# Worst Case Matters for Few-Shot Recognition

Minghao Fu, Yun-Hao Cao, and Jianxin Wu[⋆]

State Key Laboratory for Novel Software Technology
Nanjing University, Nanjing, China
{fumh, caoyh}@lamda.nju.edu.cn, wujx2001@gmail.com

**Abstract.** Few-shot recognition learns a recognition model with very few (e.g., 1 or 5) images per category, and current few-shot learning methods focus on improving the average accuracy over many episodes. We argue that in real-world applications we may often only try one episode instead of many, and hence maximizing the worst-case accuracy is more important than maximizing the average accuracy. We empirically show that a high average accuracy not necessarily means a high worst-case accuracy. Since this objective is not accessible, we propose to reduce the standard deviation and increase the average accuracy simultaneously. In turn, we devise two strategies from the bias-variance tradeoff perspective to implicitly reach this goal: a simple yet effective stability regularization (SR) loss together with model ensemble to reduce variance during fine-tuning, and an adaptability calibration mechanism to reduce the bias. Extensive experiments on benchmark datasets demonstrate the effectiveness of the proposed strategies, which outperforms current state-of-the-art methods with a significant margin in terms of not only average, but also worst-case accuracy.

## 1 Introduction

Most people have the ability to learn to recognize new patterns via one or a few samples (e.g., images), thanks to the accumulated knowledge. Naturally, *few-shot learning* [35] aims at learning from scarce data, which is already studied long before the deep learning era. In this paper, we focus on the image recognition task, also known as *few-shot image classification*, a widely studied few-shot task [33,29,9,30,5,21,1,23,7,2,36]. Deep learning techniques have further pushed few-shot learning's *average* accuracy over multiple runs (i.e., *episodes*) towards a high level that appears to be already applicable to real-world applications.

In this task, a pretrained model is first derived from the *base set*, a large set of labeled images. Then, given some unseen categories and very few training images per category (the *novel set*), the model must learn to adapt to classifying new examples from these novel categories. Differently sampled novel sets lead to different episodes, different trained models and test accuracies. The common evaluation criterion is to run a large number of (usually 500 to 10000) episodes for the same task, and report the average accuracy and its 95% confidence interval.

[⋆] J. Wu is the corresponding author.

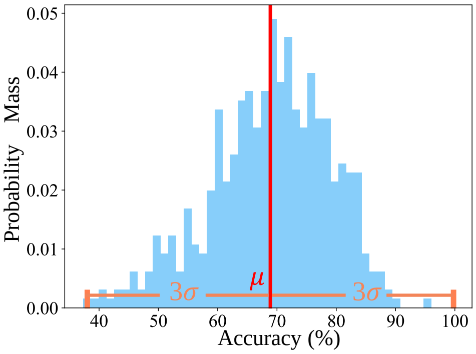


**Fig. 1.** Distribution of accuracy rates of 500 episodes in our experiments, with 5-way 1-shot experiments of the LR-DC [36] method on *mini*-ImageNet. Best viewed in color.

We aim at making few-shot recognition more practical, too. But, we argue that both metrics (mean accuracy and 95% confidence interval) are *not helping us towards reaching this goal*, given the recent progress in this task. Fig. 1 shows the distribution of accuracy of 500 episodes for the same task, whose average accuracy estimate is 68.96% and the 95% confidence interval is $[68.07, 69.85]$—this interval has 95% chance of the true average accuracy landing on it, not the chance of a single episode's accuracy dropping inside! In other words, both metrics are used to describe *the mean accuracy of 500 episodes*, *not the accuracy of a single experiment*.

In fact, the worst episode among these 500 in Fig. 1 has only 37.33% accuracy while that of the best is close to 100%. That is, few-shot learning is very *unstable*, and the accuracy varies dramatically. The *worst-case lags far behind the average*.

Furthermore, now that we aim at making few-shot learning practical, we have to accept that in most real-world applications, we can run the experiment *only once*, but do not have the luxury of running 500 (or more) experiments and pick the best or average episode among them. In other words, *the worst-case scenario* (if one is unlucky) is naturally more important than the average case—in commercial applications, we welcome a product with an acceptable worst-case performance more than one that performs well on average but occasionally goes calamitous. We argue that we need to pay attention to one episode instead of the average of 500 episodes, and to *maximize the worst case's accuracy* instead of the mean.

But, it is in general very challenging to explicitly optimize the worst-case scenario even when there are many training examples, let alone in the few-shot setting. We propose an alternative, indirect line of attack. As shown in Fig. 1, the accuracy distribution often fits well to a Gaussian. Then, the worst-case accuracy is naturally estimated by the $3\sigma$ rule as $\mu - 3\sigma$, where $\mu$ is the average accuracy and $\sigma$ is the accuracy's standard deviation—in a normal distribution, the chance to fall on the left of $\mu - 3\sigma$ is only 0.135%. Fig. 1 clearly supports this approximation. In other words, we do need to increase the mean accuracy $\mu$ (as in current methods), but also need to *simultaneously minimize the standard deviation* $\sigma$.

Unfortunately, directly optimizing $\sigma$ is not plausible, because it involves different episodes and different models, which is beyond the capability of end-to-end deep learning. To tackle this issue, we propose to resort to the bias-variance decomposition theory [14], which states that the expected error $(1 - \mu)$ is the sum of squared bias plus variance ($\sigma^2$). From this classic theory, we know that *stable* learning machines lead to small $\sigma$, and hence propose a novel *stability regularization* (SR) to utilize the base set to achieve stability. We also incorporate model ensemble, because it is well-known for its effect of reducing both bias and variance. Similarly, this theory also states that a model with larger capacity often lead to smaller bias, hence we propose an *adaptability calibration* (AC) strategy to properly increase the model capacity without overfitting, so as to indirectly reduce the bias and to increase the average accuracy $\mu$ in turn.

Our contributions can be summarized as follows:

1. To the best of our knowledge, we are the first to emphasize the importance and to advocate the adoption of worst case accuracy in few-shot learning.
2. Motivated by Fig. 1, we argue that in addition to maximizing the average accuracy $\mu$, we must also *simultaneously* reduce the standard deviation $\sigma$.
3. We propose to achieve this goal from the bias-variance tradeoff perspective. We propose a simple yet effective stability regularization (SR) loss together with model ensemble to reduce variance during fine-tuning. The SR loss is computed in an unsupervised fashion, and can be easily generalized to data beyond the base set. We also propose adaptability calibration (AC) to vary the number of learnable parameters to reduce bias.

As a result, our method not only achieves higher average accuracy than current state-of-the-art methods, but more importantly enjoys significantly higher worst case accuracy on benchmark few-shot recognition datasets.

## 2 Related Work

Generalizing from base categories, few-shot learning aims at designing effective recognition systems with limited training data from novel categories. Current few-shot learning research can be roughly divided into two branches: those based on meta learning and those based on transfer learning. We now give a brief introduction of representative methods within both branches. Besides, since we propose using the ensemble technique for better regularizing the model stability, model ensemble methods in few-shot problems will be discussed as well.

**Meta learning.** This line of methods model the training process of few-shot learning by pretending the episodes (which are usually large in quantity) are to be learned in a way of "learning to learn". One line of work [9,20,22,28,10] learns a set of good initial weights to fast adapt to unseen episodes with a limited number of gradient descent steps. Another line of work [29,30,33,4,17] leverages the characteristics of different distance metrics to classify unknown samples by comparing with embeddings or their variants (e.g., prototypes) derived from the training examples.

