

# Meta-Baseline: Exploring Simple Meta-Learning for Few-Shot Learning

Yinbo Chen  
UC San Diego

Zhuang Liu  
UC Berkeley

Huijuan Xu  
Penn State University

Trevor Darrell  
UC Berkeley

Xiaolong Wang  
UC San Diego

## Abstract

*Meta-learning has been the most common framework for few-shot learning in recent years. It learns the model from collections of few-shot classification tasks, which is believed to have a key advantage of making the training objective consistent with the testing objective. However, some recent works report that by training for whole-classification, i.e. classification on the whole label-set, it can get comparable or even better embedding than many meta-learning algorithms. The edge between these two lines of works has yet been underexplored, and the effectiveness of meta-learning in few-shot learning remains unclear. In this paper, we explore a simple process: meta-learning over a whole-classification pre-trained model on its evaluation metric. We observe this simple method achieves competitive performance to state-of-the-art methods on standard benchmarks. Our further analysis shed some light on understanding the trade-offs between the meta-learning objective and the whole-classification objective in few-shot learning. Our code is available at <https://github.com/yinboc/few-shot-meta-baseline>.*

## 1. Introduction

While humans have shown incredible ability to learn from very few examples and generalize to many different new examples, the current deep learning approaches still rely on a large scale of training data. To mimic this human ability of generalization, few-shot learning [3, 27] is proposed for training networks to understand a new concept based on a few labeled examples. While directly learning a large number of parameters with few samples is very challenging and most likely leads to overfitting, a practical setting is applying transfer learning: train the network on common classes (also called base classes) with sufficient samples, then transfer the model to learn novel classes with a few examples.

The meta-learning framework for few-shot learning follows the key idea of learning to learn. Specifically, it samples few-shot classification tasks from training samples be-

longing to the base classes and optimizes the model to perform well on these tasks. A task typically takes the form of  $N$ -way and  $K$ -shot, which contains  $N$  classes with  $K$  support samples and  $Q$  query samples in each class. The goal is to classify these  $N \times Q$  query samples into the  $N$  classes based on the  $N \times K$  support samples. Under this framework, the model is directly optimized on few-shot classification tasks. The consistency between the objectives of training and testing is considered as the key advantage of meta-learning. Motivated by this idea, many recent works [25, 5, 24, 28, 4, 21, 10, 30] focus on improving the meta-learning structure, and few-shot learning itself has become a common testbed for evaluating meta-learning algorithms.

However, some recent works find that training for whole-classification, i.e. classification on the whole training label-set (base classes), provides the embedding that is comparable or even better than many recent meta-learning algorithms. The effectiveness of whole-classification models has been reported in both prior works [5, 1] and some concurrent works [29, 26]. Meta-learning makes the form of training objective consistent with testing, but why it turns out to learn even worse embedding than simple whole-classification? While there are several possible reasons, e.g. optimization difficulty or overfitting, the answer has not been clearly studied yet. It remains even unclear that whether meta-learning is still effective compared to whole-classification in few-shot learning.

In this work, we aim at exploring the edge between whole-classification and meta-learning by decoupling the discrepancies. We start with Classifier-Baseline: a whole-classification method that is similarly proposed in concurrent works [29, 26]. In Classifier-Baseline, we first train a classifier on base classes, then remove the last fully-connected (FC) layer which is class-dependent. During test time, it computes mean embedding of support samples for each novel class as their centroids, and classifies query samples to the nearest centroid with cosine distance. We observe this baseline method outperforms many recent meta-learning algorithms.

In order to understand whether meta-learning is still ef-

fective compared to whole-classification, a natural experiment is to see what happens if we perform further meta-learning over a converged Classifier-Baseline on its evaluation metric (i.e. cosine nearest-centroid). As a resulting method, it is similar to MatchingNet [27] or ProtoNet [23] with an additional classification pre-training stage. We observe that meta-learning can still improve Classifier-Baseline, and it achieves competitive performance to state-of-the-art methods on standard benchmarks. We call this simple method Meta-Baseline. We highlight that as a method, all the individual components of Meta-Baseline have been proposed in prior works, but to the best of our knowledge, it has been overlooked that none of the prior works studies them as a whole. We further decouple the discrepancies by evaluating on two types of generalization: *base class generalization* denotes performance on few-shot classification tasks from unseen data in the base classes, which follows the common definition of generalization (i.e. evaluated in the training distribution); and *novel class generalization* denotes performance on few-shot classification tasks from data in novel classes, which is the goal of the few-shot learning problem. We observe that: (i) During meta-learning, improving base class generalization can lead to worse novel class generalization; (ii) When training Meta-Baseline from scratch (i.e. without whole-classification training), it achieves higher base-class generalization but much lower novel class generalization.

Our observations suggest that there could be a trade-off between the objectives of meta-learning and whole-classification. It is likely that meta-learning learns the embedding that works better for  $N$ -way  $K$ -shot tasks, while whole-classification learns the embedding with stronger class transferability. We find that the main advantage of training for whole-classification before meta-learning is likely to be improving class transferability. Our further experiments provide a potential explanation of what makes Meta-Baseline a strong baseline: by inheriting one of the most effective evaluation metrics of the whole-classification model, it maximizes the reusing of the embedding with strong class transferability. From another perspective, our results also rethink the comparison between meta-learning and whole-classification from the perspective of datasets. When base classes are collected to cover the distribution of novel classes, novel-class generalization should converge to base-class generalization and the strength of meta-learning may overwhelm the strength of whole-classification.

