processing Systems*, 34:12978–12991, 2021.

Puneet Mangla, Nupur Kumari, Abhishek Sinha, Mayank Singh, Balaji Krishnamurthy, and Vineeth N Balasubramanian. Charting the right manifold: Manifold mixup for few-shot learning. In *Proceedings of the IEEE/CVF winter conference on applications of computer vision*, pp. 2218–2227, 2020.

Renkun Ni, Manli Shu, Hossein Souri, Micah Goldblum, and Tom Goldstein. The close relationship between contrastive learning and meta-learning. In *International conference on learning representations*, 2021.

Alex Nichol, Joshua Achiam, and John Schulman. On first-order meta-learning algorithms. *arXiv preprint arXiv:1803.02999*, 2018.

Siyuan Qiao, Chenxi Liu, Wei Shen, and Alan L Yuille. Few-shot image recognition by predicting parameters from activations. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 7229–7238, 2018.

Aniruddh Raghu, Maithra Raghu, Samy Bengio, and Oriol Vinyals. Rapid learning or feature reuse? towards understanding the effectiveness of maml. *arXiv preprint arXiv:1909.09157*, 2019.

Sachin Ravi and Hugo Larochelle. Optimization as a model for few-shot learning. In *International conference on learning representations*, 2016.

Mengye Ren, Eleni Triantafillou, Sachin Ravi, Jake Snell, Kevin Swersky, Joshua B Tenenbaum, Hugo Larochelle, and Richard S Zemel. Meta-learning for semi-supervised few-shot classification. *arXiv preprint arXiv:1803.00676*, 2018.

Andrei A Rusu, Dushyant Rao, Jakub Sygnowski, Oriol Vinyals, Razvan Pascanu, Simon Osindero, and Raia Hadsell. Meta-learning with latent embedding optimization. *arXiv preprint arXiv:1807.05960*, 2018.

Harshay Shah, Kaustav Tamuly, Aditi Raghunathan, Prateek Jain, and Praneeth Netrapalli. The pitfalls of simplicity bias in neural networks. *Advances in Neural Information Processing Systems*, 33:9573–9585, 2020.

Christian Simon, Piotr Koniusz, Richard Nock, and Mehrtash Harandi. Adaptive subspaces for few-shot learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 4136–4145, 2020.

Jake Snell, Kevin Swersky, and Richard Zemel. Prototypical networks for few-shot learning. *Advances in neural information processing systems*, 30, 2017.

Siyuan Sun and Hongyang Gao. Meta-adam: An meta-learned adaptive optimizer with momentum for few-shot learning. *Advances in Neural Information Processing Systems*, 36, 2024.

Damien Teney, Ehsan Abbasnejad, Simon Lucey, and Anton Van den Hengel. Evading the simplicity bias: Training a diverse set of models discovers solutions with superior ood generalization. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 16761–16772, 2022.

Yonglong Tian, Yue Wang, Dilip Krishnan, Joshua B Tenenbaum, and Phillip Isola. Rethinking few-shot image classification: a good embedding is all you need? In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XIV 16*, pp. 266–282. Springer, 2020.

Eleni Triantafillou, Tyler Zhu, Vincent Dumoulin, Pascal Lamblin, Utku Evci, Kelvin Xu, Ross Goroshin, Carles Gelada, Kevin Swersky, Pierre-Antoine Manzagol, et al. Meta-dataset: A dataset of datasets for learning to learn from few examples. *arXiv preprint arXiv:1903.03096*, 2019.

Hung-Yu Tseng, Hsin-Ying Lee, Jia-Bin Huang, and Ming-Hsuan Yang. Cross-domain few-shot classification via learned feature-wise transformation. *arXiv preprint arXiv:2001.08735*, 2020.

Ihsan Ullah, Dustin Carrión-Ojeda, Sergio Escalera, Isabelle Guyon, Mike Huisman, Felix Mohr, Jan N van Rijn, Haozhe Sun, Joaquin Vanschoren, and Phan Anh Vu. Meta-album: Multi-domain meta-dataset for few-shot image classification. *Advances in Neural Information Processing Systems*, 35:3232–3247, 2022.

Oriol Vinyals, Charles Blundell, Timothy Lillicrap, Daan Wierstra, et al. Matching networks for one shot learning. *Advances in neural information processing systems*, 29, 2016.