**Transfer learning.** These methods [5,13,3,12,8,21,1,23,7,2,16,36] take advantage of the standard transfer learning pipeline, which first pretrain a model on base classes, then revise the feature embeddings output by the pretrained model with limited novel samples. [5] used cosine classifier to normalize the magnitude of both embeddings and classification weights for compacting the intra-class intensity. [12] proposed to pretrain with self-supervised auxiliary tasks for boosting few-shot learning. [21] introduced a negative margin loss during pretraining to increase the discriminability of features. [1] used the associative alignment strategy to learn novel features by aligning them with closely related samples from the base set. [23] explored the way to pretrain with manifold mixup [32] for good generalization. [7] proposed a baseline for transductive fine-tuning with novel samples. [16] introduced a new pipeline, which first preprocessed features, then classified them with an optimal-transport inspired algorithm. [36] adjusted novel distributions using base categories to generate imaginary training samples.

**Model ensemble.** It is a well-known strategy [14] to incorporate a number of weak models to build a strong model, which will effectively increase the stability and robustness during inference. Very recently, for the few-shot classification problem, [8] harmonized the cooperation and diversity between deep models to pursue better ensemble performance. E$^3$BM [22] tackled few-shot problems with the ensemble of epoch-wise base-learners whose hyper-parameters were generated by task-specific information. Different from these methods, our ensemble stability regularization directly increases the diversity of deep models by dealing with different parts of data from the base set, keeping simplicity while achieving superior performance.

## 3 The Worst-case Accuracy and Its surrogate

We first give some background information about how a few-shot problem is defined, then describe the relationship between the commonly used 95% confidence interval and the standard deviation ($\sigma$). Lastly, our solution that advocates indirectly optimizing for the worst-case will be presented.

In the few-shot recognition setting, there exists a dataset with abundant labeled images called the base set, denoted as $D_b = \{x_i^b, y_i^b\}_{i=1}^{N_b}$, where $x_i^b \in R^D$ is the $i$-th training image, $y_i^b \in \mathcal{Y}_b$ is its corresponding category label, and $N_b$ is the number of examples. In addition, there also exists another dataset with scarce labeled images from new categories (usually 1 or 5 per category) called the novel set, denoted as $D_n = \{x_j^n, y_j^n\}_{j=1}^{N_n}$, where $y_j^n \in \mathcal{Y}_n$ and $\mathcal{Y}_b \cap \mathcal{Y}_n = \emptyset$, with $N_n$ being the number of examples it contains. In every episode, $N$ categories will be sampled to constitute the novel set, and $K$ examples will be sampled from each of those categories (called $N$-way $K$-shot). Learned with this tiny set

**Table 1.** 5-way 1-shot recognition results of existing methods on *mini*-ImageNet (500 episodes). $\flat$ means that $\sigma$ is calculated by our implementation due to undisclosed $n$ in published papers. $\text{ACC}_m$, $\text{ACC}_1$ and $\text{ACC}_{10}$: higher is better; $Z_{95\%}$ and $\sigma$: lower is better. The best results are shown in boldface.

| Method | $\text{ACC}_m$ | $Z_{95\%}$ | $\sigma$ | $\text{ACC}_1$ | $\text{ACC}_{10}$ |
|---|---|---|---|---|---|
| Negative-Cosine [21] | 61.72 | 0.81 | 10.12 | 24.27 | 36.13 |
| MixtFSL [2] | 64.31 | 0.79 | 9.87 | 30.67 | 35.07 |
| $\text{S2M2}_R$ [23] | 64.93 | **0.18** | **9.18** | 37.58 | 42.87 |
| PT+NCM [16] | 65.35 | 0.20 | 10.20 | 32.00 | 38.13 |
| CGCS [11] | 67.02 | 0.20 | 10.20 | **38.70** | **44.00** |
| LR-DC [36] | **68.57** | 0.55 | $10.28^\flat$ | 37.33 | 42.72 |

of images (and optionally $D_b$), a few-shot recognizer needs to recognize unseen examples from the $N$ categories in this episode.

### 3.1 Existing and Proposed Metrics

Existing methods evaluate their performance by averaging the accuracy of $n$ episodes, including the average accuracy $\mu$ and a 95% confidence interval $[\mu - Z_{95\%}, \mu + Z_{95\%}]$. As aforementioned, both metrics are estimates for the average accuracy random variable, not estimating the accuracy of one episode. Hence, the interval radius $Z_{95\%}$ is often surprisingly small, as shown in Table 1.

As established in Sec. 1, we need to focus more on the worst-case accuracy among all $n$ episodes, which is denoted as $\text{ACC}_1$. In addition, we also report the average accuracy of the 10 worst cases as $\text{ACC}_{10}$. The empirical average accuracy is denoted as $\text{ACC}_m$. Although it is a general trend that $\text{ACC}_m$ and $\text{ACC}_1$ are positively correlated, the 6 methods rank significantly differently using $\text{ACC}_m$ and $\text{ACC}_1$. For example, $\text{S2M2}_R$ is 3.64% lower than LR-DC in terms of $\mu$ ($\text{ACC}_m$), but 0.25% higher in $\text{ACC}_1$ (worst-case accuracy). That is, although maximizing $\mu$ is useful, it is far from being enough. We argue that for few-shot recognition to be practically usable, we need to maximize $\text{ACC}_1$ instead.

The $Z_{95\%}$ metric is also misleading, as it measures uncertainty in estimating $\mu$. In fact, based on its definition, we have

$$\sigma = Z_{95\%} \cdot \frac{\sqrt{n}}{1.96}\,, \tag{1}$$

where $\sigma$ is the standard deviation of accuracy across episodes. Hence, a small $Z_{95\%}$ may well be because $n$ is large (e.g., $n = 10000$), instead of due to a small $\sigma$, which is illustrated clearly in Table 1. For example, MixtFSL has almost 3x larger $Z_{95\%}$ than that of CGCS, but has a smaller $\sigma$ within the pair. And different papers often use different $n$ values, which renders $Z_{95\%}$ difficult to interpret.

Furthermore, the proposed worst-case accuracy $\text{ACC}_1$ is not only semantically more meaningful than $\text{ACC}_m$ ($\mu$), but also more stable. We use $\text{ACC}_k$ to denote the average accuracy of the $k$ worst episodes. Results in Table 1 exhibits that $\text{ACC}_1$ and $\text{ACC}_{10}$ ranks the methods consistently.

### 3.2 Implicitly Optimizing the Worst-Case Accuracy

As briefly introduced in Sec. 1, we can only implicitly maximize $\mathrm{ACC}_1$, and we propose to use $\mu - 3\sigma$ as a surrogate. As empirically shown in Fig. 1, when the accuracy distributes as a Gaussian, $\mu - 3\sigma$ is a perfect surrogate, as it pins to the 0.135-th percentile of $\mathrm{ACC}_1$, because $\Phi(-3) = 0.00135$ in which $\Phi$ is the cumulative distribution function of the standard normal distribution $N(0, 1)$. Even if the accuracy distribution is highly non-normal (which empirical data suggests otherwise), the one-sided Chebyshev inequality also guarantees that $\mu - 3\sigma$ is no larger than the 10th percentile (i.e., no better than the 10% worst cases), because the inequality states that $\Pr(X \leq \mu - k\sigma) \leq \frac{1}{1+k^2}$ for any distribution $X$ and any $k > 0$, while $k = 3$ in our case.