In summary, our contributions are as following:

- We present a simple Meta-Baseline that has been overlooked in prior work. It achieves competitive performance to state-of-the-art methods on standard benchmarks and is easy to follow.
- We observe a trade-off between the objectives of meta-

learning and whole-classification, which potentially explains the success of Meta-Baseline and rethinks the effectiveness of both objectives in few-shot learning.

## 2. Related Work

Most recent approaches for few-shot learning follow the meta-learning framework. The various meta-learning architectures for few-shot learning can be roughly categorized into three groups. *Memory-based methods* [18, 14, 22, 12, 13] are based on the idea to train a meta-learner with memory to learn novel concepts (e.g. an LSTM-based meta-learner). *Optimization-based methods* [6, 21] follows the idea of differentializing an optimization process over support-set within the meta-learning framework: MAML [4] finds an initialization of the neural network that can be adapted to any novel task using a few optimization steps. MetaOptNet [10] learns the feature representation that can generalize well for a linear support vector machine (SVM) classifier. Besides explicitly considering the dynamic learning process, *metric-based methods* [27] meta-learn a deep representation with a metric in feature space. For example, Prototypical Networks [23] compute the average feature for each class in support-set and classify query samples by the nearest-centroid method. They use Euclidean distance since it is a Bregman divergence. Relation Networks [25] further generalizes this framework by proposing a relation module as a learnable metric jointly trained with deep representations. TADAM [15] proposes to use a task conditioned metric resulting in a task-dependent metric space.

While significant progress is made in the meta-learning framework, some recent works challenge the effectiveness of meta-learning with simple whole-classification, i.e. a classification model on the whole training label-set. Cosine classifier [5] and Baseline++ [1] perform whole-classification training by replacing the top linear layer with a cosine classifier, and they adapt the classifier to a few-shot classification task of novel classes by performing nearest centroid or fine-tuning a new layer respectively. They show these whole-classification models can achieve competitive performance compared to several popular meta-learning models. Another recent work [2] studies on a transductive setting. Along with these baseline methods, more advanced meta-learning methods [24, 10, 30] are proposed and they set up new state-of-the-art results. The effectiveness of whole-classification is then revisited in two of the concurrent works [29, 26] with improved design choices.

By far, the effectiveness of meta-learning compared to whole-classification in few-shot learning is still unclear, since the edge between whole-classification models and meta-learning models remains underexplored. The goal of this work is to explore the insights behind the phenomena. Our experiments show a potential trade-off between the

Method	Whole-classification training	Meta-learning	Others
Matching Networks [27]	no / yes (large models)	attention + cosine	FCE
Prototypical Networks [23]	no	centroid + Euclidean	-
Baseline++ [1]	yes (cosine classifier)	-	fine-tuning
Meta-Baseline (ours)	yes	centroid + cosine (* $\tau$ )	-

Table 1: **Overview of method comparison.** We summarize the differences between Meta-Baseline and prior methods.

meta-learning and whole-classification objectives, which provides a more clear understanding of the comparison between both objectives for few-shot learning.

As a method, similar ideas to Classifier-Baseline are concurrently reported in recent works [29, 26]. Unlike some prior works [5, 1], Classifier-Baseline does not replace the last layer with cosine classifier during training, it trains the whole-classification model with a linear layer on the top and applies cosine nearest-centroid metric during the test time for few-shot classification on novel classes. The Meta-Baseline is meta-learning over a converged Classifier-Baseline on its evaluation metric (cosine nearest-centroid). It is similar (with inconspicuous and important differences as shown in Table 1) to those simple and classical metric-based meta-learning methods [27, 23]. The main purpose of Meta-Baseline in this paper is to understand the comparison between whole-classification and meta-learning objectives, but we find it is also a simple meta-learning baseline that has been overlooked. While every individual component in Meta-Baseline is not novel, to the best of our knowledge, none of the prior works studies them as a whole.

### 3. Method

#### 3.1. Problem definition

In standard few-shot classification, given a labeled dataset of base classes  $C_{base}$  with a large number of images, the goal is to learn concepts in novel classes  $C_{novel}$  with a few samples. In an  $N$ -way  $K$ -shot few-shot classification task, the *support-set* contains  $N$  classes with  $K$  samples per class, the *query-set* contains samples from the same  $N$  classes with  $Q$  samples per class, and the goal is to classify the  $N \times Q$  query images into  $N$  classes.

