A New Meta-Baseline for Few-Shot Learning

Yinbo Chen [1]   Xiaolong Wang [2]   Zhuang Liu [2]   Huijuan Xu [2]   Trevor Darrell [2]

Abstract

Meta-learning has become a popular framework for few-shot learning in recent years, with the goal of learning a model from collections of few-shot classification tasks. While more and more novel meta-learning models are being proposed, our research has uncovered simple baselines that have been overlooked. We present a Meta-Baseline method, by pre-training a classifier on all base classes and meta-learning on a nearest-centroid based few-shot classification algorithm, it outperforms recent state-of-the-art methods by a large margin. Why does this simple method work so well? In the meta-learning stage, we observe that a model generalizing better on unseen tasks from base classes can have a decreasing performance on tasks from novel classes, indicating a potential objective discrepancy. We find both pre-training and inheriting a good few-shot classification metric from the pre-trained classifier are important for Meta-Baseline, which potentially helps the model better utilize the pre-trained representations with stronger transferability. Furthermore, we investigate when we need meta-learning in this Meta-Baseline. Our work sets up a new solid benchmark for this field and sheds light on further understanding the phenomenons in the meta-learning framework for few-shot learning.

1. Introduction

While human has shown incredible ability to learn from very few examples and then 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 (Fei-Fei et al., 2006; Vinyals et al., 2016) is proposed for training deep networks to understand a new concept based on only a few

---

[1]Tsinghua University [2]University of California, Berkeley. Correspondence to: Yinbo Chen <yinboc96@gmail.com>.

The source code is available on https://github.com/cyvius96/few-shot-meta-baseline

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. We can first train the deep model on common classes (base classes) with sufficient samples, then transfer the model to learn novel classes based on only a few examples.

Meta-learning framework for few-shot learning follows the key idea of learning to learn. Specifically, it samples few-shot learning tasks from training samples belonging 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 N classes after learning from the $N \times K$ support samples. Under the framework of meta-learning, the model is directly optimized to perform well on few-shot tasks. Motivated by this idea, recent works focus on improving the meta-learning structure, and few-shot learning itself has become a common test bed for evaluating meta-learning algorithms. While more and more meta-learning approaches (Snell et al., 2017; Sung et al., 2018; Gidaris & Komodakis, 2018; Sun et al., 2019; Wang et al., 2019; Finn et al., 2017; Rusu et al., 2019; Lee et al., 2019) are proposed for few-shot learning, very few efforts (Gidaris & Komodakis, 2018; Chen et al., 2019) have been made on improving those baseline methods.

We first adopt a simple yet effective baseline for few-shot learning, we name this baseline as Classifier-Baseline. In the Classifier-Baseline, we first pre-train a classifier on base classes for learning visual representation, and then remove the last fully-connected (FC) layer which is class-dependent. Given a novel class with a few samples, we compute the average feature of the given samples, and classify query samples by nearest-centroid using cosine distance in the feature space, namely cosine nearest-centroid. This process can also be viewed as estimating the last FC weights for novel classes, but no parameters are required to be trained for novel classes. Although similar attempts have been made in (Chen et al., 2019), they finetune a new FC layer for novel classes and follow a setting different from recent state-of-the-art methods. Instead of following them, we simply use the nearest-centroid metric and train our model with standard ImageNet-like optimization setting (He et al.,
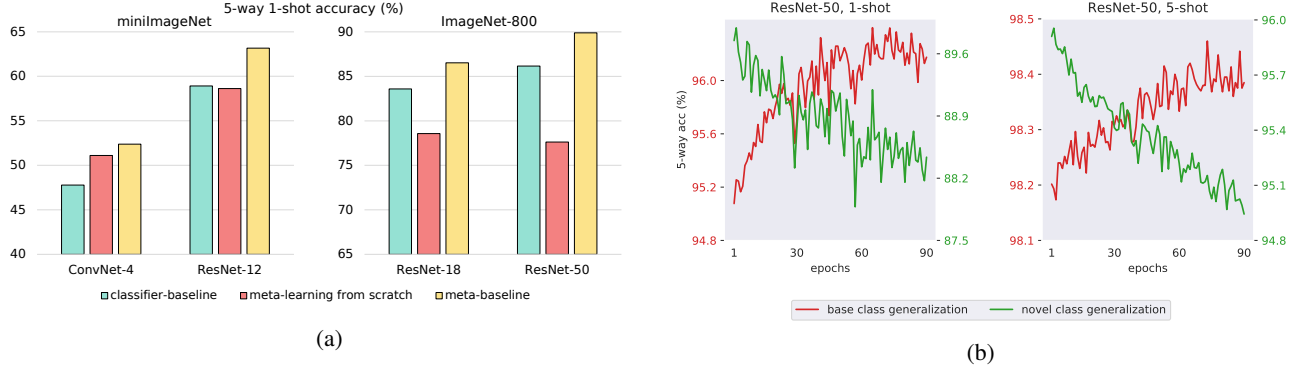
(a)

(b)

Figure 1. We present simple yet effective Classifier-Baseline and Meta-Baseline in this paper. From 1a, we observe the Meta-Baseline outperforms meta-learning from scratch especially in deep backbone and large dataset. During the meta-learning stage of Meta-Baseline shown in 1b, we observe when the model is generalizing better on unseen tasks in base classes, its few-shot classification performance in novel classes can be decreasing instead, indicating an objective discrepancy in the meta-learning framework.

2016), which leads to better model efficiency and performance. We observe this Classifier-Baseline performs on par with and even outperforms many recent state-of-the-art meta-learning methods, despite that it is neither optimizing towards few-shot tasks as meta-learning nor having additional fine-tuning.

Can we make meta-learning actually improve Classifier-Baseline? In this paper, we argue that both classification pre-training and meta-learning have their own advantages for few-shot learning, and we present Meta-Baseline which takes both strengths with the simplest form. In Meta-Baseline, we initialize the model with a pre-trained Classifier-Baseline and perform meta-learning with the cosine nearest-controid metric, which is the evaluation metric in Clssifier-Baseline. Empirically, we observe Meta-Baseline can further improve Classifier-Baseline's performance as shown in Figure 1a, and the improvement is stable across different datasets since Meta-Baseline can learn with minimum change from the pre-trained Classifier-Baseline.

