Bridging the Gap between Few-Shot and Many-Shot Learning via Distribution Calibration

Shuo Yang, Songhua Wu, Tongliang Liu *Senior Member, IEEE*, Min Xu *Member, IEEE*

Abstract—A major gap between few-shot and many-shot learning is the data distribution empirically observed by the model during training. In few-shot learning, the learned model can easily become over-fitted based on the biased distribution formed by only a few training examples, while the ground-truth data distribution is more accurately uncovered in many-shot learning to learn a well-generalized model. In this paper, we propose to calibrate the distribution of these few-sample classes to be more unbiased to alleviate such an over-fitting problem. The distribution calibration is achieved by transferring statistics from the classes with sufficient examples to those few-sample classes. After calibration, an adequate number of examples can be sampled from the calibrated distribution to expand the inputs to the classifier. Specifically, we assume every dimension in the feature representation from the same class follows a Gaussian distribution so that the mean and the variance of the distribution can borrow from that of similar classes whose statistics are better estimated with an adequate number of samples. Extensive experiments on three datasets, *mini*ImageNet, *tiered*ImageNet, and CUB, show that a simple linear classifier trained using the features sampled from our calibrated distribution can outperform the state-of-the-art accuracy by a large margin. Besides the favorable performance, the proposed method also exhibits high flexibility by showing consistent accuracy improvement when it is built on top of any off-the-shelf pretrained feature extractors and classification models without extra learnable parameters. The visualization of these generated features demonstrates that our calibrated distribution is an accurate estimation thus the generalization ability gain is convincing. We also establish a generalization error bound for the proposed distribution-calibration-based few-shot learning, which consists of the *distribution assumption error*, the *distribution approximation error*, and the *estimation error*. This generalization error bound theoretically justifies the effectiveness of the proposed method.

Index Terms—few-shot learning, image classification, transfer learning, generalization error

1 Introduction

Learning from a limited number of training samples has drawn increasing attention due to the high cost of collecting and annotating a large amount of data. Researchers have developed algorithms to improve the performance of models that have been trained with very few data. [1], [2] train models in a meta-learning fashion so that the model can adapt quickly to tasks with only a few training samples available. [3], [4] try to synthesize data or features by learning a generative model to alleviate the data insufficiency problem. [5] propose to leverage unlabeled data and predict pseudo labels to improve the performance of few-shot learning.

While most previous works focus on developing stronger models, scant attention has been paid to the property of the data itself. It is natural that when the number of data grows, the ground-truth distribution can be more accurately uncovered. Models trained with a wide coverage of data can generalize well during evaluation. On the other hand, when training a model with only a few training data, the model tends to overfit on these few samples by minimizing the training loss over these samples. These phenomena are illustrated in Figure 1. This biased distribution based on a few examples greatly limits the generalization ability of the model since it is far from mirroring the ground-truth distribution from which test cases are sampled during evaluation.

- *S. Yang and M. Xu are with the School of Electrical and Data Engineering, Faculty of Engineering and Information Technology, University of Technology Sydney, 15 Broadway, Ultimo, NSW 2007, Australia; shuo.yang@student.uts.edu.au, min.xu@uts.edu.au*
- *S. Wu and T. Liu are with the Trustworthy Machine Learning Lab, School of Computer Science, the University of Sydney, 1 Cleveland St, Darlington, NSW 2008, Australia; songhua.wu, tongliang.liu@sydney.edu.au*

Corresponding author: Min Xu

	Arctic fox	
	mean sim	var sim
white wolf	97%	97%
malamute	85%	78%
lion	81%	70%
meerkat	78%	70%
black footed ferret	77%	73%
golden retriever	74%	64%
jellyfish	46%	26%
orange	40%	19%
beer bottle	34%	11%

TABLE 1: The class mean similarity ("mean sim") and class variance similarity ("var sim") between Arctic fox and different classes.

Here, we consider calibrating this biased distribution into a more accurate approximation of the ground-truth distribution. In this way, a model trained with inputs sampled from the calibrated distribution can generalize over a broader range of data from a more accurate distribution rather than only fitting itself to those few samples.

Instead of calibrating the distribution of the original data space, we try to calibrate the distribution in the feature space, which has much lower dimensions and is easier to calibrate [6]. We assume every dimension in the feature vectors from the same class follows a Gaussian distribution and observe that similar classes usually have similar mean and variance of the feature representations, as shown in Table 1. Thus, the mean and variance of the Gaussian distribution can be transferred across similar classes [7]. Meanwhile, the statistics can be estimated more accurately when there are adequate samples in the source domain. Based on these observations, we

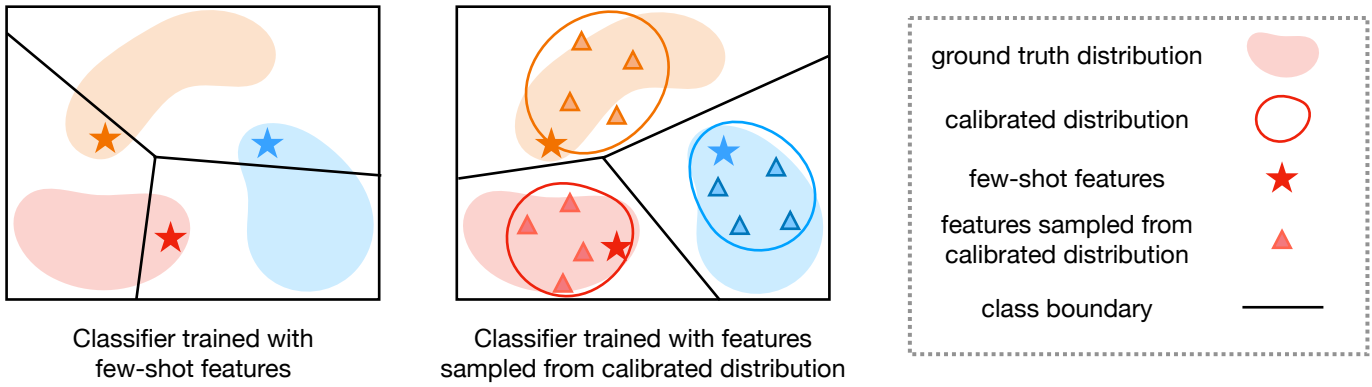


Fig. 1: Training a classifier from few-shot features makes the classifier overfit to the few examples (Left). Classifier trained with features sampled from calibrated distribution has better generalization ability (Right).

reuse the statistics from many-shot classes and transfer them to better estimate the distribution of the few-shot classes according to their class similarity. More samples can be generated according to the estimated distribution which provides sufficient supervision for training the classification model.