#### 3.2. Classifier-Baseline

Classifier-Baseline is a whole-classification model, i.e. a classification model trained for the whole label-set. It refers to training a classifier with classification loss on all base classes and performing few-shot tasks with the cosine nearest-centroid method. Specifically, we train a classifier on all base classes with standard cross-entropy loss, then remove its last FC layer and get the encoder  $f_\theta$ , which maps the input to embedding. Given a few-shot task with the

support-set  $S$ , let  $S_c$  denote the few-shot samples in class  $c$ , it computes the average embedding  $w_c$  as the centroid of class  $c$ :

$$w_c = \frac{1}{|S_c|} \sum_{x \in S_c} f_\theta(x), \quad (1)$$

then for a query sample  $x$  in a few-shot task, it predicts the probability that sample  $x$  belongs to class  $c$  according to the cosine similarity between the embedding of sample  $x$  and the centroid of class  $c$ :

$$p(y = c | x) = \frac{\exp(\langle f_\theta(x), w_c \rangle)}{\sum_{c'} \exp(\langle f_\theta(x), w_{c'} \rangle)}, \quad (2)$$

where  $\langle \cdot, \cdot \rangle$  denotes the cosine similarity of two vectors.

Similar methods to Classifier-Baseline have also been proposed in concurrent works [29, 26]. Compared to Baseline++ [1], the Classifier-Baseline does not use the cosine classifier for training or perform fine-tuning during testing, while it performs better on standard benchmarks. In this work, we choose Classifier-Baseline as the representative of whole-classification models for few-shot learning. For simplicity and clarity, we do not introduce additional complex techniques for this whole-classification training.

#### 3.3. Meta-Baseline

Figure 1 visualizes the Meta-Baseline. The first stage is the classification training stage, it trains a Classifier-Baseline, i.e. training a classifier on all bases classes and remove its last FC layer to get  $f_\theta$ . The second stage is the meta-learning stage, which optimizes the model on the evaluation metric of Classifier-Baseline. Specifically, given the classification-trained feature encoder  $f_\theta$ , it samples  $N$ -way  $K$ -shot tasks (with  $N \times Q$  query samples) from training samples in base classes. To compute the loss for each task, in support-set it computes the centroids of  $N$  classes defined in Equation 1, which are then used to compute the predicted probability distribution for each sample in query-set defined in Equation 2. The loss is a cross-entropy loss computed from  $p$  and the labels of the samples in the query-set. During training, each training batch can contain several tasks and the average loss is computed.

Since cosine similarity has the value range of  $[-1, 1]$ , when it is used to compute the logits, it can be helpful to

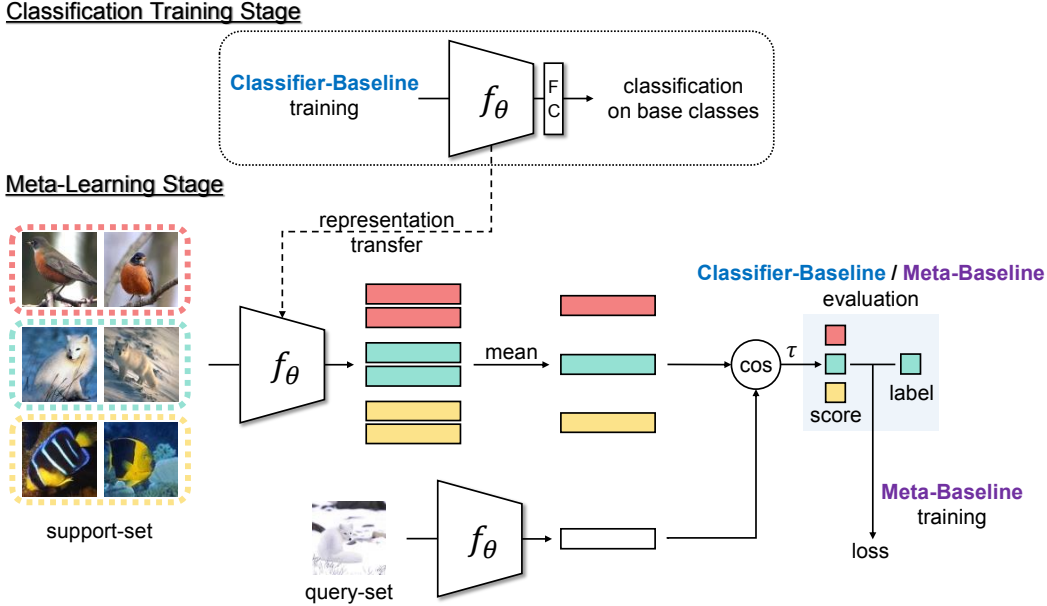


Figure 1: **Classifier-Baseline and Meta-Baseline.** Classifier-Baseline is to train a classification model on all base classes and remove its last FC layer to get the encoder  $f_\theta$ . Given a few-shot task, it computes the average feature for samples of each class in support-set, then it classifies a sample in query-set by nearest-centroid with cosine similarity as distance. In Meta-Baseline, it further optimizes a converged Classifier-Baseline on its evaluation metric, and an additional learnable scalar  $\tau$  is introduced to scale cosine similarity.

scale the value before applying Softmax function during training (a common practice in recent work [5, 16, 15]). We multiply the cosine similarity by a learnable scalar  $\tau$ , and the probability prediction in training becomes:

$$p(y = c | x) = \frac{\exp(\tau \cdot \langle f_\theta(x), w_c \rangle)}{\sum_{c'} \exp(\tau \cdot \langle f_\theta(x), w_{c'} \rangle)}. \quad (3)$$

In this work, the main purpose of Meta-Baseline is to investigate whether the meta-learning objective is still effective over a whole-classification model. As a method, while every component in Meta-Baseline has been proposed in prior works, we find none of the prior works studies them as a whole. Therefore, Meta-Baseline should also be an important baseline that has been overlooked.