Given these encouraging results, we also observe the test performance drops in training process, therefore we further perform ablative analysis in our experiments. We evaluate on two types of generalization in the context of meta-learning: (i) *base class generalization* denotes generalization on few-shot classification tasks from unseen data in the base classes, which follows the common definition of generalization but we rename it for clarification; and (ii) *novel class generalization* denotes generalization on few-shot classification tasks from data in novel classes, which further indicates the transferability of the representation from base classes to novel classes. Along different training iterations, we find that while Meta-Baseline is improving base class generalization, its novel class generalization can be decreasing instead, which indicates that: (i) There is potentially an objective discrepancy in the meta-learning stage, and it is possible that improving base class general-

ization leads to worse novel class generalization. (ii) Since the model is pre-trained before meta-learning, classification pre-training may have provided extra transferability for the meta-learning model. (iii) The advantages of meta-learning over the Classifier-Baseline should be more obvious when novel classes are more similar to base classes. In our further experiments, we get consistent observations which support these hypotheses.

In summary, our contributions are as following:

- We present a simple Meta-Baseline that takes the advantages of both classification pre-training and meta-learning, and it outperforms previous state-of-the-art methods with a large margin.

- We observe the inconsistency between *base class generalization* and *novel class generalization*, which potentially reviews the key challenge to tackle in meta-learning for few-shot classification.

- We investigate how the factors on both dataset aspect (class similarity, scale) and backbone aspect (model size) affect the advantages of meta-learning over the Classifier-Baseline.

2. Related Work

The goal of few-shot learning is adapting the classification model to new classes with only a few labelled samples. Early attempt (Fei-Fei et al., 2006) propose to utilize the knowledge learned in previous classes with a Bayesian implementation. The recent few-shot learning approaches most follow the meta-learning framework, which trains the model with few-shot tasks sampled in training data. Under this umbrella, various meta-learning architectures for few-shot learning are designed, which can be roughly cat-
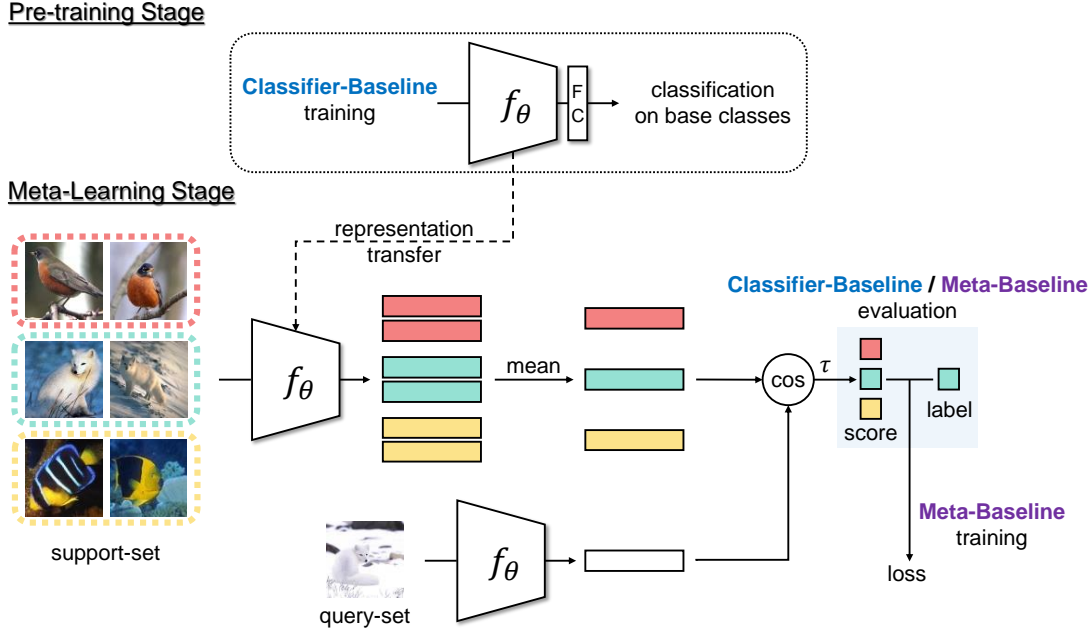
Figure 2. **Classifier-Baseline and Meta-Baseline.** In Classifier-Baseline, we pre-train a classification model on all base classes and remove its last FC layer to get the encoder f_θ. Given a few-shot task, we compute the average feature for samples of each class in support-set, then we classify a sample in query-set by nearest-centroid method with cosine similarity. In Meta-Baseline, we further optimize a pre-trained Classifier-Baseline on its evaluation algorithm, and an additional parameter τ is introduced to scale cosine similarity.

egorized into three main types: memory-based methods, optimization-based methods and metric-based methods.

Memory-based methods. The key idea of memory-based methods is to train a meta-learner with memory to learn novel concepts. (Ravi & Larochelle, 2017) proposes to learn the few-shot optimization algorithm with an LSTM-based meta-learner. (Munkhdalai et al., 2017) modifies the activation values of a network by shifting them according to the task-specific information. In (Santoro et al., 2016), a particular class of memory-augmented neural network is used for meta-learning. (Mishra et al., 2018) proposes a Simple Neural Attentive Meta-Learner using a combination of temporal convolutions and soft attention. In MetaNet (Munkhdalai & Yu, 2017), a fast parameterization approach is proposed for learning meta-level knowledge across tasks.

Optimization-based methods. Another line of work follows the idea of differentializing an optimization process over support-set within the meta-learning framework. MAML (Finn et al., 2017) finds an initialization of the neural network that can be adapted to any novel task using a few optimization steps. (Grant et al., 2018) reformulates MAML for probabilistic inference within a Bayesian framework. The LEO model (Rusu et al., 2019) follows a similar idea as MAML and overcomes the overfitting problem for high-dimensional parameter space in low data regimes by learning low dimensional latent space to perform gradient-

based meta-learning. The MetaOptNet (Lee et al., 2019) aims to learn the feature representation that can generalize well for a linear support vector machine (SVM) classifier.