We analyze the generalization error for the proposed distribution-calibration-based few-shot learning. The error is bounded by the *distribution assumption error*: the gap between the ground-truth feature representation distribution and the assumed Gaussian distribution; the *distribution approximation error*: the gap between the assumed Gaussian distribution and the approximated calibrated Gaussian distribution; and the *estimation error* induced by the deviation of the obtained network parameters to the optimal ones because of finite samples, that theoretically justify the effectiveness of the proposed method.

In the experiments, we show that a simple logistic regression classifier trained with our strategy can achieve state-of-the-art accuracy on three datasets. Our distribution calibration strategy can be paired with any classifier and feature extractor with no extra learnable parameters. Training with samples selected from the calibrated distribution can achieve a 12% accuracy gain compared to the baseline which is only trained with the few samples given in a 5way1shot task. We also visualize the calibrated distribution and show that it is an accurate approximation of the ground-truth that can better cover the test cases.

This work is an extension of an ICLR 2021 oral presentation [8]. Compared to the preliminary version, a theoretical framework that bounds the generalization error of the proposed method is newly established in Section 4. The proof of the established generalization error bound is provided in Section 7. Additional discussions and empirical results that verify the established theoretical framework are also included in Section 5.5.1 and Section 5.5.2, respectively. Our code is open-sourced at: https://github.com/ShuoYang-1998/Few_Shot_Distribution_Calibration.

2 Related Work

Few-shot classification is a challenging machine learning problem in weakly-supervised learning [1], [9], [10], [11], [12], [13], [14], [15], [16], which usually requires a pretrained classifier or learning algorithm can quickly adapt to new tasks with very limited training examples. Researchers have explored the idea of learning to learn or meta-learning to improve the quick adaptation ability to alleviate the few-shot challenge. One of the most general algorithms for meta-learning is the optimization-based algorithm. MAML [1] and Meta-SGD [9] proposed to learn how to optimize the gradient descent procedure so that the learner can have a good initialization, update direction, and learning rate. For the classification problem, researchers proposed simple but effective algorithms based on metric learning. MatchingNet [17] and ProtoNet [2] learned to classify samples by comparing the distance to the representatives of each class. Our distribution calibration and feature sampling procedure does not include any learnable parameters and the classifier is trained in a traditional supervised learning way.

Another line of algorithms is to compensate for the insufficient number of available samples by generation. Most methods use the idea of Generative Adversarial Networks (GANs) [18] or autoencoder [19] to generate samples [20], [21], [22], [23] or features [6], [24], [25] to augment the training set. Specifically, [20] and [6] proposed to synthesize data by introducing an adversarial generator conditioned on tasks. [24] tried to learn a variational autoencoder to approximate the distribution and predict labels based on the estimated statistics. The autoencoder can also augment samples by projecting between the visual space and the semantic space [21] or encoding the intra-class deformations [22]. While these methods can generate extra samples or features for training, they require the design of a complex model and loss function to learn how to generate. However, our distribution calibration strategy is simple and does not need extra learnable parameters.

Data augmentation is a traditional and effective way of increasing the number of training samples. Qin et al. [26] and Antoniou et al. [27] proposed the use of the traditional data augmentation technique to construct pretext tasks for unsupervised few-shot learning. [4] and [3] leveraged the general idea of data augmentation, they designed a hallucination model to generate the augmented version of the image with different choices for the model's input, i.e., an image and a noise [4] or the concatenation of multiple features [3], [28]. [29] tried to augment feature representations by sampling from an estimated variance. These methods learn to augment from the original samples or their feature representation while we try to estimate the class-level distribution and thus can eliminate the inductive bias from a single sample and provide more diverse generations from the calibrated distribution.

3 Main Approach

In this section, we introduce the few-shot classification problem definition in Section 3.1 and details of our proposed approach in

Section 3.2.

3.1 Problem Definition

We follow a typical few-shot classification setting. Given a dataset with data-label pairs $\mathcal{D} = \{(\boldsymbol{x}_i, y_i)\}$ where $\boldsymbol{x}_i \in \mathbb{R}^d$ is the feature vector of a sample and $y_i \in C$ is the class label of $\boldsymbol{x}_i$, where C denotes the set of classes. This set of classes is divided into *base classes* C_b and *novel classes* C_n, where $C_b \cap C_n = \varnothing$ and $C_b \cup C_n = C$. The goal is to train a model on the data from the base classes so that the model can generalize well on tasks sampled from the novel classes. In order to evaluate the fast adaptation ability or the generalization ability of the model, there are only a few available labeled samples for each task $\mathcal{T}$. The most common way to build a task is called an N-way-K-shot task [17], where N classes are sampled from the novel set and only K (e.g., 1 or 5) labeled samples are provided for each class. The few available labeled data are called *support set* $\mathcal{S} = \{(\boldsymbol{x}_i, y_i)\}_{i=1}^{N\times K}$ and the model is evaluated on another query set $\mathcal{Q} = \{(\boldsymbol{x}_i, y_i)\}_{i=N\times K+1}^{N\times K+N\times q}$, where every class in the task has q test cases. Thus, the performance of a model is evaluated as the averaged accuracy on (the query set of) multiple tasks sampled from the novel classes.

3.2 Distribution Calibration

As introduced in Section 3.1, the base classes have a sufficient amount of data while the evaluation tasks sampled from the novel classes only have a limited number of labeled samples. The statistics of the distribution for the base class can be estimated more accurately compared to the estimation based on few-shot samples, which is an ill-posed problem. As shown in Table 1, we observe that if we assume the feature distribution is Gaussian, the mean and variance with respect to each class are correlated to the semantic similarity of each class. With this in mind, the statistics can be transferred from the base classes to the novel classes if we learn how similar the two classes are. In the following sections, we discuss how we calibrate the distribution estimation of the classes with only a few samples (Section 3.2.2) with the help of the statistics of the base classes (Section 3.2.1). We will also elaborate on how do we leverage the calibrated distribution to improve the performance of few-shot learning (Section 3.2.3).

Note that our distribution calibration strategy is over the feature-level and is agnostic to any feature extractor. Thus, it can be built on top of any pretrained feature extractors without further costly fine-tuning. In our experiments, we use the pretrained WideResNet following previous work [30]. The WideResNet is trained to classify the base classes, along with a self-supervised pretext task to learn the general-purpose representations suitable for image understanding tasks. Please refer to their paper for more details on training the feature extractor.