But, this surrogate loss is still a qualitative one instead of a variable that can be directly maximized, because $\sigma$ non-linearly involves all $n$ episodes, including the $n$ models and the $n$ training sets. Hence, we transform our objective to *simultaneously maximize $\mu$ and minimize $\sigma$*, and propose to use the bias-variance tradeoff for achieving both objectives indirectly.

### 3.3 The Bias-Variance Tradeoff in the Few-Shot Scenario

The bias-variance decomposition states that the expectation of error rate (i.e., $1 - \mu$) equals the sum of the squared bias and the variance [14].[1] Although the definitions for bias and variance of a classifier are not unique, the following qualitative properties are commonly agreed upon [14]:

1. A classifier has large variance if it is unstable (small changes in the input cause large changes in the prediction), and vice versa; Hence, we expect *a smaller $\sigma$ (equivalently, variance) if we make the recognizer more stable.*
2. A classifier with larger capacity (e.g., more learnable weights) in general has a smaller bias, and vice versa. Hence, we prefer *a model with larger capacity* to reduce the bias.
3. Although minimizing both bias and variance simultaneously amounts to maximizing the expected accuracy, the two terms are often contradictory to each other. Larger models are more prone to overfitting and in turn larger variance, too. Hence, *a properly calibrated capacity increase* is necessary to strike a balance.

### 3.4 Reducing Variance: Stability Regularization

To fulfill these goals, our framework (cf. Fig. 2) follows the common practice to train a backbone network $f(\mathbf{x})$ (its classification head discarded) using the base set $D_b$. Then, in one episode, a $N$-way $K$-shot training set is sampled as the novel set $D_n$ to fine-tune the backbone (with a randomly initialized classification head $W$) into $\hat{f}(\mathbf{x})$ using a usual cross entropy classification loss $\mathcal{L}_C$.

[1] Here $\mu$ is the population mean, but we also use the same notation for sample mean. They can be easily distinguished by the context.

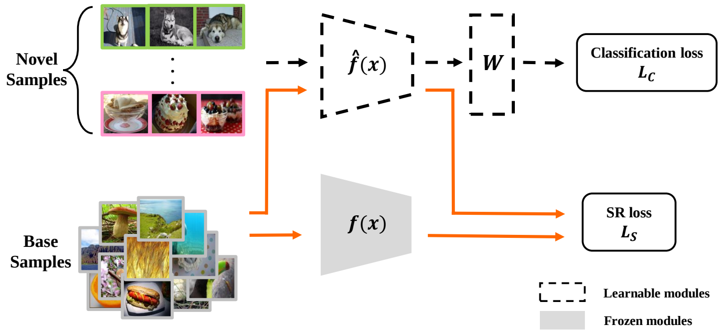


**Fig. 2.** The proposed few-shot recognition framework. A backbone network $f(\mathbf{x})$ is trained using the base set samples and its classification head is discarded. Using images from the novel set, the backbone is fine-tuned into $\hat{f}(\mathbf{x})$, plus a new classification head $W$. Stability of $\hat{f}(\mathbf{x})$ is maintained by the stability regularization (SR) loss $\mathcal{L}_S$, while the capacity increase (adaptability) is calibrated by fine-tuning a selected subset of layers in the backbone $\hat{f}(\mathbf{x})$. The classification loss $\mathcal{L}_C$ minimizes error (i.e., to increase the average accuracy $\mu$). Best viewed in color.

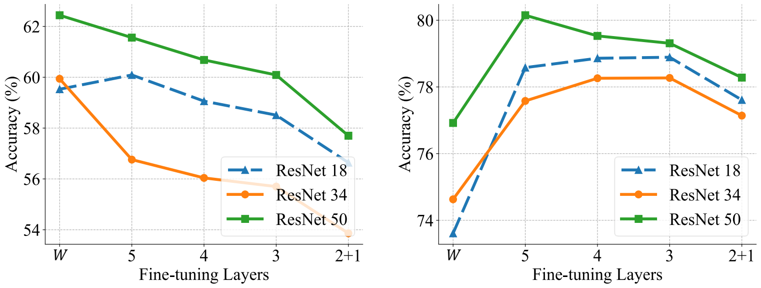
The danger is: because $K$ is very small (mostly 1 or 5), the fine-tuning process easily gets overfit and is extremely *unstable*—different sampled training sets lead to dramatically different prediction accuracy. To increase the stability (and thus to reduce the variance $\sigma^2$), we must not allow the weights in $\hat{f}(\mathbf{x})$ to be dominated by the small novel training set and completely forget the representation learned from the base set, i.e., the knowledge in $f(\mathbf{x})$.

Because the base classification head has been discarded, the stability is regularized by requiring the original learned representation ($f(\mathbf{x})$ for an input $\mathbf{x}$) and the fine-tuned representation ($\hat{f}(\mathbf{x})$ for the same input) to remain similar to each other, or at least not to deviate dramatically. A simple negative cosine similarity loss function realizes this stability requirement:

$$\mathcal{L}_S(\mathbf{x}) = -\frac{f(\mathbf{x}) \cdot \hat{f}(\mathbf{x})}{\|f(\mathbf{x})\| \|\hat{f}(\mathbf{x})\|} \,. \tag{2}$$

In order to avoid overfitting, the $\mathbf{x}$ in our stability regularization (SR) loss is not sampled from the novel set. In each mini-batch, we randomly sample 256 images from the base set with replacement to calculate $\mathcal{L}_S$ and back-propagate to $\hat{f}(\mathbf{x})$ (but $f(\mathbf{x})$ is frozen). The proposed SR loss is minimized if $\hat{f}(\mathbf{x})$ and $f(\mathbf{x})$ are the same (modulo a scale factor), which makes $\hat{f}(\mathbf{x})$ produce similar representations to $f(\mathbf{x})$ for *all* base set images, hence we can expect that it will be stable given different training sets sampled from the novel split.

It is worth noting that the proposed SR loss is very flexible since only unlabeled images are required to compute the loss. Hence, it can be easily extended to use other unlabeled images. Results on using images from other than the base or novel set are presented in Sec. 4.4.



(a) 1-shot (b) 5-shot

**Fig. 3.** Results of 5-way accuracy on *mini*-ImageNet w.r.t. the number of learnable blocks using the ResNet backbone [15]. The left-most column means only $W$ is learnable. The column '5' means $W$ plus the 'res5' group in ResNet are learnable, etc. The whole backbone together with $W$ are updated at the right-most column '2+1'.

The other loss function $\mathcal{L}_C$ in Fig. 2 is a regular cross entropy loss, which aims at maximizing the average accuracy $\mu$. Hence, the overall objective is $\mathcal{L} = \mathcal{L}_C + \alpha \cdot \mathcal{L}_S$, where $\alpha$ is always 0.1 in all experiments.

### 3.5 Reducing Bias: Adaptability Calibration

As aforementioned, to reduce the bias, we adjust the capacity of our model. In many few-shot learning methods [21,2,36,29,16], either the features extracted by the backbone $f(\mathbf{x})$ are directly used, or the backbone is freezed during fine-tuning. In other words, $\hat{f}(\mathbf{x}) \equiv f(\mathbf{x})$ and the capacity of the model is only determined by the linear classification head $W$ (or its alike), which has very low capacity, and in turn leads to high bias according to the bias-variance tradeoff. Other methods [7] fine-tune the entire backbone network, i.e., all parameters in the backbone $f(\mathbf{x})$ are learnable and $\hat{f}(\mathbf{x})$ is completely different from $f(\mathbf{x})$. Such a high capacity model inevitably leads to overfitting and ultra high variance, despite having a small bias. As aforementioned, we need to strike a good balance.