Metric-based methods. Besides explicitly considering the dynamic learning process, metric-based methods meta-learn a deep representation with a metric in feature space. Matching Networks (Vinyals et al., 2016) computes attention values with cosine distance to classify query samples and proposes a Full Context Encoding module (FCE) which conditions the weights on the whole support-set across classes. Prototypical Networks (Snell et al., 2017) follows a similar idea, where they compute the average feature for each class in support-set and classify query samples by the nearest-centroid method, and they replace cosine similarity with squared Euclidean distance which is a Bregman divergence. Relation Networks (Sung et al., 2018) further generalizes this framework by proposing a relation module as a learnable metric jointly trained with deep representations. (Oreshkin et al., 2018) proposes to use a task conditioning metric resulting in a task-dependent metric space.

While significant progresses are made in the meta-learning framework, some recent work study the performance of a pre-trained classifier. (Gidaris & Komodakis, 2018) evaluates a classifier trained with cosine metric and shows that this classifier outperforms several previous meta-learning methods. (Chen et al., 2019) conducts studies on two clas-

Model	Backbone	1-shot	5-shot
Matching Networks (Vinyals et al., 2016)	ConvNet-4	43.56 ± 0.84	55.31 ± 0.73
Prototypical Networks (Snell et al., 2017)	ConvNet-4	48.70 ± 1.84	63.11 ± 0.92
Activation to Parameter (Qiao et al., 2018)	WRN-28-10	59.60 ± 0.41	73.74 ± 0.19
LEO (Rusu et al., 2019)	WRN-28-10	61.76 ± 0.08	77.59 ± 0.12
Chen *et al.* (Chen et al., 2019)	ResNet-18	51.87 ± 0.77	75.68 ± 0.63
SNAIL (Mishra et al., 2018)	ResNet-12	55.71 ± 0.99	68.88 ± 0.92
AdaResNet (Munkhdalai et al., 2017)	ResNet-12	56.88 ± 0.62	71.94 ± 0.57
TADAM (Oreshkin et al., 2018)	ResNet-12	58.50 ± 0.30	76.70 ± 0.30
MTL (Sun et al., 2019)	ResNet-12	61.20 ± 1.80	75.50 ± 0.80
MetaOptNet$^\sharp$(Lee et al., 2019)	ResNet-12	62.64 ± 0.61	78.63 ± 0.46
MetaOptNet (Lee et al., 2019)	ResNet-12	60.33 ± 0.61	76.61 ± 0.46
Classifier-Baseline (ours)	ResNet-12	58.91 ± 0.23	77.76 ± 0.17
Meta-Baseline (ours)	ResNet-12	$\mathbf{63.17 \pm 0.23}$	$\mathbf{79.26 \pm 0.17}$

Table 1. **Comparison to prior works on miniImageNet.** Average 5-way accuracy (%) is reported with 95% confidence interval. $\sharp$ refers to applying DropBlock (Ghiasi et al., 2018) and label smoothing.

sifiers which are trained with linear-classifier and cosine metric in feature space respectively, and they choose to fine-tune a new FC layer for few-shot tasks and show it can achieve competitive performance when compared to previous meta-learning models. While these works compare variants of pre-trained classifiers to several previous meta-learning methods, more advanced works (Rusu et al., 2019; Sun et al., 2019; Lee et al., 2019) are then proposed, setting up new state-of-the-art results.

The Meta-Baseline we present takes the strength of both pre-trained classifier and meta-learning with the simplest form, it outperforms previous state-of-the-art methods and demonstrates the effectiveness and potential issues of the meta-learning framework for few-shot learning.

3. Methodology

3.1. Problem definition

In the standard setting of *few-shot classification*, given a labeled dataset of base classes C_{base} with a large amount of images in each class, the goal is to learn concepts in novel classes C_{novel} with a few samples in each class, where $C_{base} \cap C_{novel} = \emptyset$. In an *N-way K-shot few-shot task*, the *support-set* contains N classes with K samples in each class, the *query-set* contains the same N classes with Q samples in each class, and the goal is to classify the $N \times Q$ unlabelled samples in query-set into N classes. During evaluation, the average accuracy is computed from many tasks sampled from the data in C_{novel}.

3.2. Classifier-Baseline

Classifier-Baseline refers to training a classifier with classification on all base classes and perform few-shot tasks with

the cosine nearest-centroid method. Specifically, we train a classifier on all base classes with standard cross-entropy loss, then we remove its last FC layer and get the encoder f_θ, which maps the input to feature space. Given a few-shot task with the support-set S, let S_c denote the few-shot samples in class c, we compute the average feature w_c as the centroid of class c:

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

Then for a query sample x in a few-shot task, we predict the probability that sample x belongs to class c as the cosine similarity between the feature vector of sample x and the centroid of class c:

$$p(y = c \mid x) = \frac{\exp\left(\langle f_\theta(x), w_c \rangle\right)}{\sum_{c'} \exp\left(\langle f_\theta(x), w_{c'} \rangle\right)}, \qquad (2)$$

where $\langle \cdot, \cdot \rangle$ denotes the cosine similarity of two vectors. Note that w_c can be also viewed as the predicted weights of the new FC layer for novel concepts.

3.3. Meta-Baseline

Figure 2 illustrates the Meta-Baseline method. In general, the Meta-Baseline contains two training stages. The first stage is *pre-training stage*, which is to train a Classifier-Baseline method (i.e. training a classifier on all bases classes and remove its last FC layer to get f_θ). The second stage is *meta-learning stage*, that we optimize the model on the evaluation algorithm of Classifier-Baseline. Specifically, given the pre-trained feature encoder f_θ, we sample N-way K-shot tasks (with $N \times Q$ query samples) from training data in base classes. To compute the loss for each task, in support-set we compute the centroids of N classes defined

Model	Backbone	1-shot	5-shot
MAML (Finn et al., 2017)	ConvNet-4	51.67 ± 1.81	70.30 ± 1.75
Prototypical Networks* (Snell et al., 2017)	ConvNet-4	53.31 ± 0.89	72.69 ± 0.74
Relation Networks* (Sung et al., 2018)	ConvNet-4	54.48 ± 0.93	71.32 ± 0.78
LEO (Rusu et al., 2019)	WRN-28-10	66.33 ± 0.05	81.44 ± 0.09
MetaOptNet (Lee et al., 2019)	ResNet-12	65.99 ± 0.72	81.56 ± 0.53
Classifier-Baseline (ours)	ResNet-12	68.07 ± 0.26	$\mathbf{83.74 \pm 0.18}$
Meta-Baseline (ours)	ResNet-12	$\mathbf{68.62 \pm 0.27}$	83.29 ± 0.18

Table 2. **Comparison to prior works on tieredImageNet.** Average 5-way accuracy (%) is reported with 95% confidence interval. * refers to results from (Lee et al., 2019)

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 query-set. Note we treat each task as a data-point in training, that each batch may contain several tasks and the average loss is computed.