3.2.1 Statistics of the base classes

We assume the feature distribution of base classes is Gaussian. The mean of the feature vector from a base class i is calculated as the mean of every single dimension in the vector:

$$\boldsymbol{\mu}_i = \frac{\sum_{j=1}^{n_i}(\boldsymbol{x}_j)}{n_i}, \tag{1}$$

where $\boldsymbol{x}_j$ is a feature vector of the j-th sample from the base class i and n_i is the total number of samples in class i. As the feature vector $\boldsymbol{x}_j$ is multi-dimensional, we use covariance for a better representation of the variance between any pair of elements in the feature vector. The covariance matrix $\boldsymbol{\Sigma}_i$ for the features from class i is calculated as:

$$\boldsymbol{\Sigma}_i = \frac{1}{n_i - 1}\sum_{j=1}^{n_i}(\boldsymbol{x}_j - \boldsymbol{\mu}_i)(\boldsymbol{x}_j - \boldsymbol{\mu}_i)^T. \tag{2}$$

Algorithm 1: Training procedure for an N-way-K-shot task

Require: Support set features $\mathcal{S} = (\boldsymbol{x}_i, y)_{i=1}^{N\times K}$
Require: Base classes' statistics $\{\boldsymbol{\mu}_i\}_{i=1}^{|C_b|}$, $\{\boldsymbol{\Sigma}_i\}_{i=1}^{|C_b|}$
Transform $(\boldsymbol{x}_i)_{i=1}^{N\times K}$ with Tukey's Ladder of Powers as Equation 3;
for $(\boldsymbol{x}_i, y_i) \in \mathcal{S}$ **do**
  Calibrate the mean $\boldsymbol{\mu}'$ and the covariance $\boldsymbol{\Sigma}'$ for class y_i using $\boldsymbol{x}_i$ with Equation 6;
  Sample features for class y_i from the calibrated distribution as Equation 7;
end
Train a classifier using both support set features and all sampled features as Equation 8;

3.2.2 Calibrating statistics of the novel classes

Here, we consider an N-way-K-shot task sampled from the novel classes.

Tukey's Ladder of Powers Transformation

To make the feature distribution more Gaussian-like, we first transform the features of the support set and query set in the target task using Tukey's Ladder of Powers transformation [31]. Tukey's Ladder of Powers transformation is a family of power transformations which can reduce the skewness of distributions and make distributions more Gaussian-like. Tukey's Ladder of Powers transformation is formulated as:

$$\tilde{\mathbf{x}} = \begin{cases} \boldsymbol{x}^\lambda & \text{if } \lambda \neq 0 \\ \log(\boldsymbol{x}) & \text{if } \lambda = 0 \end{cases} \tag{3}$$

where λ is a hyper-parameter to adjust how to correct the distribution. The original feature can be recovered by setting λ as 1. Decreasing λ makes the distribution less positively skewed and vice versa.

Calibration through statistics transfer

Using the statistics from the base classes introduced in Section 3.2.1, we transfer the statistics from the base classes which are estimated more accurately on sufficient data to the novel classes. The transfer is based on the Euclidean distance between the feature space of the novel classes and the mean of the features from the base classes $\boldsymbol{\mu}_i$ as computed in Equation 1. Specifically, we select the top k base classes with the closest distance to the feature of a sample $\tilde{\boldsymbol{x}}$ from the support set:

$$\mathbb{S}_d = \{-\|\boldsymbol{\mu}_i - \tilde{\boldsymbol{x}}\|^2 \mid i \in C_b\}, \tag{4}$$

$$\mathbb{S}_N = \{i \mid -\|\boldsymbol{\mu}_i - \tilde{\boldsymbol{x}}\|^2 \in topk(\mathbb{S}_d)\}, \tag{5}$$

where $topk(\cdot)$ is an operator to select the top elements from the input distance set $\mathbb{S}_d$. $\mathbb{S}_N$ stores the k nearest base classes with respect to a feature vector $\tilde{\boldsymbol{x}}$. Then, the mean and covariance of the distribution is calibrated by the statistics from the nearest base classes:

$$\boldsymbol{\mu}' = \frac{\Sigma_{i\in\mathbb{S}_N}\boldsymbol{\mu}_i + \tilde{\boldsymbol{x}}}{k+1}, \boldsymbol{\Sigma}' = \frac{\Sigma_{i\in\mathbb{S}_N}\boldsymbol{\Sigma}_i}{k} + \alpha, \tag{6}$$

where α is a dispersion hyper-parameter that helps to reduce the distribution approximation error, since base class features usually have a relatively smaller intra-class variance because of sufficient training examples.

For few-shot learning with more than one shot, the aforementioned procedure of the distribution calibration should be undertaken multiple times with each time using one feature vector from the support set. This avoids the bias provided by one specific sample and potentially achieves a more diverse and accurate distribution estimation. Thus, for simplicity, we denote the calibrated distribution as a set of statistics. For a class $y \in C_n$, we denote the set of statistics as $\mathbb{S}_y = \{(\boldsymbol{\mu}_1^{'}, \boldsymbol{\Sigma}_1^{'}), ..., (\boldsymbol{\mu}_K^{'}, \boldsymbol{\Sigma}_K^{'})\}$, where $\boldsymbol{\mu}_i^{'}$, $\boldsymbol{\Sigma}_i^{'}$ are the calibrated mean and covariance, respectively, computed based on the i-th feature in the support set of class y. Here, the size of the set is the value of K for an N-way-K-shot task.

3.2.3 How to leverage the calibrated distribution?

With a set of calibrated statistics $\mathbb{S}_y$ for class y in a target task, we generate a set of feature vectors with label y by sampling from the calibrated Gaussian distributions:

$$\mathbb{D}_y = \{(\boldsymbol{x}, y) | \boldsymbol{x} \sim \mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma}), \forall (\boldsymbol{\mu}, \boldsymbol{\Sigma}) \in \mathbb{S}_y\}. \tag{7}$$

Here, the total number of generated features per class is set as a hyperparameter and they are equally distributed for every calibrated distribution in $\mathbb{S}_y$. The generated features along with the original support set features for a few-shot task are then served as the training data for a task-specific classifier. We train the classifier for a task by minimizing the cross-entropy loss over both the features of its support set $\mathcal{S}$ and the generated features $\mathbb{D}_y$:

$$\ell = \sum_{(\boldsymbol{x}, y) \sim \tilde{S} \cup \mathbb{D}_{y, y \in \mathcal{Y}^{\mathcal{T}}}} - \log \Pr(y | \boldsymbol{x}; \theta), \tag{8}$$

where $\mathcal{Y}^{\mathcal{T}}$ is the set of classes for the task $\mathcal{T}$. $\tilde{\mathcal{S}}$ denotes the support set with features transformed by Turkey's Ladder of Powers transformation and the classifier model is parameterized by θ.

4 Generalization Error Bound

We formulate the above problem in the traditional risk minimization framework [32]. The expected and empirical risks of a classifier f can be defined as

$$R(f) = E_{(X,Y) \sim \mathcal{D}}[\ell(f(X), Y)], \tag{9}$$

and

$$\hat{R}(f) = \frac{1}{N} \sum_{i=1}^{N} \ell(f(X_i), Y_i), \tag{10}$$

where N is size of training sample drawn from $\mathcal{D}$.