We propose a simple remedy called adaptability calibration (AC), which freezes *part of the layers* in the backbone $f(\mathbf{x})$ and fine-tune the rest layers to form $\hat{f}(\mathbf{x})$. The less layers frozen, the higher is the model's adaptability. The calibration is experimentally determined.

Backbone models in the ResNet [15] family have 5 groups of residual blocks, denoted as 'res1' (close to the input) till 'res5' (close to the classification head $W$). In Fig. 3, from left to right we gradually make 'res5' to 'res1' learnable. In the 5-shot scenario, allowing both $W$ and 'res5' updatable in fact consistently strikes the best tradeoff between bias and variance. However, in the difficult 1-shot case, only learning $W$ seems to be the best while $W$+'res5' is the runner-up. Hence, considering both cases, our adaptability calibration chooses to update both $W$ and 'res5' (the column '5' in Fig. 3).

### 3.6 Reducing both Variance and Bias: Model Ensemble

It is well-known that model ensemble is an effective way in reducing both the variance and the bias of a classifier system [38], and hence it is potentially useful for our task. Diversity among models in the ensemble (i.e., models need to predict differently for the same input) is crucial to the success of ensemble learning, while diversity is often obtained by sampling different training sets (or example weights) for different models on the *same* task (e.g., in one episode of few-shot learning).

However, the sampled novel sets for fine-tuning few-shot recognizers are so small that it leaves almost no space for sampling training sets once more for ensemble learning (e.g., consider a 1-shot task). With the proposed stability regularization and its inclusion of the base set $D_b$, ensemble learning becomes possible and natural.

We randomly divide the base set $D_b$ into $M = 4$ disjoint subsets, $D_b^m$ ($m = 1, 2, 3, 4$). Then we train $M$ few-shot recognizers for *one* episode, each trained with the sampled novel set and stability regularized by one $D_b^m$. During inference, the classification probabilities output by these $M$ models are averaged to generate recognition results of the ensemble model.

## 4 Experiments

### 4.1 Implementation Details

**Datasets.** We evaluate the proposed method on three benchmark datasets, which are *mini*-ImageNet [33], CUB-200-2011 (CUB) [34] and CIFAR-FS [4]. *mini*-ImageNet consists of 100 categories randomly selected from the ImageNet dataset [27] with each category containing 600 images sized 84×84. Following [26], it is further split to 64 base, 16 validation and 20 novel categories. CUB is composed of 200 classes (11,788 images) with size of 84×84, which is split into 100 base, 50 validation and 50 novel categories. CIFAR-FS is produced by arbitrarily dividing CIFAR-100 [19] into 64 base, 16 validation and 20 novel classes. There are 600 images with size of 32×32 pixels within each class.

**Training Details.** To validate the effectiveness of the proposed method, we take WRN-28-10 [37] or models in the ResNet series [15] as the backbone $f(\mathbf{x})$, and the cosine classifier used in [5] as the classification head $W$. We follow [23] to conduct pretraining. At the fine-tuning stage, we train $\hat{f}(\mathbf{x})$ and $W$ by adapting the pretrained model $f(\mathbf{x})$ to novel categories. For each few-shot episode, the sampled novel set is used to fine-tune the model for 100 epochs with the label smoothed cross-entropy loss [31] ($\epsilon$=0.1), by using SGD with learning rate 0.1, weight decay 1e-4 and momentum 0.9.

**Evaluation Protocols.** We conduct evaluation in the form of $N$-way $K$-shot $Q$-query, where firstly $N$ categories from the novel split of different datasets are selected arbitrarily, then mutually exclusive $K$ and $Q$ samples of each category are further chosen in the same manner, respectively. The $N \times K$ samples

**Table 2.** 5-way accuracy (%) on *mini*-ImageNet, CUB and CIFAR-FS with WRN-28-10 as the backbone. $\mathrm{ACC}_m$ is the average accuracy copied from published papers if not otherwise noted.

| Dataset | Method | **1-shot** | | | | **5-shot** | | | |
|---|---|---|---|---|---|---|---|---|---|
| | | $\mathrm{ACC}_m$ | $\mathrm{ACC}_1$ | $\mathrm{ACC}_{10}$ | $\mathrm{ACC}_{100}$ | $\mathrm{ACC}_m$ | $\mathrm{ACC}_1$ | $\mathrm{ACC}_{10}$ | $\mathrm{ACC}_{100}$ |
| *mini*-ImageNet | ProtoNet$^\dagger$ [29] | 54.16 | 19.76 | 26.08 | 37.62 | 73.68 | 43.74 | 49.78 | 59.46 |
| | Negative-Cosine [21] | 61.72 | 24.27 | 36.13 | 46.92 | 81.79 | 53.30 | 58.12 | 68.86 |
| | MixtFSL [2] | 64.31 | 30.67 | 35.07 | 46.68 | 81.66 | 46.67 | 60.13 | 71.23 |
| | $\mathrm{S2M2}_R$ [23] | 64.93 | 37.58 | 42.87 | 53.40 | 83.18 | 58.66 | 66.21 | 74.73 |
| | PT+NCM [16] | 65.35 | 32.00 | 38.13 | 48.41 | 83.87 | 56.00 | 64.00 | 73.89 |
| | Transductive-FT [7] | 65.73 | 24.00 | 33.73 | 47.60 | 78.40 | 50.67 | 53.73 | 63.09 |
| | CGCS [11] | 67.02 | 38.70 | 44.00 | 53.50 | 82.32 | 49.30 | 56.30 | 67.30 |
| | LR-DC [36] | 68.57 | 37.33 | 42.72 | 53.54 | 82.88 | 60.52 | 64.98 | 74.24 |
| | AC+SR (ours) | 69.38 | 40.52 | 44.51 | 54.97 | 85.87 | 63.20 | 66.51 | 76.28 |
| | AC+EnSR (ours) | **69.59** | **40.52** | **44.94** | **55.32** | **85.97** | **63.48** | **66.74** | **76.40** |
| CUB | Negative-Cosine [21] | 72.66 | 36.00 | 42.20 | 55.22 | 89.40 | 70.70 | 72.69 | 79.51 |
| | ProtoNet$^\dagger$ [29] | 72.99 | 28.00 | 35.14 | 47.90 | 86.64 | 53.33 | 60.65 | 70.20 |
| | MixtFSL [2] | 73.94 | 40.00 | 44.93 | 55.76 | 86.01 | 57.33 | 66.40 | 76.65 |
| | CGCS [11] | 74.66 | 50.67 | 56.00 | 68.59 | 88.37 | 57.33 | 63.33 | 72.76 |
| | LR-DC [36] | 79.56 | 44.00 | 54.80 | 66.52 | 90.67 | 68.80 | 76.16 | 84.30 |
| | PT+NCM [16] | 80.57 | 40.00 | 52.67 | 65.83 | 91.15 | 69.33 | 76.13 | 84.67 |
| | $\mathrm{S2M2}_R$ [23] | 80.68 | 52.00 | 56.56 | 69.70 | 90.85 | 73.86 | 77.72 | 85.47 |
| | AC+SR (ours) | 85.14 | 52.78 | 57.46 | 71.31 | 94.42 | 76.00 | 80.83 | 87.76 |
| | AC+EnSR (ours) | **85.42** | **53.04** | **58.10** | **71.58** | **94.53** | **77.58** | **81.12** | **87.99** |
| CIFAR-FS | ProtoNet$^\diamond$ [29] | 61.60 | 25.34 | 30.62 | 43.08 | 79.08 | 51.20 | 57.40 | 66.62 |
| | Inductive-FT [7] | 68.72 | 33.33 | 38.40 | 50.32 | 86.11 | 56.00 | 61.47 | 71.73 |
| | Negative-Cosine$^\diamond$ [21] | 68.90 | 26.66 | 39.76 | 52.58 | 83.82 | 52.00 | 61.54 | 71.88 |
| | MixtFSL$^\diamond$ [2] | 69.42 | 29.33 | 38.53 | 51.51 | 81.05 | 57.33 | 62.40 | 69.19 |
| | LR-DC$^\diamond$ [36] | 72.52 | 32.26 | 42.08 | 54.72 | 83.92 | 57.60 | 63.30 | 72.48 |
| | CGCS [11] | 73.00 | 33.33 | 41.33 | 55.53 | 85.80 | 56.00 | 62.13 | 69.16 |
| | PT+NCM [16] | 74.64 | 34.67 | 42.53 | 56.60 | **87.64** | 54.67 | 62.13 | 71.57 |
| | $\mathrm{S2M2}_R$ [23] | 74.81 | 35.74 | 42.06 | 56.49 | 87.47 | 56.26 | 63.38 | 72.17 |
| | AC+SR (ours) | 74.00 | **36.54** | 42.73 | 56.83 | 86.65 | 56.26 | 65.97 | 75.23 |
| | AC+EnSR (ours) | **74.85** | 34.96 | **42.91** | **57.70** | 87.27 | **59.48** | **66.61** | **75.84** |