Scaling the cosine similarity. Since cosine similarity has the value range of $[-1, 1]$, when it is used to compute the logits, it is important to scale the value before applying Softmax function during training. We add a learnable parameter τ similar to recent works (Gidaris & Komodakis, 2018; Qi et al., 2018; Oreshkin et al., 2018), that the predicted probability in training becomes:

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

In our experiments, we observe this designing choice is also important for improving the model performance.

4. Experiments

4.1. Datasets

The *miniImageNet* dataset (Vinyals et al., 2016) is a common benchmark for few-shot learning. It contains 100 classes sampled from ILSVRC-2012 (Russakovsky et al., 2015), which are then randomly split to 64, 16, 20 classes as training, validation, testing set respectively. Each class contains 600 images of size 84×84. We follow the release in (Lee et al., 2019).

The *tieredImageNet* dataset (Ren et al., 2018) 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 to 20, 6, 8 super-categories, resulting in 351, 97, 160 classes as training, validation, testing set respectively. The image size is 84×84. Note that this setting is more challenging since base classes and novel classes come from different super-categories.

We further propose a new dataset *ImageNet-800* to test our

methods, 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 in original training set, the novel classes contain the images in original validation set. We propose this dataset for experiments with larger scale and making the setting standard following image classification task (He et al., 2016). Note in this dataset there is no validation split, since the class difference can cause objective discrepancy as shown in our experiments, selecting the model according to validation set may introduce additional bias for model comparison.

4.2. Implementation details

We use *ResNet-12* (details in Appendix D) that follows most recent work (Oreshkin et al., 2018; Sun et al., 2019; Lee et al., 2019) on miniImageNet and tieredImageNet, and we use ResNet-18, ResNet-50 (He et al., 2016) on ImageNet-800. For *pre-training stage*, we use 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, and learning rate decays at epoch 90. On tieredImageNet, we train 120 epochs with batch size 512 on 4 GPUs, and learning rate decays at epoch 40 and 80. On ImageNet-800, we train 90 epochs with batch size 256 on 8 GPUs, and learning rate decays at epoch 30 and 60. The weight decay for ResNet-12 is 0.0005, and for ResNet-18 or ResNet-50 is 0.0001. Standard data augmentation is applied, including random resized crop and horizontal flip. For *meta-learning stage*, we use SGD optimizer with momentum 0.9, with a fixed learning rate 0.001. The batch size is 4, namely that each training batch contains 4 few-shot tasks to compute the average loss. The cosine scaling parameter τ is initialized as 10.

We also apply *consistent sampling* for evaluating the performance. For the novel classes split in a dataset, the sampling of testing few-shot tasks follows a deterministic order. The consistent sampling allows us to get a less biased model comparison with the same number of sampled tasks. In our ablation studies (i.e. except the results in Section 4.3), the same 800 testing tasks are sampled for estimating the

Model	miniImageNet		tieredImageNet	
	1-shot	5-shot	1-shot	5-shot
Classifier-Baseline	81.33 ± 0.42	88.14 ± 0.31	86.87 ± 0.37	92.28 ± 0.26
Meta-Baseline	**84.91 ± 0.41**	**90.95 ± 0.29**	**87.49 ± 0.39**	**92.41 ± 0.27**

Table 3. **Evaluation on single-class few-shot tasks.** Average ROAUC score is reported with 95% confidence interval. The backbone is ResNet-12 in all the experiments.

Model	Backbone	1-shot	5-shot
Classifier-Baseline (ours)	ResNet-18	83.51 ± 0.22	**94.82 ± 0.10**
Meta-Baseline (ours)	ResNet-18	**86.39 ± 0.22**	**94.82 ± 0.10**
Classifier-Baseline (ours)	ResNet-50	86.07 ± 0.21	**96.14 ± 0.08**
Meta-Baseline (ours)	ResNet-50	**89.70 ± 0.19**	**96.14 ± 0.08**

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

4.3. Results on standard benchmarks

Following the standard setting, we conduct experiments on miniImageNet and tieredImageNet, the results are shown in Table 1 and Table 2 respectively. To get a fair comparison to prior works, we perform model selection according to validation set. While prior works have different data augmentation strategies, we choose not to apply data augmentation in the meta-learning stage.

On both datasets, we observe the Meta-Baseline outperforms previous state-of-the-art meta-learning methods with a large-margin despite its simple design. We also notice that the simple Classifier-Baseline can already achieve competitive performance when compared to previous methods, especially in 5-shot tasks.

To further confirm the improvements come from not only getting specialized on N-way K-shot tasks, we construct *single-class K-shot tasks* for comparing Classifier-Baseline and Meta-Baseline. In this task, the support-set contains K samples from a class c, while the query-set contains two different classes c and c' with K samples in each class, the AUC (Area under ROC curve) score is computed as the performance of binary classification for the query samples. In Table 3, we observe the improvement of meta-learning still exists, indicating effectiveness of meta-learning stage for few-shot learning.

We further evaluate our methods on a large scale dataset ImageNet-800. In this large scale experiment we find freezing Batch Normalization layer (Ioffe & Szegedy, 2015) is beneficial, and the results are shown in Table 4. From the results, we observe that in this large dataset, Meta-Baseline improves Classifier-Baseline with a large margin in 1-shot, while it is not improving the performance in 5-shot.

4.4. Why the test performance drops quickly?

Despite the improvements of Meta-Baseline over Classifier-Baseline, we observe the testing performance drops quickly during meta-learning stage. While a common assumption for this phenomenon is over-fitting, we however observe that this issue seems not being mitigated on the larger dataset. 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.