In our case, for a specific few-shot classification task, the distributions of the generated set, support set, and query set are denoted by $\mathcal{D}_g$, $\mathcal{D}_s$ and $\mathcal{D}_q$, respectively.

The classifier $\hat{f}$ is learned from the support set $\mathcal{S}$ and the generated set $\mathcal{G}$,

$$\hat{f} = \arg\min_{f \in \mathcal{F}} \hat{R}_{(s+g)}(f), \tag{11}$$

Thus the generalization error is defined as

$$R_q(\hat{f}) = E_{(X,Y) \sim \mathcal{D}_q}[\ell(\hat{f}(X), Y)]. \tag{12}$$

Then we have,

$$\begin{aligned} &R_q(\hat{f}) - \hat{R}_{s+g}(\hat{f}) \\ &\quad = R_q(\hat{f}) - R_s(\hat{f}) + R_s(\hat{f}) - \hat{R}_{s+g}(\hat{f}), \end{aligned} \tag{13}$$

where $R_{\#}$ and $\hat{R}_{\#}$ are the expected and empirical risks on the distribution $\mathcal{D}_{\#}$.

In this paper, we bound the generalization error by employing *Rademacher Complexity*, which is one of the most frequently used complexity measures of function classes. It can be used to derive data-dependent upper-bounds on the learnability of function classes. Intuitively, a function class with smaller Rademacher complexity is easier to learn. The following is the definition of the empirical Rademacher complexity.

Definition 4.1 ([33]). *Let $\mathcal{F}$ be a function class and $\{\mathbf{z}_n\}_{n=1}^N$ be a sample drawn from $\mathcal{Z}$. Denote $\{\sigma_n\}_{n=1}^N$ be a set of random variables independently taking either value from $\{-1, 1\}$ with equal probability. Then, the Rademacher complexity of $\mathcal{F}$ with respect to the sample is defined as*

$$\hat{\mathfrak{R}}(\mathcal{F}) := \mathrm{E}_{\sigma} \sup_{f \in \mathcal{F}} \left[\frac{1}{N} \sum_{n=1}^{N} \sigma_n f(\mathbf{z}_n) \right]. \tag{14}$$

For the first two terms in the right part of Equation 13, since query and support set are both sampled from the novel set, the distributions and thereby the expected risks are the same, i.e., $R_q(\hat{f}) - R_s(\hat{f}) = 0$.

For the next two terms in the right part of Equation 13, the gap, $R_s(\hat{f}) - \hat{R}_{(s+g)}(\hat{f})$, is caused by distribution shift. We leverage a calibrated Gaussian distribution to approximate the assumed Gaussian distribution, and thereby the ground-truth distribution of the support (novel) set, and thus there exists a distribution shift error, consisting of a distribution assumption error and a distribution approximation error.

Empirically, for $\hat{R}_{(s+g)}$, given a $\tau \in [0, 1)$, we consider the convex combination of the support risk and the generated risk:

$$\hat{R}_{s+g}(\hat{f}) = \tau \hat{R}_s(\hat{f}) + (1 - \tau) \hat{R}_g(\hat{f}). \tag{15}$$

As discussed before [34], [35], setting τ introduces a trade-off between the support set that is reliable but not sufficient and the generated set that is sufficient but not reliable. Setting $\tau = 0.5$ means that we treat the generated set equally as the support set.

Then, based on [36], assuming that the neural network has d layers with parameter matrices $W_1, \ldots, W_d$, and the activation functions $\sigma_1, \ldots, \sigma_{d-1}$ are Lipschitz continuous, satisfying $\sigma_j(0) = 0$. We denote by $h : X \mapsto W_d \sigma_{d-1}(W_{d-1} \sigma_{d-2}(\ldots \sigma_1(W_1 X))) \in \mathbb{R}$ the standard form of the neural network. $H = \arg\max_{i \in \{1, \ldots, c\}} h_i$. Then the output of the softmax function is defined as $f_i(X) = \exp(h_i(X)) / \sum_{j=1}^{c} \exp(h_j(X)), i = 1, \ldots, c$, $R_s(\hat{f}) - \hat{R}_{(s+g)}(\hat{f})$ can be bounded as follows:

Theorem 4.1. *Assume that $\mathcal{F}$ is a function class consisting of functions with the range $[a, b]$. Let $\mathbf{Z}_1^{N_s} = \{\mathbf{z}_n^{(s)}\}_{n=1}^{N_s}$ and $\mathbf{Z}_1^{N_g} = \{\mathbf{z}_n^{(g)}\}_{n=1}^{N_g}$ be two sets of i.i.d. samples drawn from the support (novel) domain $\mathcal{Z}^{(s)}$ and the generated calibrated domain $\mathcal{Z}^{(g)}$, respectively; the Frobenius norm of the weight matrices $W_1, \ldots, W_d$ are at most $M_1, \ldots, M_d$. Let the activation functions*

be 1-Lipschitz, positive-homogeneous, and applied element-wise (such as the ReLU). Let $\mathbf{z}$ *is upper bounded by* B*, i.e., for any* $\mathbf{z}$*,* $\|\mathbf{z}\| \leqslant B$*. Then, given* $\tau \in [0,1)$ *and for any* $\delta > 0$*, with probability at least* $1-\delta$*,*

$$
\begin{aligned}
& R_q(f) - \hat{R}_{s+g}(f) \\
\leqslant\ & (1-\tau) D_{\mathcal{F}}(\mathcal{S},\mathcal{G}) + 3(1-\tau)\sqrt{\frac{(b-a)\ln(4/\delta)}{2N_g}} \\
& + 2(1-\tau)\hat{\mathfrak{R}}_g(\mathcal{F}) + 3\tau\sqrt{\frac{(b-a)\ln(4/\delta)}{2N_s}} \\
& + 2\tau\hat{\mathfrak{R}}_s(\mathcal{F}) + \sqrt{\frac{(b-a)^2\ln(4/\delta)}{2}\left(\frac{\tau^2}{N_s} + \frac{(1-\tau)^2}{N_g}\right)} \\
\leqslant\ & (1-\tau) D_{\mathcal{F}}(\mathcal{S},\mathcal{N}) + (1-\tau) D_{\mathcal{F}}(\mathcal{N},\mathcal{G}) \\
& + 2(1-\tau)\frac{cB(\sqrt{2d\log 2}+1)\Pi_{i=1}^{d} M_i}{\sqrt{N_g}} \\
& + 3\tau\sqrt{\frac{(b-a)\ln(4/\delta)}{2N_s}} + 3(1-\tau)\sqrt{\frac{(b-a)\ln(4/\delta)}{2N_g}} \\
& + 2\tau\frac{cB(\sqrt{2d\log 2}+1)\Pi_{i=1}^{d} M_i}{\sqrt{N_s}} \\
& + \sqrt{\frac{(b-a)^2\ln(4/\delta)}{2}\left(\frac{\tau^2}{N_s} + \frac{(1-\tau)^2}{N_g}\right)}. \qquad (16)
\end{aligned}
$$

where $D_{\mathcal{F}}(\Delta,\Diamond) \triangleq \sup_{f\in\mathcal{F}} |R_{\Delta}(f) - R_{\Diamond}(f)|$*, and* $\mathcal{N}$ *is the calibrated Gaussian distribution.*

Proof is provided in Section 7.1.