$^\diamond$ $\mathrm{ACC}_m$ from our implementation $^\dagger$ $\mathrm{ACC}_m$ from [5]

are regarded as $D_n$ to adapt the pretrained model to these new categories, afterwards the model is evaluated on the $N \times Q$ test samples. The above process is collectively called one episode. We keep $N$=5, $Q$=15 and vary $K$=1 or 5 to conduct experiments. We report the average of all ($\mathrm{Acc}_m$), worst 1 ($\mathrm{Acc}_1$), average of worst 10 ($\mathrm{Acc}_{10}$), average of worst 100 ($\mathrm{Acc}_{100}$) accuracy and standard deviation ($\sigma$) over 500 pre-sampled episodes by 5 runs then average. Specifically for previous methods with only $\mathrm{Acc}_m$ reported, we get other results following their official implementations. All experiments we conduct on one dataset are based on the same 500 episodes, which makes the comparison fair.

### 4.2 Comparing with State-of-the-art Methods

We compare 5-way accuracy of our method with state-of-the-art methods in Table 2. 'AC' and 'SR' stand for adaptability calibration and stability regularization, respectively, and 'EnSR' means ensemble of SR.

The proposed method (denoted as AC+SR) outperforms existing methods using the traditional $\mathrm{ACC}_m$ metric by 1–5% in both 1-shot and 5-shot scenarios on the first two datasets, and is slightly ($< 1\%$) worse on CIFAR-FS. When model ensemble is used, our AC+EnSR almost consistently outperforms existing methods.

More importantly, in the worst-case ($\mathrm{ACC}_1$) and near-worst case average metrics ($\mathrm{ACC}_{10}$, $\mathrm{ACC}_{100}$), our methods are almost consistently better, and mostly higher by $> 2\%$ margins. In the worst-case metrics, model ensemble (AC+EnSR) is again better than single model. And, the ranking from $\mathrm{ACC}_m$ is significantly different from that based on $\mathrm{ACC}_1$, but the ranking orders are almost consistent among the 3 worst-case metrics. Hence, the worst-case metric is more *stable* than the average accuracy metric.

### 4.3 Relationships: $\mu$, $\sigma$, $\mu - 3\sigma$, $\mathrm{ACC}_1$ and $\mathrm{ACC}_m$

Table 3 presents a few important evaluation metrics, including the commonly used average accuracy $\mathrm{ACC}_m$ ($\mu$), the proposed worst-case accuracy $\mathrm{ACC}_1$ that we advocate, the standard deviation $\sigma$, and the surrogate $\mu - 3\sigma$ that motivates the proposed method.

As aforementioned, because the worst-case accuracy $\mathrm{ACC}_1$ is unable to be directly modeled or optimized using few-shot samples, we alternatively resort to the surrogate $\mu - 3\sigma$ inspired by the $3\sigma$ rule. This objective is again difficult because $\sigma$ is unable to be directly accessed, and we resort to the bias-variance decomposition: reducing $\sigma$ and increasing $\mu$ simultaneously.

As Table 3 shows, both $\mu$ and $\sigma$ are correlated with $\mathrm{ACC}_1$, but not tightly correlated. For example, $\mathrm{S2M2}_R$ [23] often has the smallest $\sigma$, but because its $\mu$ ($\mathrm{ACC}_m$) is not the most competitive, its worst-case $\mathrm{ACC}_1$ is significantly lower than ours. We do need to optimize both $\mu$ and $\sigma$ simultaneously.

$\mu - 3\sigma$, which motivates our method, correlates better with $\mathrm{ACC}_1$ than $\sigma$ or $\mu$, as Table 3 shows. However, this quantity is still far from being a perfect surrogate. It is still a great challenge to model the worst-case $\mathrm{ACC}_1$ end-to-end.

### 4.4 Ablation Analyses

**Switching backbone.** Table 4 shows the results of 5-way classification on *mini*-ImageNet by switching to different backbones. Our adaptability calibration (AC) alone is sometimes harmful in the 1-shot scenario, but always beneficial for 5-shot, which coincides well with the observations in Fig. 3.

On the other hand, the proposed stability regularization (SR) is consistently helpful in all cases and all metrics, especially in the 1-shot case. The proposed model ensemble method (EnSR) is almost consistently improving all evaluation metrics on all backbone models, in particular for the difficult 1-shot scenario.

**Switching pretraining methods.** We further conduct experiments to explore the influence of pretraining methods on the proposed strategies. Specifically, we additionally adopt two pretraining methods from [5], denoted as Baseline and Baseline++, respectively, for simplicity. As shown in Table 5, when pretrained with Baseline, after applying AC and SR, $\mathrm{ACC}_1$ and $\mathrm{ACC}_{10}$ are increased (on average by 2% in 1-shot and 1% in 5-shot), although $\mathrm{ACC}_m$ is slightly

**Table 3.** Key evaluation metrics, including the commonly used average accuracy ACC$_m$ ($\mu$), the proposed worst-case accuracy ACC$_1$, the standard deviation $\sigma$, and the surrogate $\mu - 3\sigma$ that motivates the proposed method.