Figure 3 illustrates the training process of Meta-Baseline on miniImageNet and tieredImageNet. We observe that during the meta-learning stage, while the base class generalization performance is increasing, indicating the model is learning about its objective better, the novel class generalization performance quickly starts to decrease. We also have similar observations for experiments on ImageNet-800 dataset as shown in Figure 1b. This indicates the dropping performance is likely to be caused by the objective discrepancy in meta-learning stage, namely that the meta-learning model quickly becomes specific on base classes which has negative effects on novel classes.

4.5. The importance of pre-training

From another aspect, based on our observations about the performance drop, we hypothesize that the pre-trained classifier can provide extra transferability from base classes to novel classes in the meta-learning stage. To confirm this hypothesis, we train Meta-Baseline from scratch and compare it with the one with pre-training.
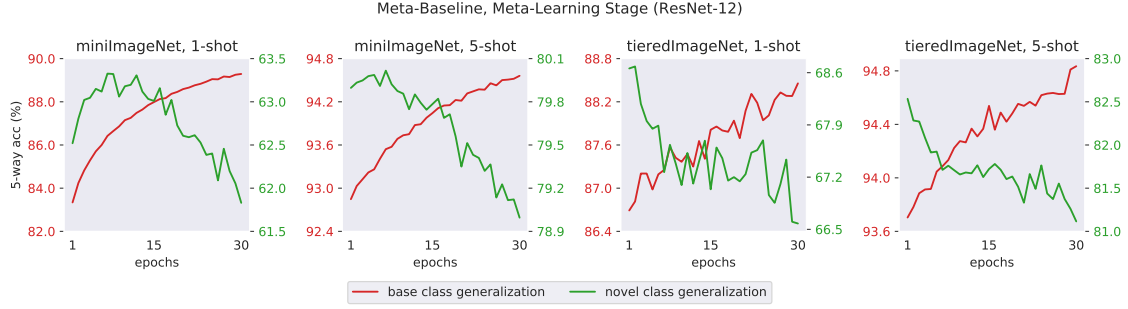
Meta-Baseline, Meta-Learning Stage (ResNet-12)

Figure 3. **Objective discrepancy in meta-learning stage.** Each epoch contains 200 training batches. Average 5-way accuracy (%) is reported.

	Training	Base gen.	Novel gen.
1-shot	Fine-tuning	86.42	**63.33**
	From scratch	**86.74**	58.54
5-shot	Fine-tuning	93.54	**80.02**
	From scratch	**94.47**	74.95

Table 5. **Effect of pre-training.** Average 5-way accuracy. With ResNet-12 on miniImageNet.

Method	1-shot	5-shot
Classifier-Baseline	60.58	79.24
Classifier-Baseline (Euclidean)	56.29	78.93
Meta-Baseline	**63.33**	**80.02**
Meta-Baseline (Euclidean)	60.19	79.50

Table 6. **Effect of inheriting a good metric.** Average 5-way accuracy (%). With ResNet-12 on miniImageNet.

On mini-imagenet, we train Meta-Baseline from scratch and choose the epoch with peak novel few-shot classification performance for both 1-shot and 5-shot, together with their corresponding base class generalization performance. We compare the chosen epochs with the training epochs selected from Meta-Baseline with pre-training, which is shown in Table 5. From the table, we observe that while Meta-Baseline trained from scratch achieves higher base class generalization, their novel class generalization is much lower when compared to Meta-Baseline with pre-training.

Interestingly, in (Oreshkin et al., 2018) they observe that co-training meta-learning method with a classification task in a controlled loss ratio is beneficial. Our results indicate that a potential important effect of classification pre-training is improving the transferability of the meta-learning model. We further observe consistent improvement of pre-training in Meta-Baseline as shown in Figure 1a.

4.6. The importance of inheriting a good metric

In addition that the pre-trained classifier is likely to provide representations with stronger transferability, here we show that inheriting a good few-shot classification metric in Classifier-Baseline is also important to Meta-Baseline.

In Prototypical Networks (Snell et al., 2017), they propose to use squared Euclidean distance as the meta-learning objective. We conduct experiments on replacing the cosine similarity with squared Euclidean distance. To get a fair compar-

ison, we also improve their method by adding the scaling parameter τ with a proper initialization value 0.1. The results are shown in Table 6. Note that Classifier-Baseline (Euclidean) refers to evaluating Classifier-Baseline with squared Euclidean distance while the pre-training remains unchanged. While in (Snell et al., 2017) they analyze that squared Euclidean distance is a Bregman divergence which is suitable for taking average, we observe cosine distance is a better choice for Meta-Baseline with pre-training. A possible reason is that the pre-trained representations are better for performing cosine similarity based few-shot classification, and when Meta-Baseline uses the metric consistent to a good one in Classifier-Baseline, it can potentially better reuse the representations in the pre-trained classifier with stronger transferability, since fewer changes are needed and transferability drops less.

4.7. When do we need meta-learning?

To conduct ablation study for answering this question, we construct four variants from the tieredImageNet dataset, an overview of them is shown in Table 8. Specifically, full-tiered refers to the orginal tieredImageNet, and full-shuffled is constructed by shuffling the classes in tieredImageNet and re-splitting the classes into training, validation and test set. Mini-tiered and mini-shuffled 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

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
	Δ	*+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
	Δ	*+0.33*	*+1.32*	*+0.00*	*+0.09*

Table 7. **Effect of dataset properties.** Average 5-way accuracy (%). Datasets are shown in Table 8. With ResNet-12 on miniImageNet.

Dataset	Scale	Novel Similarity
mini-tiered	small	low
full-tiered	large	low
mini-shuffled	small	high
full-shuffled	large	high

Table 8. **Variants constructed from tieredImageNet.** Scale refers to the size of training set, novel similarity refers to the similarity between base and novel classes.

validation and test set remain unchanged.

Class similarity. Based on our observation that base class generalization is always improving in our experiments, if novel classes are similar enough to base classes, the novel class generalization should also keep increasing. Therefore, we infer class similarity should be one of the most important factors that affect whether meta-learning improves Classifier-Baseline or not. From Table 7, we can see that from mini-tiered to mini-shuffled, and from full-tiered to full-shuffled, the improvement achieved by meta-learning stage gets significantly larger, which consistently supports our hypothesis.