Theorem 4.1 shows that the generalization error is bounded by the empirical training risk, the *distribution assumption error*, the *distribution approximation error*, and the *estimation error*. The empirical training risk can be minimized to an arbitrary small value. The distribution assumption error $D_{\mathcal{F}}(\mathcal{S},\mathcal{N})$ is the gap between the ground-truth feature representation distribution and the assumed Gaussian distribution. If the ground-truth feature representation distribution is Gaussian, this term will be 0, which motivates us to do the Tukey's Ladder of Powers transformation to make it more Gaussian-like. The distribution approximation error $D_{\mathcal{F}}(\mathcal{N},\mathcal{G})$ is the gap between the assumed Gaussian distribution and the approximated (calibrated) Gaussian distribution. If the statistics, i.e., the mean and variance, of the approximated calibrated Gaussian are the same as that of the assumed Gaussian, this term will be 0, which motivates us to select the statistics of the most k similar classes from the rich base classes. The estimation error (the rest terms in Equation 16) is caused by the finite samples, which asymptotically tends to 0 as the sample size tends to infinity. We empirically verify the theoretical analysis in Section 5.5.2.

5 Experiments

In this section, we answer the following questions:

- How does our distribution calibration strategy perform compared to the state-of-the-art methods?
- What does the calibrated distribution look like? Is it an accurate approximation for this class?
- How does Tukey's Ladder of Power transformation interact with the feature generations? How important is each in relation to performance?

5.1 Experimental Setup

5.1.1 Datasets

We evaluate our distribution calibration strategy on *mini*ImageNet [37], *tiered*ImageNet [5], and CUB [55]. *mini*ImageNet and *tiered*ImageNet have a brand range of classes including various animals and objects while CUB is a more fine-grained dataset that includes various species of birds. Datasets with different levels of granularity may have different distributions for their feature space. We want to show the effectiveness and generality of our strategy on all three datasets.

***mini*ImageNet** is derived from ILSVRC-12 dataset [56]. It contains 100 diverse classes with 600 samples per class. The image size is $84\times84\times3$. We follow the splits used in previous works [37], which split the dataset into 64 base classes, 16 validation classes, and 20 novel classes.

***tiered*ImageNet** is a larger subset of ILSVRC-12 dataset [56], which contains 608 classes sampled from hierarchical category structure. Each class belongs to one of 34 higher-level categories sampled from the high-level nodes in the ImageNet. The average number of images in each class is 1281. We use 351, 97, and 160 classes for training, validation, and test, respectively.

CUB is a fine-grained few-shot classification benchmark. It contains 200 different classes of birds with a total of 11,788 images of size $84\times84\times3$. Following previous works [57], we split the dataset into 100 base classes, 50 validation classes, and 50 novel classes.

5.1.2 Evaluation Metric

We use the top-1 accuracy as the evaluation metric to measure the performance of our method. We report the accuracy on 5way1shot and 5way5shot settings for *mini*ImageNet, *tiered*ImageNet, and CUB. The reported results are the averaged classification accuracy over 10,000 tasks.

5.1.3 Implementation Details

For feature extractor, we use the WideResNet trained following previous work [30]. For each dataset, we train the feature extractor with base classes and test the performance using novel classes. Note that the feature representation is extracted from the penultimate layer (with a ReLU activation function) from the feature extractor, thus the values are all non-negative so that the inputs to Tukey's Ladder of Powers transformation in Equation 3 are valid. At the distribution calibration stage, we compute the base class statistics and transfer them to calibrate novel class distribution for each dataset. We use the LR and SVM [58] implementation of scikit-learn [59] with the default settings. We use the same hyperparameter value for all datasets except for α. Specifically, the number of generated features is 750; the number of retrieved base classes k is 2, and the power of Tukey's transformation λ is 0.5 for all experiments. We set the dispersion parameter α as 0.21, 0.21 and 0.3 for miniImageNet, tieredImageNet, and CUB, respectively.

5.2 Comparison to State-of-the-art

Table 2, Table 3 and Table 4 present the 5way1shot and 5way5shot classification results of our method on *mini*ImageNet, CUB and *tiered*ImageNet. We compare our method distribution calibration (DC) with the three groups of the few-shot learning method, *optimization-based*, *metric-based*, and *generation-based*. Our method can be built on top of any classifier, and we use three popular and simple classifiers, namely Maximum Likelihood,

Few-shot Learning Method		miniImageNet	
		1-shot	5-shot
Optimization-based	MAML [1]	48.70 ± 1.75	63.11 ± 0.92
	Meta-SGD [9]	50.47 ± 1.87	64.03 ± 0.94
	Meta-LSTM [37]	43.56 ± 0.84	60.60 ± 0.71
	Hierarchical Bayes [38]	49.40 ± 1.83	–
	Bilevel Programming [39]	50.54 ± 0.85	64.53 ± 0.68
	adaResNet [40]	56.88 ± 0.62	71.94 ± 0.57
	MetaOptNet [41]	62.64 ± 0.35	78.63 ± 0.68
	LEO [42]	61.67 ± 0.08	77.59 ± 0.12
	Meta Transfer Learning [43]	64.3 ± 1.7	80.9 ± 0.8
	CTM [44]	64.12 ± 0.82	80.51 ± 0.13
	E3BM [45]	63.80 ± 0.40	80.29 ± 0.25
Metric-based	MatchingNets [17]	43.44 ± 0.77	55.31 ± 0.73
	ProtoNets [2]	49.42 ± 0.78	68.20 ± 0.66
	RelationNets [46]	50.44 ± 0.82	65.32 ± 0.70
	Graph neural network [47]	50.33 ± 0.36	66.41 ± 0.63
	EGNN [48]	59.63 ± 0.52	76.34 ± 0.48
	Ridge regression [49]	51.9 ± 0.2	68.7± 0.2
	TransductiveProp [50]	55.51	69.86
	Variational Few-shot [24]	61.23 ± 0.26	77.69 ± 0.17
	Negative-Cosine [51]	62.33 ± 0.82	80.94 ± 0.59
Generation-based	MetaGAN [20]	52.71 ± 0.64	68.63 ± 0.67
	Delta-Encoder [22]	59.9	69.7
	TriNet [21]	58.12 ± 1.37	76.92 ± 0.69
	Meta Variance Transfer [29]	-	67.67 ± 0.70
Ours	Maximum Likelihood with DC (Ours)	66.91 ± 0.17	80.74 ± 0.48
	SVM with DC (Ours)	**67.31 ± 0.83**	**82.30 ± 0.34**
	Logistic Regression with DC (Ours)	**68.57 ± 0.55**	**82.88 ± 0.42**

TABLE 2: 5way1shot and 5way5shot classification accuracy (%) on *mini*ImageNet with 95% confidence intervals. The numbers in **bold** have intersecting confidence intervals with the most accurate method.