| Dataset | Method | **1-shot** | | | | **5-shot** | | | |
|---|---|---|---|---|---|---|---|---|---|
| | | ACC$_1$ | $\mu - 3\sigma$ | $\sigma$ | ACC$_m$ | ACC$_1$ | $\mu - 3\sigma$ | $\sigma$ | ACC$_m$ |
| *mini*-ImageNet | ProtoNet$^\dagger$ [29] | 19.76 | 23.41 | 10.25 | 54.16 | 43.74 | 49.32 | 8.12 | 73.68 |
| | Transductive-FT [7] | 24.00 | 32.82 | 10.97 | 65.73 | 50.67 | 53.23 | 8.39 | 78.40 |
| | Negative-Cosine [21] | 24.27 | 31.36 | 10.12 | 61.72 | 53.30 | 61.18 | 6.87 | 81.79 |
| | MixtFSL [2] | 30.67 | 34.70 | 9.87 | 64.31 | 46.67 | 59.16 | 7.50 | 81.66 |
| | PT+NCM [16] | 32.00 | 34.75 | 10.20 | 65.35 | 56.00 | 63.98 | 6.63 | 83.87 |
| | LR-DC [36] | 37.33 | 37.73 | 10.28 | 68.57 | 60.52 | 63.02 | 6.62 | 82.88 |
| | S2M2$_R$ [23] | 37.58 | 37.39 | **9.18** | 64.93 | 58.66 | 66.35 | **5.61** | 83.18 |
| | CGCS [11] | 38.70 | 36.42 | 10.20 | 67.02 | 49.30 | 60.90 | 7.14 | 82.32 |
| | AC+SR (ours) | 40.52 | 40.25 | 9.71 | 69.38 | 63.20 | 66.46 | 6.47 | 85.87 |
| | AC+EnSR (ours) | **40.52** | **40.67** | 9.64 | **69.59** | **63.48** | **66.71** | 6.42 | **85.97** |
| CUB | ProtoNet$^\dagger$ [29] | 28.00 | 39.99 | 11.00 | 72.99 | 53.33 | 67.53 | 6.37 | 86.64 |
| | Negative-Cosine [21] | 36.00 | 40.80 | 10.62 | 72.66 | 70.70 | 73.29 | 5.37 | 89.40 |
| | PT+NCM [16] | 40.00 | 49.97 | 10.20 | 80.57 | 69.33 | 75.85 | 5.10 | 91.15 |
| | MixtFSL [2] | 40.00 | 32.69 | 13.75 | 73.94 | 57.33 | 67.26 | 6.25 | 86.01 |
| | LR-DC [36] | 44.00 | 49.74 | 9.94 | 79.56 | 68.80 | 75.49 | 5.06 | 90.67 |
| | CGCS [11] | 50.67 | 42.53 | 10.71 | 74.66 | 57.33 | 70.01 | 6.12 | 88.37 |
| | S2M2$_R$ [23] | 52.00 | 50.32 | 10.12 | 80.68 | 73.86 | 74.35 | 5.50 | 90.85 |
| | AC+SR (ours) | 52.78 | 58.44 | 8.90 | 85.14 | 76.00 | 81.85 | 4.19 | 94.42 |
| | AC+EnSR (ours) | **53.04** | **58.84** | **8.86** | **85.42** | **77.58** | **82.14** | **4.13** | **94.53** |
| CIFAR-FS | ProtoNet$^\diamond$ [29] | 25.34 | 23.38 | 12.74 | 61.60 | 51.20 | 53.58 | 8.50 | 79.08 |
| | Negative-Cosine$^\diamond$ [21] | 26.66 | 34.49 | 11.47 | 68.90 | 52.00 | 60.21 | 7.87 | 83.82 |
| | MixtFSL$^\diamond$ [2] | 29.33 | 33.09 | 12.11 | 69.42 | 57.33 | 56.48 | 8.19 | 81.05 |
| | LR-DC$^\diamond$ [36] | 32.26 | 36.94 | 11.86 | 72.52 | 57.60 | 60.94 | 7.66 | 83.92 |
| | Inductive-FT [7] | 33.33 | 36.29 | 10.81 | 68.72 | 56.00 | 63.37 | 7.58 | 86.11 |
| | CGCS [11] | 33.33 | 39.13 | 11.29 | 73.00 | 56.00 | 61.59 | 8.07 | 85.80 |
| | PT+NCM [16] | 34.67 | 42.51 | 10.71 | 74.64 | 54.67 | 64.69 | 7.65 | **87.64** |
| | S2M2$_R$ [23] | 35.74 | **45.74** | **9.69** | 74.81 | 56.26 | **67.58** | **6.63** | 87.47 |
| | AC+SR (ours) | **36.54** | 39.35 | 11.55 | 74.00 | 56.26 | 64.45 | 7.40 | 86.65 |
| | AC+EnSR (ours) | 34.96 | 40.14 | 11.57 | **74.85** | **59.48** | 65.16 | 7.37 | 87.27 |

$^\diamond$ ACC$_m$ from our implementation $^\dagger$ ACC$_m$ from [5]

decreased (about 0.2%). This observation not only shows our AC+SR's effectiveness, but also indicates that average and worst-case accuracy are not aligned in general. With AC+EnSR, the accuracy for all metrics gains significant margins than the naive Baseline method, in particular, 4-5% for 1-shot.

When pretraining with Baseline++, the accuracy in all cases are generally improved by gradually deploying AC+SR or AC+EnSR, too. The proposed techniques are compatible with various pretraining methods.

**Calibrating adaptability differently.** Table 6 shows the results by varying the number of learnable convolutional groups in ResNet 18 to experiment with different calibration of adaptability. First, there is a clear distinction between the first row ($W$ only) and other rows, indicating that it is harmful to freeze the entire backbone and only fine-tune the classification head $W$.

Second, when more residual groups are made learnable, there is not a clear winner, especially when we consider different evaluation metrics. For example, every row in Table 6's last 3 rows has been the champion for at least one evalua-

**Table 4.** Ablation results of 5-way classification on *mini*-ImageNet with different backbones. The first row is the baseline and our AC or SR are gradually incorporated. Note that EnSR always includes SR.

| Backbone | AC | SR | EnSR | **1-shot** | | | | | **5-shot** | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | | $ACC_m$ | $\sigma$ | $ACC_1$ | $ACC_{10}$ | $ACC_{100}$ | $ACC_m$ | $\sigma$ | $ACC_1$ | $ACC_{10}$ | $ACC_{100}$ |
| ResNet 18 | | | | 59.53 | **9.99** | 30.42 | 35.09 | 45.05 | 73.61 | 8.16 | 44.54 | 50.16 | 61.36 |
| | ✓ | | | 60.09 | 10.13 | 25.60 | 34.55 | 45.24 | 78.58 | 7.62 | 45.86 | 56.34 | 67.32 |
| | ✓ | ✓ | | 62.33 | 10.27 | 32.00 | 37.03 | 47.36 | 79.02 | 7.68 | 46.14 | 56.48 | 67.62 |
| | ✓ | ✓ | ✓ | **62.76** | 10.16 | **32.78** | **37.55** | **47.84** | **79.23** | **7.62** | **46.14** | **56.96** | **67.91** |
| ResNet 34 | | | | 59.94 | **9.64** | 30.94 | 35.73 | 46.08 | 74.63 | 7.68 | 45.60 | 52.25 | 63.17 |
| | ✓ | | | 56.76 | 9.81 | 24.56 | 32.09 | 42.51 | 77.58 | 7.54 | 47.48 | 55.54 | 66.38 |
| | ✓ | ✓ | | 61.42 | 9.87 | 32.54 | 37.17 | 47.00 | 78.59 | 7.43 | 51.46 | 57.76 | 67.43 |
| | ✓ | ✓ | ✓ | **61.80** | 9.90 | **33.86** | **37.76** | **47.26** | **78.74** | **7.41** | **51.74** | **58.03** | **67.58** |
| ResNet 50 | | | | 62.44 | 10.04 | 29.32 | 36.64 | 47.74 | 76.92 | 7.93 | 50.66 | 54.67 | 65.04 |
| | ✓ | | | 61.56 | 9.85 | 29.06 | 36.69 | 47.25 | 80.15 | **7.21** | 52.26 | **59.07** | **69.29** |
| | ✓ | ✓ | | 63.60 | 9.90 | 30.94 | 37.36 | 49.00 | 80.33 | 7.46 | 53.06 | 57.89 | 68.93 |
| | ✓ | ✓ | ✓ | **63.79** | **9.84** | **31.67** | **38.05** | **49.36** | **80.43** | 7.36 | **53.30** | 58.89 | 69.24 |
| WRN-28-10 | | | | 67.86 | 9.89 | 37.58 | 42.87 | 53.40 | 84.57 | 6.50 | 58.66 | 66.21 | 74.73 |
| | ✓ | | | 67.36 | 9.79 | 36.80 | 41.44 | 52.80 | 85.53 | 6.50 | 60.54 | 65.63 | 75.95 |
| | ✓ | ✓ | | 69.38 | 9.71 | 40.52 | 44.51 | 54.97 | 85.87 | 6.47 | 63.20 | 66.51 | 76.28 |
| | ✓ | ✓ | ✓ | **69.59** | **9.64** | **40.52** | **44.94** | **55.32** | **85.97** | **6.42** | **63.48** | **66.74** | **76.40** |