## 4. Results on Standard Benchmarks

### 4.1. Datasets

The *miniImageNet* dataset [27] is a common benchmark for few-shot learning. It contains 100 classes sampled from ILSVRC-2012 [20], which are then randomly split to 64, 16, 20 classes as training, validation, and testing set respectively. Each class contains 600 images of size  $84 \times 84$ .

The *tieredImageNet* dataset [19] is another common benchmark proposed more recently with much larger scale.

It is a subset of ILSVRC-2012, containing 608 classes from 34 super-categories, which are then split into 20, 6, 8 super-categories, resulting in 351, 97, 160 classes as training, validation, testing set respectively. The image size is  $84 \times 84$ . This setting is more challenging since *base classes and novel classes come from different super-categories*.

In addition to the datasets above, we evaluate our model on *ImageNet-800*, which is derived from ILSVRC-2012 1K classes by randomly splitting 800 classes as base classes and 200 classes as novel classes. The base classes contain the images from the original training set, the novel classes contain the images from the original validation set. This larger dataset aims at making the training setting standard as the ImageNet 1K classification task [7].

### 4.2. Implementation details

We use ResNet-12 that follows the most of recent works [15, 24, 10, 30] on miniImageNet and tieredImageNet, and we use ResNet-18, ResNet-50 [7] on ImageNet-800. For the *classification training stage*, we use the SGD optimizer with momentum 0.9, the learning rate starts from 0.1 and the decay factor is 0.1. On miniImageNet, we train 100 epochs with batch size 128 on 4 GPUs, the learning rate decays at epoch 90. On tieredImageNet, we train 120 epochs with batch size 512 on 4 GPUs, the learning rate de-

Model	Backbone	1-shot	5-shot
Matching Networks [27]	ConvNet-4	43.56 ± 0.84	55.31 ± 0.73
Prototypical Networks [23]	ConvNet-4	48.70 ± 1.84	63.11 ± 0.92
Prototypical Networks (re-implement)	ResNet-12	53.81 ± 0.23	75.68 ± 0.17
Activation to Parameter [17]	WRN-28-10	59.60 ± 0.41	73.74 ± 0.19
LEO [21]	WRN-28-10	61.76 ± 0.08	77.59 ± 0.12
Baseline++ [1]	ResNet-18	51.87 ± 0.77	75.68 ± 0.63
SNAIL [12]	ResNet-12	55.71 ± 0.99	68.88 ± 0.92
AdaResNet [14]	ResNet-12	56.88 ± 0.62	71.94 ± 0.57
TADAM [15]	ResNet-12	58.50 ± 0.30	76.70 ± 0.30
MTL [24]	ResNet-12	61.20 ± 1.80	75.50 ± 0.80
MetaOptNet [10]	ResNet-12	62.64 ± 0.61	78.63 ± 0.46
SLA-AG [9]	ResNet-12	62.93 ± 0.63	79.63 ± 0.47
ProtoNets + TRAML [11]	ResNet-12	60.31 ± 0.48	77.94 ± 0.57
ConstellationNet [30]	ResNet-12	<b>64.89 ± 0.23</b>	<b>79.95 ± 0.17</b>
Classifier-Baseline (ours)	ResNet-12	58.91 ± 0.23	77.76 ± 0.17
Meta-Baseline (ours)	ResNet-12	63.17 ± 0.23	79.26 ± 0.17

Table 2: **Comparison to prior works on miniImageNet.** Average 5-way accuracy (%) with 95% confidence interval.

Model	Backbone	1-shot	5-shot
MAML [4]	ConvNet-4	51.67 ± 1.81	70.30 ± 1.75
Prototypical Networks* [23]	ConvNet-4	53.31 ± 0.89	72.69 ± 0.74
Relation Networks* [25]	ConvNet-4	54.48 ± 0.93	71.32 ± 0.78
LEO [21]	WRN-28-10	66.33 ± 0.05	81.44 ± 0.09
MetaOptNet [10]	ResNet-12	65.99 ± 0.72	81.56 ± 0.53
Classifier-Baseline (ours)	ResNet-12	68.07 ± 0.26	<b>83.74 ± 0.18</b>
Meta-Baseline (ours)	ResNet-12	<b>68.62 ± 0.27</b>	<b>83.74 ± 0.18</b>

Table 3: **Comparison to prior works on tieredImageNet.** Average 5-way accuracy (%) with 95% confidence interval.

cays at epoch 40 and 80. On ImageNet-800, we train 90 epochs with batch size 256 on 8 GPUs, the learning rate decays at epoch 30 and 60. The weight decay is 0.0005 for ResNet-12 and 0.0001 for ResNet-18 or ResNet-50. Standard data augmentation is applied, including random resized crop and horizontal flip. For *meta-learning stage*, we use the SGD optimizer with momentum 0.9. The learning rate is fixed as 0.001. The batch size is 4, i.e. each training batch contains 4 few-shot tasks to compute the average loss. The cosine scaling parameter  $\tau$  is initialized as 10.