Few-shot Learning Method		CUB	
		1-shot	5-shot
Optimization-based	MAML [1]	50.45 ± 0.97	59.60 ± 0.84
	Meta-SGD [9]	53.34 ± 0.97	67.59 ± 0.82
Metric-based	MatchingNets [17]	56.53 ± 0.99	63.54 ± 0.85
	ProtoNets [2]	72.99 ± 0.88	86.64 ± 0.51
	Negative-Cosine [51]	72.66 ± 0.85	89.40 ± 0.43
Generation-based	Delta-Encoder [22]	69.8	82.6
	TriNet [21]	69.61 ± 0.46	84.10 ± 0.35
	Meta Variance Transfer [29]	-	80.33 ± 0.61
Ours	Maximum Likelihood with DC (Ours)	77.22 ± 0.14	89.58 ± 0.27
	SVM with DC (Ours)	**79.49 ± 0.33**	**90.26 ± 0.98**
	Logistic Regression with DC (Ours)	**79.56 ± 0.87**	**90.67 ± 0.35**

TABLE 3: 5way1shot and 5way5shot classification accuracy (%) on CUB with 95% confidence intervals. The numbers in **bold** have intersecting confidence intervals with the most accurate method.

SVM, and LR to prove the effectiveness of our method. Simple linear classifiers equipped with our method perform better than the state-of-the-art few-shot classification method and achieve the best performance on 1-shot and 5-shot settings of *mini*ImageNet, *tiered*ImageNet, and CUB. The performance of our distribution calibration surpasses the state-of-the-art *generation-based* method by 10% for the 5way1shot setting, which proves that our method can handle extremely low-shot classification tasks better. Compared to other *generation-based* methods, which require the design of a generative model with extra training costs on the learnable parameters, a simple machine learning classifier with DC is much more simple, effective, and flexible and can be equipped with any feature extractors and classifier model structures. Specifically, we show three variants, i.e, Maximum likelihood with DC, SVM with

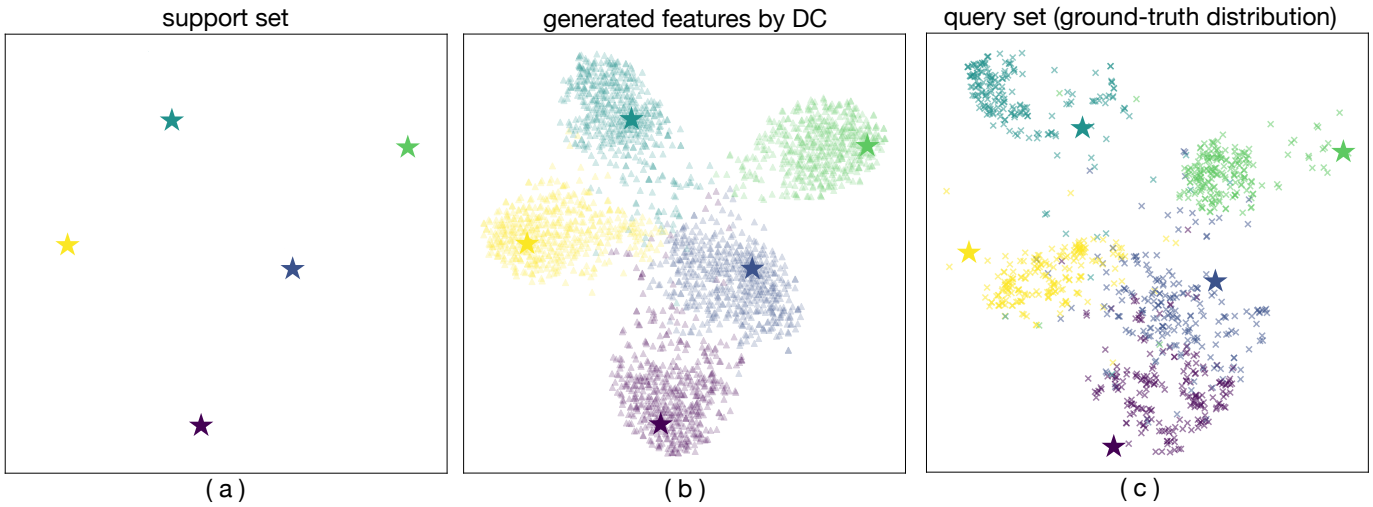


Fig. 2: t-SNE visualization of our distribution estimation. Different colors represent different classes. '★' represents support set features, 'x' in figure (c) represents query set features, '▲' in figure (b) represents generated features.

Few-shot Learning Method		tieredImagenet	
		1-shot	5-shot
Optimization-based	MAML [1] (by [50])	51.67 ± 1.81	70.30 ± 1.75
	LEO [42]	66.33 ± 0.05	81.44 ± 0.09
	CTM [52]	68.41 ± 0.39	84.28 ± 1.73
	Meta Transfer Learning [53]	72.00 ± 1.80	85.10 ± 0.80
	E3BM, [45]	71.20 ± 0.40	85.30 ± 0.30
Metric-based	ProtoNets [2] (by [5])	53.31 ± 0.89	72.69 ± 0.74
	RelationNets [46] (by [50])	54.48 ± 0.93	71.32 ± 0.78
	TransductiveProp [50]	57.41 ± 0.94	71.55 ± 0.74
	DeepEMD [54]	71.16 ± 0.87	86.03 ± 0.58
Ours	Maximum Likelihood with DC (Ours)	75.92 ± 0.60	87.84 ± 0.65
	SVM with DC (Ours)	**77.93 ± 0.12**	**89.72 ± 0.37**
	Logistic Regression with DC (Ours)	**78.19 ± 0.25**	**89.90 ± 0.41**

TABLE 4: 5way1shot and 5way5shot classification accuracy (%) on tieredImagenet with 95% confidence intervals. The numbers in **bold** have intersecting confidence intervals with the most accurate method.

DC, Logistic Regression with DC in Table 2, Table 3 and Table 4. A simple maximum likelihood classifier based on the calibrated distribution can outperform previous baselines and training an SVM classifier or Logistic Regression classifier using the samples from the calibrated distribution can further improve the performance.