Dataset scale. Another interesting phenomenon in Table 7 is that from mini-tiered to full-tiered, and from mini-shuffled to full-shuffled, when the scale of dataset gets larger, the improvements of meta-learning are significantly decreased. While the decreased improvement can partially due to that higher performance leads to less space to improve, another potential reason could be that the pre-trained classifier on larger dataset learns representations with stronger transferability while the meta-learning stage tends to break it.

Number of shots. From the results of our experiments in Table 1,2,4,7, we observe that the improvement of meta-learning stage in 5-shot is much more less than the improvement in 1-shot. We hypothesize this is because when there are more shots, taking average feature becomes a stronger choice to estimate the class center in Classifier-Baseline, therefore the advantage of meta-learning becomes less.

Backbone size. In Figure 1a, we also note that Classifier-Baseline outperforms meta-learning from scratch except with the shallow backbone ConvNet-4, which indicates a deep backbone may also change the preference of methods.

5. Additional Results on Meta-Dataset

Meta-Dataset (Triantafillou et al., 2019) is a new benchmark proposed for few-shot learning, it consists of diverse datasets for training and evaluation. They also propose to generate few-shot tasks with variable number of ways and shots, for having a setting closer to the real-world. We follow the setting in (Triantafillou et al., 2019) and use ResNet-18 as the backbone, with original image size of 126×126, which is resized to be 128×128 before feeding into the network. For pre-training stage, we apply the standard training setting similar to our setting in ImageNet-800. On the meta-learning stage, the model is trained for 5000 iterations with one task in each iteration.

The left side of Table 9 demonstrates the models trained with samples in ILSVRC-2012 only. We observe that a pre-trained Classifier-Baseline outperforms previous meta-learning method. The Meta-Baseline does not improve Classifier-Baseline under this setting in our experiments, possibly due to the average number of shots are high.

The right side of Table 9 shows the results when the models are trained on all datasets, except Traffic Signs and MSCOCO which have no training samples. The Classifier-Baseline is pre-trained as a multi-dataset classifier, i.e. an encoder together with multiple FC layers over the encoded feature to output the logits for different datasets. The pre-training stage has the same number of iterations as training on ILSVRC only, to mimic the ILSVRC training, a batch has 0.5 probability to be from ILSVRC, and 0.5 probability to be uniformly sampled from one of the other datasets. For Classifier-Baseline, comparing to the results on the left side of Table 9, we observe that while the performance on ILSVRC is worse, the performances on other datasets are mostly improved due to having their samples in training. It is also interesting to notice that the cases where Meta-Baseline improves Classifier-Baseline are mostly on the datasets which are "less similar" to ILSVRC (the dataset

Dataset	Trained on ILSVRC		Trained on all datasets		
	fo-Proto-MAML	Classifier/Meta (ours)	fo-Proto-MAML	Classifier (ours)	Meta (ours)
ILSVRC	49.5	**59.2**	46.5	**55.0**	48.0
Omniglot	63.4	**69.1**	82.7	76.9	**89.4**
Aircraft	**56.0**	54.1	75.2	69.8	**81.7**
Birds	68.7	**77.3**	69.9	**78.3**	77.3
Textures	66.5	**76.0**	68.3	**71.4**	64.5
Quick Draw	51.5	**57.3**	66.8	62.7	**74.5**
Fungi	40.0	**45.4**	42.0	55.4	**60.2**
VGG Flower	87.2	**89.6**	88.7	**90.6**	83.8
Traffic Signs	48.8	**66.2**	52.4	**69.3**	59.5
MSCOCO	43.7	**55.7**	41.7	**53.1**	43.6

Table 9. Additional results on Meta-Dataset. Average accuracy (%), with variable number of ways and shots. The fo-Proto-MAML method is from (Triantafillou et al., 2019), Classifier and Meta refers to Classifier-Baseline and Meta-Baseline respectively, 1000 tasks are sampled for evaluating Classifier or Meta. Note that Traffic Signs and MSCOCO have no training set.

"similarity" is shown in (Dvornik et al., 2020)). A potential reason is that, the multi-dataset pre-training stage samples ILSVRC with 0.5 probability, similar to ILSVRC training, the meta-learning stage is hard to improve on ILSVRC, therefore those datasets similar to ILSVRC will have similar properties so that it is hard to improve on them.

6. Conclusion

In this work, we presented a simple yet effective method for few-shot learning, namely Meta-Baseline. It outperforms recent state-of-the-art methods with a large margin, setting up a new solid benchmark for the field of few-shot classification.

While many novel meta-learning methods are proposed and some recent work argue that a pre-trained classifier is good enough for few-shot learning, our Meta-Baseline takes the strengths of both classification pre-training and meta-learning. We demonstrat that both pre-training and inheriting a good few-shot classification metric are important for Meta-Baseline to achieve strong performance, which help our model better utilize the pre-trained representations with potentially stronger class transferability.

Our experiments indicate that there might exist a potential objective discrepancy in the meta-learning framework for few-shot learning, namely a meta-learning model generalizing better on unseen tasks from base classes might have worse performance on tasks from novel classes. This probably explains why some complex meta-learning methods could not get significant better performance. While many recent works focus on improving the meta-learning structures, few of them explicitly address the issue of class transferability. Our observations suggest that the objective discrepancy

of classes might be the potential key challenge to tackle in the meta-learning framework for few-shot learning.

We also demonstrate how the preference between meta-learning and a simple pre-trained classifier depends on both dataset aspect (class similarity, scale) and backbone aspect (model size), which indicates that these factors should be noted for model comparison in future work.

References

Chen, W.-Y., Liu, Y.-C., Kira, Z., Wang, Y.-C. F., and Huang, J.-B. A closer look at few-shot classification. In *International Conference on Learning Representations*, 2019.

Dvornik, N., Schmid, C., and Mairal, J. Selecting relevant features from a universal representation for few-shot classification. *arXiv preprint arXiv:2003.09338*, 2020.

Fei-Fei, L., Fergus, R., and Perona, P. One-shot learning of object categories. *IEEE transactions on pattern analysis and machine intelligence*, 28(4):594–611, 2006.

Finn, C., Abbeel, P., and Levine, S. Model-agnostic meta-learning for fast adaptation of deep networks. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pp. 1126–1135. JMLR. org, 2017.

Ghiasi, G., Lin, T.-Y., and Le, Q. V. Dropblock: A regularization method for convolutional networks. In *Advances in Neural Information Processing Systems*, pp. 10727–10737, 2018.