We also apply *consistent sampling* for evaluating the performance. For the novel class split in a dataset, the sampling of testing few-shot tasks follows a deterministic order. Consistent sampling allows us to get a better model comparison with the same number of sampled tasks. In the following sections, when the confidence interval is omitted in the table, it indicates that a fixed set of 800 testing tasks are sam-

pled for estimating the performance.

### 4.3. Results

Following the standard-setting, we conduct experiments on miniImageNet and tieredImageNet, the results are shown in Table 2 and 3 respectively. To get a fair comparison to prior works, we perform model selection according to the validation set. On both datasets, we observe that the Meta-Baseline achieves competitive performance to state-of-the-art methods. We highlight that *many methods for comparison introduce more parameters and architecture designs (e.g. self-attention in [30])*, while *Meta-Baseline has the minimum parameters and the simplest design*. We also notice that the simple Classifier-Baseline can achieve competitive performance when compared to meta-learning methods, especially in 5-shot tasks. We observe that the meta-learning stage consistently improves Classifier-Baseline on

Model	Backbone	1-shot	5-shot
Classifier-Baseline (ours)	ResNet-18	83.51 $\pm$ 0.22	<b>94.82 <math>\pm</math> 0.10</b>
Meta-Baseline (ours)	ResNet-18	<b>86.39 <math>\pm</math> 0.22</b>	<b>94.82 <math>\pm</math> 0.10</b>
Classifier-Baseline (ours)	ResNet-50	86.07 $\pm$ 0.21	<b>96.14 <math>\pm</math> 0.08</b>
Meta-Baseline (ours)	ResNet-50	<b>89.70 <math>\pm</math> 0.19</b>	<b>96.14 <math>\pm</math> 0.08</b>

Table 4: **Results on ImageNet-800.** Average 5-way accuracy (%) is reported with 95% confidence interval.

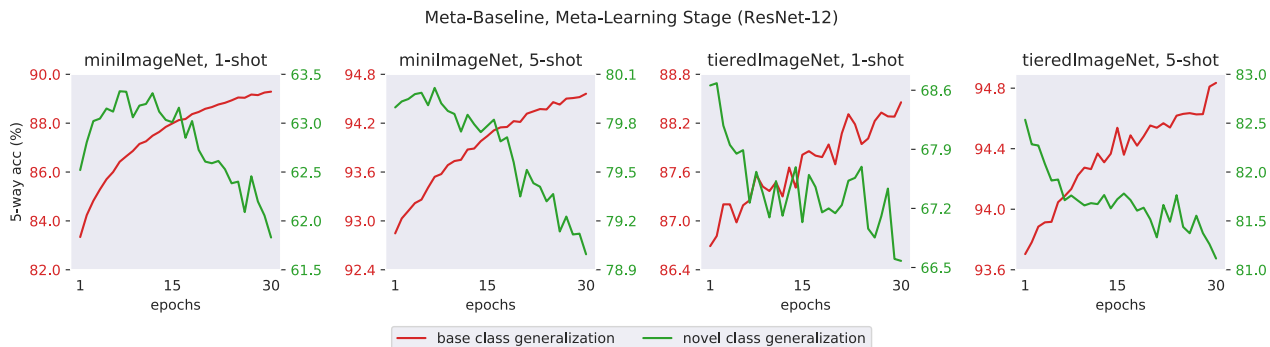


Figure 2: **Objective discrepancy of meta-learning on miniImageNet and tieredImageNet.** Each epoch contains 200 training batches. Average 5-way accuracy (%) is reported.

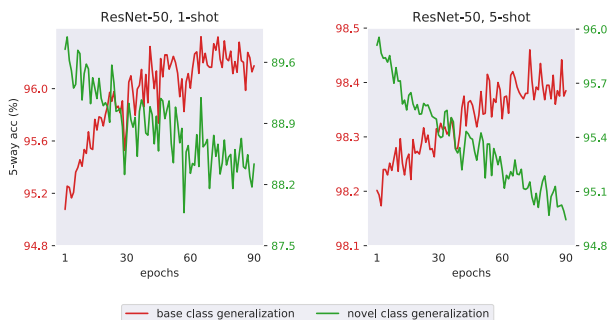


Figure 3: **Objective discrepancy of meta-learning on ImageNet-800.** Each epoch contains 500 training batches. Average 5-way accuracy (%) is reported.

miniImageNet. Compared to miniImageNet, we find that the gap between Meta-Baseline and Classifier-Baseline is smaller on tieredImageNet, and the meta-learning stage does not improve 5-shot in this case.

We further evaluate our methods on the larger dataset ImageNet-800. In this larger-scale experiment, we find freezing the Batch Normalization layer [8] (set to eval mode) is beneficial. The results are shown in Table 4. From the results, we observe that in this large dataset Meta-Baseline improves Classifier-Baseline in 1-shot, while it is not improving the performance in 5-shot.