**Table 5.** Ablation results of 5-way classification on *mini*-ImageNet with the ResNet 18 backbone pretrained by other methods.

| Pretraining Method | AC | SR | EnSR | **1-shot** | | | | | **5-shot** | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | | $ACC_m$ | $\sigma$ | $ACC_1$ | $ACC_{10}$ | $ACC_{100}$ | $ACC_m$ | $\sigma$ | $ACC_1$ | $ACC_{10}$ | $ACC_{100}$ |
| Baseline [5] | | | | 50.37 | 10.59 | 18.42 | 24.91 | 35.43 | 74.13 | **8.14** | 44.82 | 51.20 | 62.15 |
| | ✓ | ✓ | | 50.27 | 10.37 | 21.86 | 26.45 | 35.43 | 73.92 | 8.22 | 46.94 | 51.49 | 61.85 |
| | ✓ | ✓ | ✓ | **54.42** | **10.26** | **23.20** | **29.39** | **39.61** | **74.88** | 8.16 | **47.46** | **52.89** | **62.78** |
| Baseline++ [5] | | | | 57.59 | **9.50** | 29.06 | 33.58 | 43.99 | 75.35 | 7.66 | 49.08 | 55.66 | 64.17 |
| | ✓ | ✓ | | 59.30 | 9.91 | 29.58 | 34.08 | 44.77 | 77.57 | 7.63 | 54.68 | 58.36 | 66.19 |
| | ✓ | ✓ | ✓ | **59.58** | 9.82 | **30.66** | **34.83** | **45.20** | **77.73** | **7.60** | **56.80** | **58.94** | **66.51** |

tion metric in 5-shot. But, if we consider 1-shot, it is obviously observed that the second row (fine-tuning $W$+'res5') is significantly better. This is the adaptability calibration we choose in our experiments.

**Generalize SR to other data.** Since only raw images are required to compute the SR loss, it enjoys the flexibility to use others images (i.e., those not from $D_b$) for its computation (i.e., being generalizable). We use CUB [34], Cars [18], Describable Textures (DTD) [6], Pets [25], VGG Flower (Flower) [24] and CIFAR-100 [19] as the images to compute our stability regularization loss when the base set is *mini*-ImageNet. All these images are resized to 84×84.

Results in Table 7 reveals an interesting observation. No matter what images are used to compute the SR loss, the accuracy is consistently higher than the baseline (AC without SR), regardless of which metric is considered. That is, the stability regularization is indeed generalizable.

Moreover, although $D_b$ (the base set) performs the best in general, using other images for SR computation leads to results that are on par with it. That is, we *do not* require images used in this regularization to be visually similar or semantically correlated. Our stability regularization is *consistently useful.*

**Table 6.** Ablation results of 5-way classification on *mini*-ImageNet with ResNet 18 as the backbone by varying the number of learnable blocks (✓ means learnable) to control the degree of AC. The number beneath the column AC stands for the index of convolutional groups in ResNet 18. AC+EnSR is used in all experiments.

| $W$ | AC | | | | **1-shot** | | | | | **5-shot** | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | 5 | 4 | 3 | 2+1 | ACC$_m$ | $\sigma$ | ACC$_1$ | ACC$_{10}$ | ACC$_{100}$ | ACC$_m$ | $\sigma$ | ACC$_1$ | ACC$_{10}$ | ACC$_{100}$ |
| ✓ | | | | | 59.53 | **9.99** | 30.42 | 35.09 | 45.05 | 73.61 | 8.16 | 44.54 | 50.16 | 61.36 |
| ✓ | ✓ | | | | 62.76 | 10.16 | **32.78** | **37.55** | **47.84** | 79.23 | 7.62 | 46.14 | 56.96 | 67.91 |
| ✓ | ✓ | ✓ | | | 62.77 | 10.38 | 30.12 | 35.44 | 47.46 | 79.91 | **7.54** | 45.60 | **57.84** | 68.77 |
| ✓ | ✓ | ✓ | ✓ | | **62.80** | 10.50 | 22.96 | 34.93 | 47.27 | **80.04** | 7.57 | 45.58 | 57.74 | **68.85** |
| ✓ | ✓ | ✓ | ✓ | ✓ | 62.19 | 10.50 | 25.60 | 35.29 | 46.68 | 79.87 | 7.58 | **48.00** | 57.41 | 68.67 |

**Table 7.** 5-way 1-shot results on *mini*-ImageNet by deploying AC+SR with ResNet 18 as the backbone, while the SR loss is computed with images from *different* datasets, denoted as $D_{sr}$. The first row ('$-$') is the baseline with only AC, and the second row is AC+SR, where SR uses the base set (*mini*-ImageNet).

| $D_{sr}$ | ACC$_m$ | $\sigma$ | ACC$_1$ | ACC$_{10}$ | ACC$_{100}$ |
|---|---|---|---|---|---|
| - | 60.09 | 10.13 | 25.60 | 34.55 | 45.24 |
| *mini*-ImageNet | **62.33** | 10.27 | 32.00 | **37.03** | **47.36** |
| CUB [34] | 60.66 | 9.99 | 30.14 | 36.43 | 46.29 |
| Cars [18] | 61.05 | 10.29 | 30.94 | 36.64 | 46.20 |
| DTD [6] | 61.99 | 10.23 | 31.20 | 36.75 | 47.23 |
| Pets [25] | 61.57 | **9.95** | 32.00 | 36.62 | 47.27 |
| Flower [24] | 61.03 | 10.07 | **32.26** | 36.51 | 46.37 |
| CIFAR-100 [19] | 60.47 | 10.04 | 30.68 | 36.69 | 46.02 |

## 5 Conclusions

This paper advocated to use the worst-case accuracy in optimizing and evaluating few-shot learning methods, which is a better fit to real-world applications than the commonly used average accuracy. Worst-case accuracy, however, is much more difficult to work on. We designed a surrogate loss inspired by the $3\sigma$ rule, and in turn proposed two strategies to implicitly optimize this surrogate: a stability regularization (SR) loss together with model ensemble to reduce the variance, and an adaptability calibration (AC) to vary the number of learnable parameters to reduce the bias.