Haoqing Wang and Zhi-Hong Deng. Cross-domain few-shot classification via adversarial task augmentation. *arXiv preprint arXiv:2104.14385*, 2021.

Yaqing Wang, Quanming Yao, James T Kwok, and Lionel M Ni. Generalizing from a few examples: A survey on few-shot learning. *ACM computing surveys (csur)*, 53(3):1–34, 2020.

Han-Jia Ye and Wei-Lun Chao. How to train your maml to excel in few-shot classification. *arXiv preprint arXiv:2106.16245*, 2021.

Han-Jia Ye, Hexiang Hu, De-Chuan Zhan, and Fei Sha. Few-shot learning via embedding adaptation with set-to-set functions. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 8808–8817, 2020.

Le Yu, Xinde Li, Pengfei Zhang, Fir Dunkin, et al. Enabling few-shot learning with pid control: A layer adaptive optimizer. In *Forty-first International Conference on Machine Learning*, 2024.

Tiange Zhang, Qing Cai, Feng Gao, Lin Qi, and Junyu Dong. Exploring cross-domain few-shot classification via frequency-aware prompting. *arXiv preprint arXiv:2406.16422*, 2024.

Fei Zhou, Peng Wang, Lei Zhang, Wei Wei, and Yanning Zhang. Revisiting prototypical network for cross domain few-shot learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 20061–20070, 2023.

A APPENDIX / SUPPLEMENTAL MATERIAL

A.1 HYPERPARAMETERS AND CODE ENVIRONMENT OF EXPERIMENT

Hyperparameters.

The hyperparameters has shown in the Table.7Table.8Table.9

Calculate resources and Environment. Our experiment is conducted on NVIDIA A800 80GB PCIe and NVIDIA A100 40GB PCIe. We use Python version 3.10.14, PyTorch version 2.3.0, and CUDA toolkit 12.1 on A800 80GB, and use Python version 3.11.9, PyTorch version 2.3.0, and CUDA toolkit 11.8 on A100 40GB,

Table 7: Experimental Setup

Parameter	**Value**
task batch Size	4
inner loop learning rate	0.05
outer loop learning rate	0.001
outer data points	15
outer loop learning rate decay	1/10 every 10 epochs
coefficient γ (Eq.9)	1

Table 8: Augmentations for Strong-Augmented Few-Shot Scenario

Augmentation	**Parameters**	**Probability**
Random Resize	(scale: 0.5–1)	-
Color Jitter	(0.8, 0.8, 0.8, 0.2)	0.8
Grayscale Conversion	-	0.2
Gaussian Blur	Expectation: 0.1, Variance: 2	0.5
Random Horizontal Flip	-	0.5

Table 9: Augmentations for Weak-Augmented Few-Shot Scenario

Augmentation	**Parameters**	**Probability**
Center Crop	84×84	-
Color Jitter	(0.4, 0.4, 0.4, 0.1)	0.8
Grayscale Conversion	-	0.2
Gaussian Blur	Expectation: 0, Variance: 1	0.5
Random Horizontal Flip	-	0.5

A.2 ALGORITHM

The specific algorithm flow of meta-self distillation is shown in Algo.2

A.3 VISUALIZATION

We visualized the models updated by augmented data as shown in Appendix Fig.5. Specifically, during the meta-test phase, we visualized models trained with Model-Agnostic Meta-Learning (MAML) and MSD. The model was first fine-tuned using augmented support data, with the number of inner steps set to 20. Then, query data was employed as the visualized data. Grad-CAM++ (Chattopadhay et al., 2018) was utilized to visualize the critical regions that the models focused on for understanding the query data. The visualizations reveal that the model trained with MAML tends to allocate more attention to the surrounding environment, potentially prioritizing it over the classified objects, while the model trained with MSD focuses more on the objects used for classification.