## 5. Observations and Hypothesis

### 5.1. Objective discrepancy in meta-learning

Despite the improvements of meta-learning over Classifier-Baseline, we observe the test performance drops during the meta-learning stage. While a common assumption for this phenomenon is overfitting, we observe that this issue seems not to be mitigated on larger datasets. To further locate the issue, we propose to evaluate *base class generalization* and *novel class generalization*. Base class generalization is measured by sampling tasks from unseen images in base classes, while novel class generalization refers to the performance of few-shot tasks sampled from novel classes. The base class generalization is the generalization in the input distribution for which the model is trained, it decouples the commonly defined generalization and class-level transfer performance, which helps for locating the reason for the performance drop.

Figure 2 and 3 demonstrate the meta-learning stage of Meta-Baseline on different datasets. We find that during the meta-learning stage, when the base class generalization is increasing, the novel class generalization can be decreasing instead. This fact indicates that over a converged whole-classification model, the meta-learning objective itself, i.e. making the embedding generalize better in few-shot tasks from base classes, can have a negative effect on the performance of few-shot tasks from novel classes. It also gives a

Task	Model	mini-tiered	mini-shuffled	full-tiered	full-shuffled
1-shot	Classifier-Baseline	56.91	61.64	68.76	77.67
	Meta-Baseline	58.44	65.88	69.52	80.48
	$\Delta$	+1.53	+4.24	+0.76	+2.81
5-shot	Classifier-Baseline	74.30	79.26	84.07	90.58
	Meta-Baseline	74.63	80.58	84.07	90.67
	$\Delta$	+0.33	+1.32	+0.00	+0.09

Table 5: **Effect of dataset properties.** Average 5-way accuracy (%), with ResNet-12.

	Training	Base gen.	Novel gen.
1-shot	w/ ClsTr	86.42	<b>63.33</b>
	w/o ClsTr	<b>86.74</b>	58.54
5-shot	w/ ClsTr	93.54	<b>80.02</b>
	w/o ClsTr	<b>94.47</b>	74.95

Table 6: **Comparison on Meta-Baseline training from scratch.** Average 5-way accuracy (%), with ResNet-12 on miniImageNet. ClsTr: classification training stage.

possible explanation for why such phenomenon is not mitigated on larger datasets, as this is not sample-level overfitting, but *class-level overfitting*, which is caused by the objective discrepancy that the underlying training class distribution is different from testing class distribution.

This observation suggests that we may reconsider the motivation of the meta-learning framework for few-shot learning. In some settings, *optimizing towards the training objective with a consistent form as the testing objective (except the inevitable class difference) may have an even negative effect*. It is also likely that the whole-classification learns the embedding with stronger class transferability, and meta-learning makes the model perform better at  $N$ -way  $K$ -shot tasks but tends to lose the class transferability.

## 5.2. Effect of whole-classification training before meta-learning

According to our hypothesis, the whole-classification pre-trained model has provided extra class transferability for the meta-learning model, therefore, it is natural to compare Meta-Baseline with and without the classification training stage. The results are shown in Table 6. We observe that Meta-Baseline trained without classification training stage can actually achieve higher base class generalization, but its novel class generalization is much lower when compared to Meta-Baseline with whole-classification training.

These results support our hypothesis, that the whole-classification training provides the embedding with stronger class transferability, which significantly helps novel class generalization. Interestingly, TADAM [15] finds that co-training the meta-learning objective with a whole-

Method	1-shot	5-shot
Classifier-Baseline	60.58	79.24
Classifier-Baseline (Euc.)	56.29	78.93
Meta-Baseline	<b>63.33</b>	<b>80.02</b>
Meta-Baseline (Euc.)	60.19	79.50

Table 7: **Importance of inheriting a good metric.** Average 5-way accuracy (%), with ResNet-12 on miniImageNet.

classification task is beneficial, which may be potentially related to our hypothesis. While our results show it is likely that the key effect of the whole-classification objective is improving the class transferability, it also indicates a potential *trade-off* that the whole-classification objective can have a negative effect on base class generalization.

## 5.3. What makes Meta-Baseline a strong baseline?

As a method with a similar objective as ProtoNet [23], Meta-Baseline achieves nearly 10% higher accuracy on 1-shot in Table 2. The observations and hypothesis in previous sections potentially explain its strength, as it starts with the embedding of a whole-classification model which has stronger class transferability.

We perform further experiments, that in Meta-Baseline (with classification training stage) we replace the cosine distance with the squared Euclidean distance proposed in ProtoNet [23]. To get a fair comparison, we also include the learnable scalar  $\tau$  with a proper initialization value 0.1. The results are shown in Table 7. While ProtoNet [23] finds that squared Euclidean distance (as a Bregman divergence) works better than cosine distance when performing meta-learning from scratch, here we start meta-learning from Classifier-Baseline and we observe that cosine similarity works much better. A potential reason is that, as shown in Table 7, cosine nearest-centroid works much better than nearest-centroid with squared Euclidean distance in Classifier-Baseline (note that this is just the evaluation metric and has no changes in training). *Inheriting a good metric for Classifier-Baseline might be the key that makes Meta-Baseline strong.* According to our hypothesis, the embed-

ding from the whole-classification model has strong class transferability, inheriting a good metric potentially minimizes the future modifications on the embedding from the whole-classification model, thus it can keep the class transferability better and achieve higher performance.