Gidaris, S. and Komodakis, N. Dynamic few-shot visual learning without forgetting. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 4367–4375, 2018.

Grant, E., Finn, C., Levine, S., Darrell, T., and Griffiths, T. Recasting gradient-based meta-learning as hierarchical bayes. *arXiv preprint arXiv:1801.08930*, 2018.

He, K., Zhang, X., Ren, S., and Sun, J. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770–778, 2016.

Ioffe, S. and Szegedy, C. Batch normalization: Accelerating deep network training by reducing internal covariate shift. *arXiv preprint arXiv:1502.03167*, 2015.

Lee, K., Maji, S., Ravichandran, A., and Soatto, S. Meta-learning with differentiable convex optimization. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 10657–10665, 2019.

Mishra, N., Rohaninejad, M., Chen, X., and Abbeel, P. A simple neural attentive meta-learner. In *International Conference on Learning Representations*, 2018.

Munkhdalai, T. and Yu, H. Meta networks. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pp. 2554–2563. JMLR. org, 2017.

Munkhdalai, T., Yuan, X., Mehri, S., and Trischler, A. Rapid adaptation with conditionally shifted neurons. *arXiv preprint arXiv:1712.09926*, 2017.

Oreshkin, B., López, P. R., and Lacoste, A. Tadam: Task dependent adaptive metric for improved few-shot learning. In *Advances in Neural Information Processing Systems*, pp. 721–731, 2018.

Qi, H., Brown, M., and Lowe, D. G. Low-shot learning with imprinted weights. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 5822–5830, 2018.

Qiao, S., Liu, C., Shen, W., and Yuille, A. L. 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.

Ravi, S. and Larochelle, H. Optimization as a model for few-shot learning. In *In International Conference on Learning Representations (ICLR)*, 2017.

Ren, M., Triantafillou, E., Ravi, S., Snell, J., Swersky, K., Tenenbaum, J. B., Larochelle, H., and Zemel, R. S. Meta-learning for semi-supervised few-shot classification. *arXiv preprint arXiv:1803.00676*, 2018.

Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., Huang, Z., Karpathy, A., Khosla, A., Bernstein, M., et al. Imagenet large scale visual recognition challenge. *International journal of computer vision*, 115(3): 211–252, 2015.

Rusu, A. A., Rao, D., Sygnowski, J., Vinyals, O., Pascanu, R., Osindero, S., and Hadsell, R. Meta-learning with latent embedding optimization. In *International Conference on Learning Representations*, 2019.

Santoro, A., Bartunov, S., Botvinick, M., Wierstra, D., and Lillicrap, T. Meta-learning with memory-augmented neural networks. In *International conference on machine learning*, pp. 1842–1850, 2016.

Snell, J., Swersky, K., and Zemel, R. Prototypical networks for few-shot learning. In *Advances in Neural Information Processing Systems*, pp. 4077–4087, 2017.

Sun, Q., Liu, Y., Chua, T.-S., and Schiele, B. Meta-transfer learning for few-shot learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 403–412, 2019.

Sung, F., Yang, Y., Zhang, L., Xiang, T., Torr, P. H., and Hospedales, T. M. Learning to compare: Relation network for few-shot learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1199–1208, 2018.

Triantafillou, E., Zhu, T., Dumoulin, V., Lamblin, P., Evci, U., Xu, K., Goroshin, R., Gelada, C., Swersky, K., Manzagol, P.-A., et al. Meta-dataset: A dataset of datasets for learning to learn from few examples. *arXiv preprint arXiv:1903.03096*, 2019.

Vinyals, O., Blundell, C., Lillicrap, T., Wierstra, D., et al. Matching networks for one shot learning. In *Advances in neural information processing systems*, pp. 3630–3638, 2016.

Wang, X., Yu, F., Wang, R., Darrell, T., and Gonzalez, J. E. Tafe-net: Task-aware feature embeddings for low shot learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1831–1840, 2019.

Xu, B., Wang, N., Chen, T., and Li, M. Empirical evaluation of rectified activations in convolutional network. *arXiv preprint arXiv:1505.00853*, 2015.

A. Comparison to (Chen et al., 2019)

We connect our Classifier-Baseline to the one proposed in (Chen et al., 2019) by conducting ablation studies on mini-ImageNet, the results are shown in Table 10. We observe that fine-tuning is outperformed by simple nearest-contoid method with cosine metric, and using a standard ImageNet-like optimizer significantly improves the performance of the pre-trained classifier.

B. Relation to prior work

B.1. Comparison to prior methods

Classifier-Baseline. In training, we do not replace the last FC layer with cosine metric as in (Gidaris & Komodakis, 2018; Chen et al., 2019). In evaluation, while (Vinyals et al., 2016) uses a nearest-neighbor method, (Chen et al., 2019) fine-tunes a new FC layer, we use nearest-centroid method instead.

Pre-training. There are several recent works (Rusu et al., 2019; Qiao et al., 2018; Gidaris & Komodakis, 2018; Sun et al., 2019) also perform classification pre-training before meta-learning, while most of them freeze the pre-trained parameters and train additional parameters with the fixed pre-trained representations. Differently, Meta-Baseline does not freeze the pre-trained representations or introduce any additional parameters, it directly learns the pre-trained parameters instead, which is much simpler and arguably a better strict comparison to the pre-trained classifier (e.g. Classifier-Baseline).

Model architecture. Comparing to Matching Network (Vinyals et al., 2016) and Prototypical Network (Snell et al., 2017), Meta-Baseline computes the class centers as in (Snell et al., 2017) but different to (Vinyals et al., 2016), while it uses cosine similarity as in (Vinyals et al., 2016) but different to (Snell et al., 2017). While (Snell et al., 2017) analyses the advantage of using Euclidean distance as a Bregman divergence, we observe that inheriting a good metric for the representations of the pre-trained classifier leads to better performance. We also note the scaling factor (Gidaris & Komodakis, 2018; Qi et al., 2018; Oreshkin et al., 2018) is important for cosine similarity. In addition, Meta-Baseline does not have FCE as in (Vinyals et al., 2016) and does not train with higher few-shot classification ways as in (Snell et al., 2017).