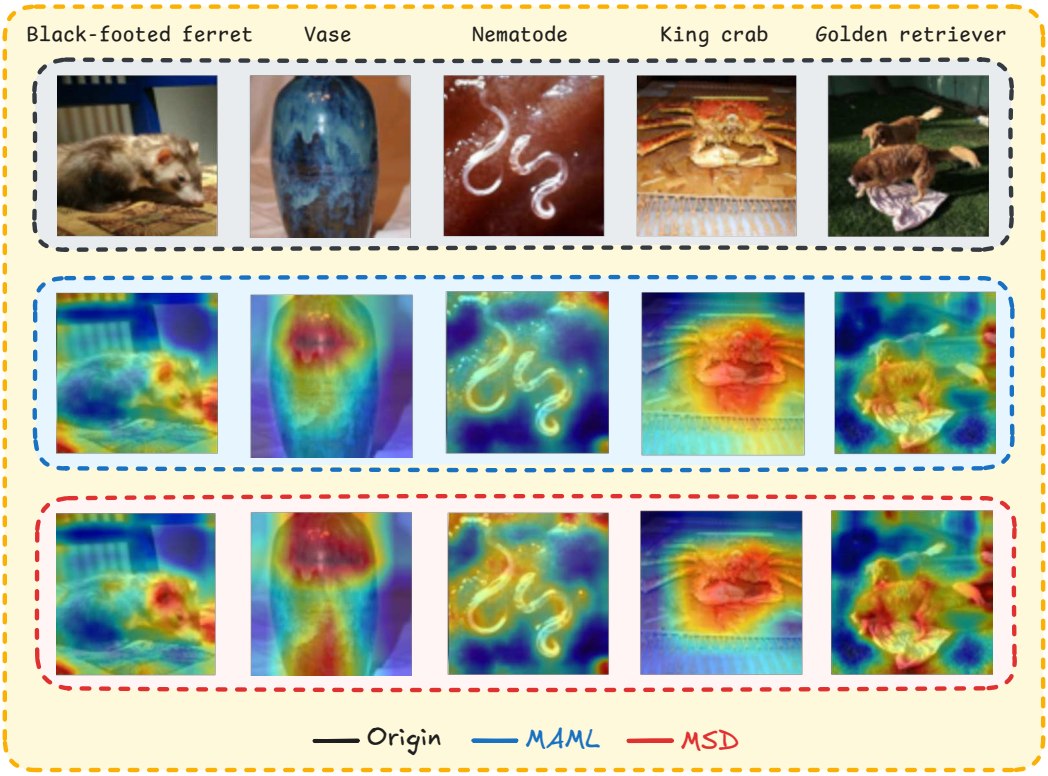

Figure 5: **The results of the visual analysis** on the test set of *Mini*ImageNet with MAML and MSD.

Algorithm 2: Meta Self-Distillation

Given the learned initialization θ^0 pretrained on meta-train set
for $t \in \{1, \cdots, T\}$ **do**
  Sample task $\mathcal{T} = (\mathcal{S}, \mathcal{Q}) \sim \mathcal{D}_{\text{meta-train}}$
  for $i \in \{1, \cdots, n\}$ **do**
    Random Augmented the support data: Get $\mathcal{S}_i = \mathcal{A}ug(\mathcal{S})$
    Update θ^{t-1} by augmented support data $\mathcal{S}_i$: Get $\theta_i^{t-1} = \mathcal{U}^k(\theta^{t-1}, \mathcal{S}_i)$
    Get the output of the query data x_q by θ_i^{t-1}: Get $v_i = f_{\theta_i^{t-1}}(x_q)$
  end
  Calculate the outer loop loss: $\mathcal{L}_{\text{total}} = \mathcal{L}_{\text{cls}} + \gamma \cdot \mathcal{L}_{\text{CK}}$
  Update the parameters by outer loop loss: $\theta^t = \theta^{t-1} - \beta \cdot \nabla_{\theta^{t-1}} \mathcal{L}_{\text{total}}$
end
Return θ^T

A.4 Compute comsumption

We counted the training time of MSD and MAML during the meta-train phase. Specifically, one epoch includes the optimization of 100 batches, where MAML uses 4 tasks for each batch for optimization, while MSD uses 1 task and enhances each batch 4 times for optimization. MSD has the same complexity as maml and thus has similar optimization times.

Table 10: **The training time of MSD and MAML during the meta-train phase**

Time(Min)	Mini-ImageNet	Tiered-ImageNet
MAML (Finn et al., 2017)	2.61	2.67
MSD(Ours)	2.76	2.85