#### 5.4. Effect of dataset properties

We construct four variants from the tieredImageNet dataset. Specifically, *full-tiered* refers to the original tieredImageNet, *full-shuffled* is constructed by randomly shuffling the classes in tieredImageNet and re-splitting the classes into training, validation, and test set. The *mini-tiered* and *mini-shuffled* datasets are constructed from full-tiered and full-shuffled respectively, their training set is constructed by randomly selecting 64 classes with 600 images from each class in the full training set, while the validation set and the test set remain unchanged. Since tieredImageNet separates training classes and testing classes into different super categories, shuffling these classes will mix the classes in different super categories together and make the distribution of base classes and novel classes closer.

Our previous experiments show that base class generalization is always improving, if novel classes are covered by the distribution of base classes, the novel class generalization should also keep increasing. From Table 5, we can see that from mini-tiered to mini-shuffled, and from full-tiered to full-shuffled, the improvement achieved by the meta-learning stage gets significantly larger, which consistently supports our hypothesis. Therefore, our results indicate it is likely that *meta-learning is mostly effective over whole-classification training when novel classes are similar to base classes*.

We also observe that other factors may affect the improvement of meta-learning. From mini-tiered to full-tiered and from mini-shuffled to full-shuffled, when the dataset gets larger the improvements become less. A potential hypothesis could be that the class transferability advantage of whole-classification training becomes more obvious when trained on large datasets. From the results of our experiments in Table 2, 3, 4, 5, we observe that the improvement of the meta-learning stage in 5-shot is less than 1-shot. We hypothesize this is because when there are more shots, taking average embedding becomes a more reasonable choice to estimate the class center in Classifier-Baseline, therefore the advantage of meta-learning becomes less.

#### 5.5. The trade-off between meta-learning and whole-classification

All of the experiments in previous sections support a key hypothesis, that there exists a trade-off: the meta-learning objective learns better embedding for  $N$ -way  $K$ -shot tasks (in the same distribution), while the whole-classification objective learns embedding with stronger class transferabil-

ity. Optimizing towards one objective may hurt the strength of another objective. With this hypothesis, Meta-Baseline balances this trade-off by choosing to calibrate the whole-classification embedding with meta-learning and inherit the metric with high initial performance.

The discrepancy between base class generalization and novel class generalization also considers the effectiveness of meta-learning and whole-classification from the perspective of datasets. Specifically, when novel classes are similar enough to base classes or the base classes are sufficient to cover the distribution of novel classes, novel class generalization should converge to base class generalization. In practice, this can be potentially achieved by collecting base classes that are similar to the target novel classes. In this case, it may be possible that the novel meta-learning algorithms outperform the whole-classification baselines again.

## 6. Conclusion and Discussion

In this work, we presented a simple Meta-Baseline that has been overlooked for few-shot learning. Without any additional parameters or complex design choices, it is competitive to state-of-the-art methods on standard benchmarks.

Our experiments indicate that there might be an objective discrepancy in the meta-learning framework for few-shot learning, i.e. a meta-learning model generalizing better on unseen tasks from base classes might have worse performance on tasks from novel classes. This provides a possible explanation that why some complex meta-learning methods could not get significantly better performance than simple whole-classification. While most recent works focus on improving the meta-learning structures, many of them did not explicitly address the issue of class transferability. Our observations suggest that the objective discrepancy might be a potential key challenge to tackle.

While many novel meta-learning algorithms are proposed and some recent works report that simple whole-classification training is good enough for few-shot learning, we show that meta-learning is still effective over whole-classification models. We observe a potential trade-off between the objectives of meta-learning and whole-classification. From the perspective of datasets, we demonstrate how the preference between meta-learning and whole-classification changes according to class similarity and other factors, indicating that these factors may need more attention for model comparisons in future work.

**Acknowledgements.** This work was supported, in part, by grants from DARPA LwLL, NSF 1730158 CI-New: Cognitive Hardware and Software Ecosystem Community Infrastructure (CHASE-CI), NSF ACI-1541349 CC\*DNI Pacific Research Platform, and gifts from Qualcomm, TuSimple and Picsart. Prof. Darrell was supported, in part, by DoD including DARPA's XAI, LwLL, and/or SemaFor programs, as well as BAIR's industrial alliance programs.