In our experiments, we observe that the representations learned in pre-trained classifier can have stronger transferability, both pre-training and inheriting a good metric for the pre-trained classifier are important for Meta-Baseline to achieve better performance, which potentially make Meta-Baseline better utilize the pre-trained representations with fewer changes.

B.2. Relation to prior observations

In (Vinyals et al., 2016), they observe when novel classes become fine-grained comparing to base classes, meta-learning objective can not improve a pre-trained classifier. Since being fine-grained is likely requiring the model to classify classes (e.g. different dogs) which should belong to the same concept at training time, (Vinyals et al., 2016) hypothesize by sampling training classes also from fine-grained classes, the improvements could be attained. Our results demonstrate that in many cases the limited improvement is due to the objective discrepancy caused by class similarity.

In (Chen et al., 2019), they observe a baseline of learning a new FC layer outperforms meta-learning methods when domain difference gets larger (e.g. novel classes from a different dataset), while they hypothesize this is because further adaptation by fine-tuning is important for domain shift. However, we observe the advantage of the pre-trained classifier still exists even without fine-tuning, that the pre-trained representations may have stronger transferability when compared to meta-learning from scratch.

C. Comparison to cosine pre-training

We also compare the effect of pre-training with replacing the last linear-classifier with cosine nearest-neighbor metric as proposed in (Gidaris & Komodakis, 2018; Chen et al., 2019), the results are shown in Table 11, where Cosine denotes pre-training with cosine metric and Linear denotes the standard pre-training. On miniImageNet, we observe that Cosine outperforms Linear in 1-shot, but has worse performance in 5-shot. On tieredImageNet, we observe Linear outperforms Cosine in both 1-shot and 5-shot.

D. Details of ResNet-12

The ResNet-12 backbone consists of 4 residual blocks that each residual block has 3 convolutional layers. Each convolutional layer has a 3×3 kernel, followed by Batch Normalization (Ioffe & Szegedy, 2015) and Leaky ReLU (Xu et al., 2015) with 0.1 slope. The channels of convolutional layers in each residual block is 64, 128, 256, 512 respectively, a 2×2 max-pooling layer is applied after each residual block. Images are resized to 80×80 before feeding into the network, that finally a 5×5 global average pooling is applied to get a 512-dimensional feature vector.

E. Training plot on ImageNet-800

Besides miniImageNet and tieredImageNet, we also observe the objective discrepancy in the large scale dataset ImageNet-800, the base class generalization and novel class generalization performance are plotted in Figure 4. With both backbones of ResNet-18 and ResNet-50, the base class

difference	5-way 1-shot accuracy (%)
(default, reported (Chen et al., 2019))	51.75 ± 0.80
(default, re-implemented)	50.84 ± 0.80
finetune → cosine nearest-centroid	52.15 ± 0.83
epoch-300 → epoch-50	53.37 ± 0.71
remove color-jittering	56.06 ± 0.71
224×224 input size → resizing 84×84 to 224×224	50.49 ± 0.71
ResNet-18 → ResNet-12	53.59 ± 0.72
Adam (lr=0.001, batch=16) → SGD (lr=0.1, batch=128)	59.19 ± 0.71

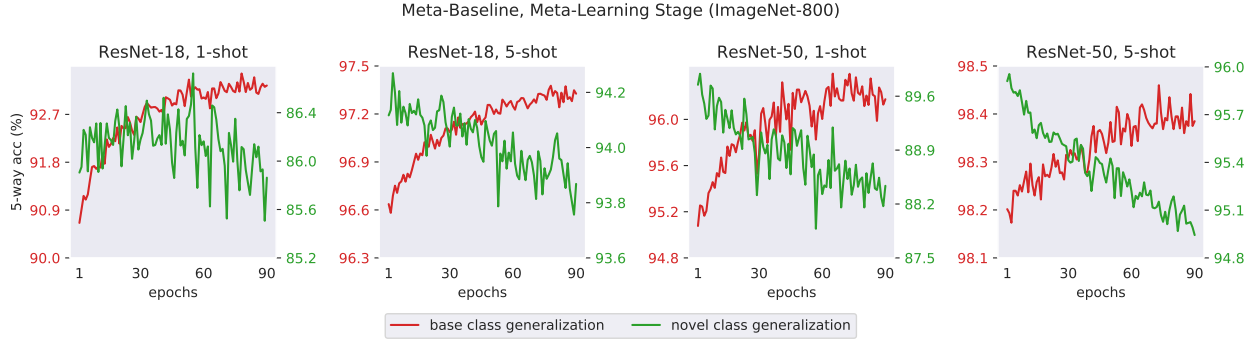*Table 10.* Comparison between Classifier-Baseline and (Chen et al., 2019).

Figure 4. **Objective discrepancy in meta-learning stage.** Average 5-way accuracy (%). Each epoch contains 500 training batches.

Dataset	Classifier	1-shot	5-shot
miniImageNet	Linear	60.58	**79.24**
	Cosine	**61.93**	78.73
tieredImageNet	Linear	**68.76**	**84.07**
	Cosine	67.58	83.31

Table 11. Comparison to classifier pre-trained with cosine metric, with backbone ResNet-12.

generalization performance is increasing during the training epochs, while the novel class generalization performance quickly starts to be decreasing.

F. Meta-Baseline training from scratch

We plot the training process of Meta-Baseline training from scratch on miniImageNet in Figure 5. We observe that when the learning rate decays, the novel class generalization quickly starts to be decreasing. While it is able to achieve higher base class generalization than Meta-Baseline with pre-training, its peak novel class generalization is still much worse, suggesting pre-training may provide representations with extra transferability.
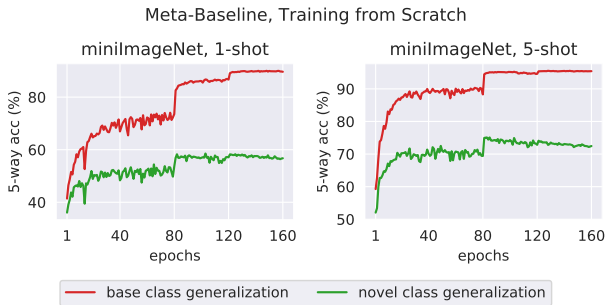

Figure 5. Meta-Baseline training from scratch on miniImageNet.