5.3 Visualization of Generated Samples

We show what the calibrated distribution looks like by visualizing the generated features sampled from the distribution. In Figure 2, we show the t-SNE representation [60] of the original support set (a), the generated features (b) as well as the query set (c). Based on the calibrated distribution, the sampled features form a Gaussian distribution. Due to the limited number of examples in the support set, only 1 in this case, the samples from the query set usually cover a greater area and are a mismatch with the support set. This mismatch can be fixed to some extent by the generated features, i.e., the generated features in (b) can overlap areas of the query set. Thus, training with these generated features can alleviate the mismatch between the distribution estimated only from the few-shot samples and the ground-truth distribution.

5.4 Applicability of distribution calibration

Our distribution calibration strategy is agnostic to backbones / classifiers. Table 6 shows the consistent performance boost when applying distribution calibration on different backbones, i.e, four convolutional layers (Conv4), six convolutional layers (Conv6), ResNet10 [61], ResNet18 [61], WRN28 [62] and WRN28 trained with rotation loss [30] and on different classifiers, i.e, logistic regression and support vector machine [58]. Distribution calibration achieves around 10% accuracy improvement compared to the backbones trained with different baselines.

5.5 Ablation Study and Theoretical Analysis Verification

5.5.1 Ablation Study

Table 5 shows the effect of distribution assumption, the performance when our model is trained without Tukey's Ladder of Powers

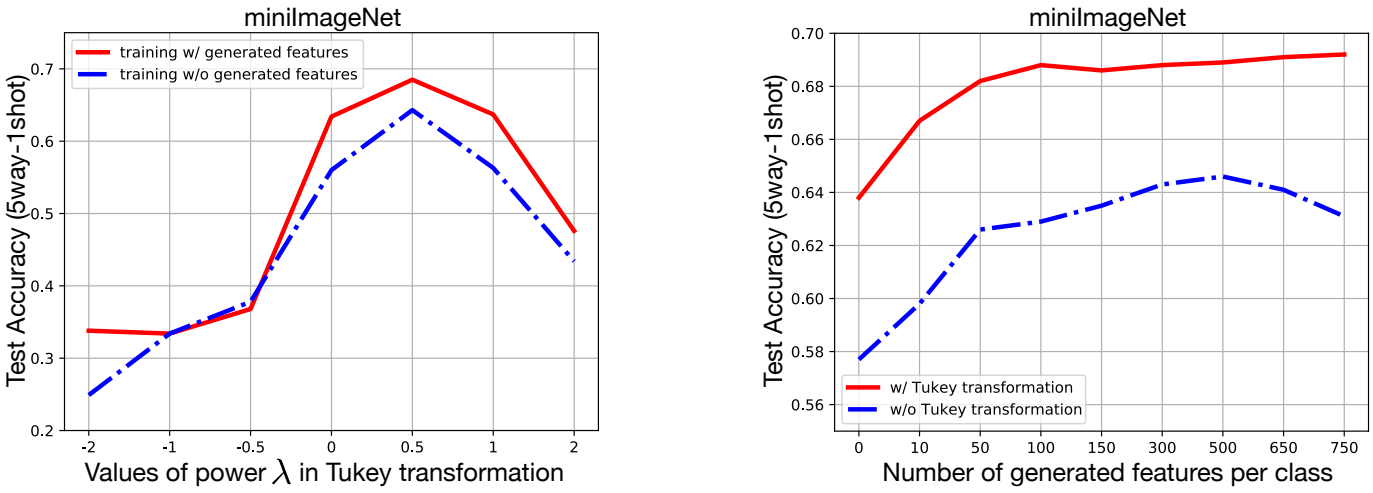


Fig. 3: Left: Accuracy when increasing the power in Tukey's transformation when training with (red) or without (blue) the generated features. Right: Accuracy when increasing the number of generated features with the features are transformed by Tukey's transformation (red) and without Tukey's transformation (blue).

Distribution assumption	Tukey transformation	Training with generated features	*mini*ImageNet	
			5way1shot	5way5shot
None	✗	✗	56.37 ± 0.68	79.03 ± 0.51
Gaussian	✗	✓	63.70 ± 0.38	82.26 ± 0.73
Laplacian	✗	✓	62.39 ± 0.17	81.96 ± 0.22
Multimodal	✗	✓	61.45 ± 0.33	80.73 ± 0.49
Gaussian	✓	✗	64.30 ± 0.53	81.33 ± 0.35
Gaussian	✓	✓	$\mathbf{68.57 \pm 0.55}$	$\mathbf{82.88 \pm 0.42}$

TABLE 5: Ablation study on *mini*ImageNet 5way1shot and 5way5shot showing accuracy (%) with 95% confidence intervals.

Backbones	Classifiers	without DC	with DC
Conv4	Logistic Regression	42.11 ± 0.71	$\mathbf{54.62 \pm 0.64}$ ($\uparrow$ **12.51**)
Conv4	Support Vector Machine	41.24 ± 0.75	$\mathbf{54.24 \pm 0.37}$ ($\uparrow$ **13.00**)
Conv6	Logistic Regression	46.07 ± 0.26	$\mathbf{57.14 \pm 0.45}$ ($\uparrow$ **11.07**)
Conv6	Support Vector Machine	46.03 ± 0.17	$\mathbf{57.33 \pm 0.54}$ ($\uparrow$ **11.30**)
ResNet10 [61]	Logistic Regression	53.17 ± 0.31	$\mathbf{64.41 \pm 0.33}$ ($\uparrow$ **11.24**)
ResNet10 [61]	Support Vector Machine	54.01 ± 0.71	$\mathbf{64.03 \pm 0.11}$ ($\uparrow$ **10.02**)
ResNet18 [61]	Logistic Regression	52.32 ± 0.82	$\mathbf{61.50 \pm 0.47}$ ($\uparrow$ **9.180**)
ResNet18 [61]	Support Vector Machine	51.41 ± 0.27	$\mathbf{60.03 \pm 0.19}$ ($\uparrow$ **8.620**)
WRN28 [62]	Logistic Regression	54.53 ± 0.56	$\mathbf{64.38 \pm 0.63}$ ($\uparrow$ **9.850**)
WRN28 [62]	Support Vector Machine	53.27 ± 0.55	$\mathbf{63.38 \pm 0.10}$ ($\uparrow$ **10.11**)
WRN28 + Rotation Loss [30]	Support Vector Machine	54.27 ± 0.68	$\mathbf{67.31 \pm 0.83}$ ($\uparrow$ **13.04**)
WRN28 + Rotation Loss [30]	Logistic Regression	56.37 ± 0.68	$\mathbf{68.57 \pm 0.55}$ ($\uparrow$ **12.20**)

TABLE 6: 5way1shot classification accuracy (%) on *mini*ImageNet with different backbones and classifiers.

transformation for the features as in Equation 3, and when it is trained without the generated features as in Equation 7. A better distribution assumption helps to reduce the generalization error bound. Empirically, Gaussian assumption performs slightly better than Laplacian assumption and Multimodal assumption.