## References

- [1] Wei-Yu Chen, Yen-Cheng Liu, Zsolt Kira, Yu-Chiang Frank Wang, and Jia-Bin Huang. A closer look at few-shot classification. In *International Conference on Learning Representations*, 2019. 1, 2, 3, 5
- [2] Guneet S Dhillon, Pratik Chaudhari, Avinash Ravichandran, and Stefano Soatto. A baseline for few-shot image classification. *arXiv preprint arXiv:1909.02729*, 2019. 2
- [3] Li Fei-Fei, Rob Fergus, and Pietro Perona. One-shot learning of object categories. *IEEE transactions on pattern analysis and machine intelligence*, 28(4):594–611, 2006. 1
- [4] Chelsea Finn, Pieter Abbeel, and Sergey Levine. Model-agnostic meta-learning for fast adaptation of deep networks. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pages 1126–1135. JMLR. org, 2017. 1, 2, 5
- [5] Spyros Gidaris and Nikos Komodakis. Dynamic few-shot visual learning without forgetting. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4367–4375, 2018. 1, 2, 3, 4
- [6] Erin Grant, Chelsea Finn, Sergey Levine, Trevor Darrell, and Thomas Griffiths. Recasting gradient-based meta-learning as hierarchical bayes. *arXiv preprint arXiv:1801.08930*, 2018. 2
- [7] 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*, pages 770–778, 2016. 4
- [8] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. *arXiv preprint arXiv:1502.03167*, 2015. 6
- [9] Hankook Lee, Sung Ju Hwang, and Jinwoo Shin. Self-supervised label augmentation via input transformations. In *International Conference on Machine Learning*, pages 5714–5724. PMLR, 2020. 5
- [10] Kwonjoon Lee, Subhansu Maji, Avinash Ravichandran, and Stefano Soatto. Meta-learning with differentiable convex optimization. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 10657–10665, 2019. 1, 2, 4, 5
- [11] Aoxue Li, Weiran Huang, Xu Lan, Jiashi Feng, Zhenguo Li, and Liwei Wang. Boosting few-shot learning with adaptive margin loss. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12576–12584, 2020. 5
- [12] Nikhil Mishra, Mostafa Rohaninejad, Xi Chen, and Pieter Abbeel. A simple neural attentive meta-learner. In *International Conference on Learning Representations*, 2018. 2, 5
- [13] Tsendsuren Munkhdalai and Hong Yu. Meta networks. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pages 2554–2563. JMLR. org, 2017. 2
- [14] Tsendsuren Munkhdalai, Xingdi Yuan, Soroush Mehri, and Adam Trischler. Rapid adaptation with conditionally shifted neurons. *arXiv preprint arXiv:1712.09926*, 2017. 2, 5
- [15] Boris Oreshkin, Pau Rodríguez López, and Alexandre Lacoste. Tadam: Task dependent adaptive metric for improved few-shot learning. In *Advances in Neural Information Processing Systems*, pages 721–731, 2018. 2, 4, 5, 7
- [16] Hang Qi, Matthew Brown, and David G Lowe. Low-shot learning with imprinted weights. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5822–5830, 2018. 4
- [17] 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*, pages 7229–7238, 2018. 5
- [18] Sachin Ravi and Hugo Larochelle. Optimization as a model for few-shot learning. In *International Conference on Learning Representations (ICLR)*, 2017. 2
- [19] 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. 4
- [20] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, et al. Imagenet large scale visual recognition challenge. *International journal of computer vision*, 115(3):211–252, 2015. 4
- [21] Andrei A. Rusu, Dushyant Rao, Jakub Sygnowski, Oriol Vinyals, Razvan Pascanu, Simon Osindero, and Raia Hadsell. Meta-learning with latent embedding optimization. In *International Conference on Learning Representations*, 2019. 1, 2, 5
- [22] Adam Santoro, Sergey Bartunov, Matthew Botvinick, Daan Wierstra, and Timothy Lillicrap. Meta-learning with memory-augmented neural networks. In *International conference on machine learning*, pages 1842–1850, 2016. 2
- [23] Jake Snell, Kevin Swersky, and Richard Zemel. Prototypical networks for few-shot learning. In *Advances in Neural Information Processing Systems*, pages 4077–4087, 2017. 2, 3, 5, 7
- [24] Qianru Sun, Yaoyao Liu, Tat-Seng Chua, and Bernt Schiele. Meta-transfer learning for few-shot learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 403–412, 2019. 1, 2, 4, 5
- [25] Flood Sung, Yongxin Yang, Li Zhang, Tao Xiang, Philip HS Torr, and Timothy M Hospedales. Learning to compare: Relation network for few-shot learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1199–1208, 2018. 1, 2, 5
- [26] Yonglong Tian, Yue Wang, Dilip Krishnan, Joshua B Tenenbaum, and Phillip Isola. Rethinking few-shot image classification: a good embedding is all you need? *arXiv preprint arXiv:2003.11539*, 2020. 1, 2, 3
- [27] Oriol Vinyals, Charles Blundell, Timothy Lillicrap, Daan Wierstra, et al. Matching networks for one shot learning. In *Advances in neural information processing systems*, pages 3630–3638, 2016. 1, 2, 3, 4, 5
- [28] Xin Wang, Fisher Yu, Ruth Wang, Trevor Darrell, and Joseph E Gonzalez. Tafe-net: Task-aware feature embed-

- dings for low shot learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1831–1840, 2019. [1](#)
- [29] Yan Wang, Wei-Lun Chao, Kilian Q Weinberger, and Laurens van der Maaten. Simpleshot: Revisiting nearest-neighbor classification for few-shot learning. *arXiv preprint arXiv:1911.04623*, 2019. [1](#), [2](#), [3](#)
- [30] Weijian Xu, yifan xu, Huaijin Wang, and Zhuowen Tu. Constellation nets for few-shot learning. In *International Conference on Learning Representations*, 2021. [1](#), [2](#), [4](#), [5](#)