The proposed strategies have achieved significantly higher worst-case (and also average) accuracy than existing methods. In the future, we will design more direct attacks to reduce the worst-case error, because the current surrogate is not a highly accurate approximation of the worst-case performance yet.

**Acknowledgments.** This research was partly supported by the National Natural Science Foundation of China under Grant 61921006 and Grant 61772256.

## References

1. Afrasiyabi, A., Lalonde, J.F., Gagné, C.: Associative alignment for few-shot image classification. In: European Conference on Computer Vision (2020)
2. Afrasiyabi, A., Lalonde, J.F., Gagné, C.: Mixture-based feature space learning for few-shot image classification. In: IEEE/CVF International Conference on Computer Vision (2021)
3. Bateni, P., Goyal, R., Masrani, V., Wood, F., Sigal, L.: Improved few-shot visual classification. In: IEEE/CVF Conference on Computer Vision and Pattern Recognition (2020)
4. Bertinetto, L., Henriques, J.F., Torr, P.H., Vedaldi, A.: Meta-learning with differentiable closed-form solvers. In: International Conference on Learning Representations (2019)
5. Chen, W.Y., Liu, Y.C., Kira, Z., Wang, Y.C.F., Huang, J.B.: A closer look at few-shot classification. In: International Conference on Learning Representations (2019)
6. Cimpoi, M., Maji, S., Kokkinos, I., Mohamed, S., Vedaldi, A.: Describing textures in the wild. In: IEEE Conference on Computer Vision and Pattern Recognition (2014)
7. Dhillon, G.S., Chaudhari, P., Ravichandran, A., Soatto, S.: A baseline for few-shot image classification. In: International Conference on Learning Representations (2020)
8. Dvornik, N., Schmid, C., Mairal, J.: Diversity with cooperation: Ensemble methods for few-shot classification. In: IEEE/CVF International Conference on Computer Vision (2019)
9. Finn, C., Abbeel, P., Levine, S.: Model-agnostic meta-learning for fast adaptation of deep networks. In: International Conference on Machine Learning (2017)
10. Finn, C., Xu, K., Levine, S.: Probabilistic model-agnostic meta-learning. In: Advances in Neural Information Processing Systems (2018)
11. Gao, Z., Wu, Y., Jia, Y., Harandi, M.: Curvature generation in curved spaces for few-shot learning. In: IEEE/CVF International Conference on Computer Vision (2021)
12. Gidaris, S., Bursuc, A., Komodakis, N., Pérez, P., Cord, M.: Boosting few-shot visual learning with self-supervision. In: IEEE/CVF International Conference on Computer Vision (2019)
13. Gidaris, S., Komodakis, N.: Generating classification weights with gnn denoising autoencoders for few-shot learning. In: IEEE/CVF Conference on Computer Vision and Pattern Recognition (2019)
14. Hastie, T., Tibshirani, R., Friedman, J.H., Friedman, J.H.: The elements of statistical learning: data mining, inference, and prediction, vol. 2. Springer (2009)
15. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: IEEE Conference on Computer Vision and Pattern Recognition (2016)
16. Hu, Y., Gripon, V., Pateux, S.: Leveraging the feature distribution in transfer-based few-shot learning. In: International Conference on Artificial Neural Networks (2021)
17. Kim, J., Oh, T.H., Lee, S., Pan, F., Kweon, I.S.: Variational prototyping-encoder: One-shot learning with prototypical images. In: IEEE/CVF Conference on Computer Vision and Pattern Recognition (2019)
18. Krause, J., Stark, M., Deng, J., Fei-Fei, L.: 3d object representations for fine-grained categorization. In: IEEE International Conference on Computer Vision Workshops (2013)

19. Krizhevsky, A., Hinton, G.: Learning multiple layers of features from tiny images. Tech. rep., University of Toronto (2009)
20. Lee, K., Maji, S., Ravichandran, A., Soatto, S.: Meta-learning with differentiable convex optimization. In: IEEE/CVF Conference on Computer Vision and Pattern Recognition (2019)
21. Liu, B., Cao, Y., Lin, Y., Li, Q., Zhang, Z., Long, M., Hu, H.: Negative margin matters: Understanding margin in few-shot classification. In: European Conference on Computer Vision (2020)
22. Liu, Y., Schiele, B., Sun, Q.: An ensemble of epoch-wise empirical bayes for few-shot learning. In: European Conference on Computer Vision (2020)
23. Mangla, P., Kumari, N., Sinha, A., Singh, M., Krishnamurthy, B., Balasubramanian, V.N.: Charting the right manifold: Manifold mixup for few-shot learning. In: IEEE/CVF Winter Conference on Applications of Computer Vision (2020)
24. Nilsback, M.E., Zisserman, A.: Automated flower classification over a large number of classes. In: 2008 Sixth Indian Conference on Computer Vision, Graphics & Image Processing (2008)
25. Parkhi, O.M., Vedaldi, A., Zisserman, A., Jawahar, C.: Cats and dogs. In: IEEE Conference on Computer Vision and Pattern Recognition (2012)
26. Ravi, S., Larochelle, H.: Optimization as a model for few-shot learning. In: International Conference on Learning Representations (2017)
27. 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) (2015)
28. Rusu, A.A., Rao, D., Sygnowski, J., Vinyals, O., Pascanu, R., Osindero, S., Hadsell, R.: Meta-learning with latent embedding optimization. In: International Conference on Learning Representations (2019)
29. Snell, J., Swersky, K., Zemel, R.: Prototypical networks for few-shot learning. In: Advances in Neural Information Processing Systems (2017)
30. Sung, F., Yang, Y., Zhang, L., Xiang, T., Torr, P.H., Hospedales, T.M.: Learning to compare: Relation network for few-shot learning. In: IEEE Conference on Computer Vision and Pattern Recognition (2018)
31. Szegedy, C., Vanhoucke, V., Ioffe, S., Shlens, J., Wojna, Z.: Rethinking the inception architecture for computer vision. In: IEEE Conference on Computer Vision and Pattern Recognition (2016)
32. Verma, V., Lamb, A., Beckham, C., Najafi, A., Mitliagkas, I., Lopez-Paz, D., Bengio, Y.: Manifold mixup: Better representations by interpolating hidden states. In: International Conference on Machine Learning (2019)
33. Vinyals, O., Blundell, C., Lillicrap, T., Wierstra, D., et al.: Matching networks for one shot learning. In: Advances in Neural Information Processing Systems (2016)
34. Wah, C., Branson, S., Welinder, P., Perona, P., Belongie, S.: The Caltech-UCSD birds-200-2011 dataset. Tech. Rep. CNS-TR-2011-001, California Institute of Technology (2011)
35. Wang, Y., Yao, Q., Kwok, J.T., Ni, L.M.: Generalizing from a few examples: A survey on few-shot learning. ACM Computing Surveys **53**(3) (2020)
36. Yang, S., Liu, L., Xu, M.: Free lunch for few-shot learning: Distribution calibration. In: International Conference on Learning Representations (2021)
37. Zagoruyko, S., Komodakis, N.: Wide residual networks. In: British Machine Vision Conference (2016)
38. Zhou, Z.H.: Ensemble Methods: Foundations and Algorithms. Chapman and Hall/CRC (2012)