Figure 4 shows how Tukey's transformation helps improve the classifier. The extracted base class features are ideally from Gaussian distributions since the backbone network was trained over the base classes. However, the unseen (novel) class feature distributions are relatively skewer. We apply Tukey's transformation to calibrate the novel class feature distributions to be more Gaussian, which is aligned with our Gaussian assumption.

5.5.2 *Theoretical Analysis Verification*

As discussed in Theorem 4.1, the generalization error of the proposed distribution-calibration-based few-shot learning is bounded by the distribution assumption error $D_{\mathcal{F}}(\mathcal{S}, \mathcal{N})$, the distribution approximation error $D_{\mathcal{F}}(\mathcal{N}, \mathcal{G})$ and the estimation error. We empirically verify the theoretical analysis in the following paragraphs.

The distribution assumption error. The distribution assumption error $D_{\mathcal{F}}(\mathcal{S}, \mathcal{N})$ measures the discrepancy between the ground-truth feature representation distribution and the assumed Gaussian distribution. A better distribution assumption leads to better generalization ability. Based on the fact that the CNN extracted image features from the same class are often clus-

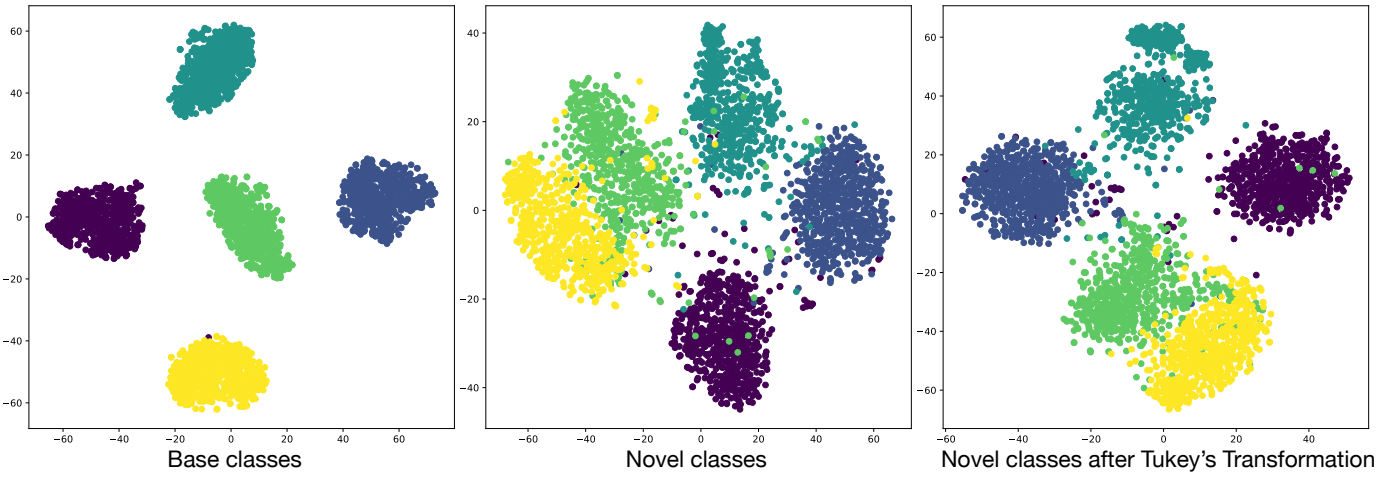


Fig. 4: The left shows t-SNE [60] visualization of feature distributions of 5 randomly selected base classes. The middle and the right show the feature distributions of 5 randomly selected novel classes before Tukey's transformation and after Tukey's transformation, respectively.

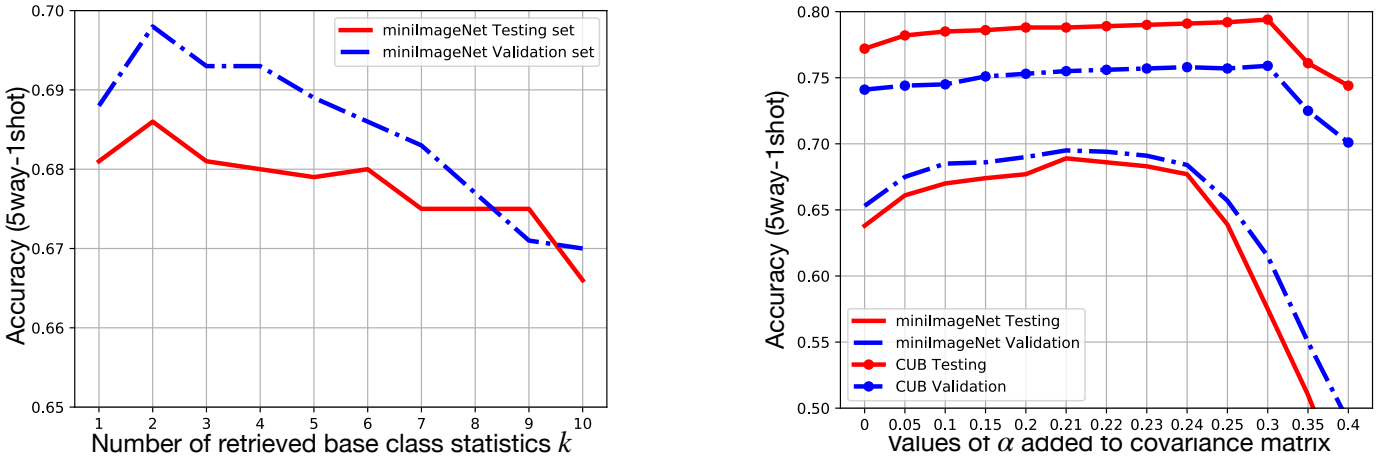


Fig. 5: The effect of different values of k and α.

tered well, as visualized in Figure 4, we choose the Gaussian distribution as our assumed distribution. To verify the rationality of the Gaussian assumption, we also conduct experiments on Laplacian and Multimodal distribution. As shown in Table 5, the Gaussian distribution assumption brings better performance than others. To further close the gap between the ground-truth feature representation distribution and the assumed Gaussian distribution, we then apply Tukey's transformation on the ground-truth feature representation distribution to make it more Gaussian-like.

The left side of Figure 3 shows the 5way1shot accuracy when choosing different powers for the Tukey's transformation in Equation 3 when training the classifier with the generated features (red) and without (blue). Note that when the power λ equals 1, the transformation keeps the original feature representations. There is a consistent general tendency for training with and without the generated features and in both cases, we found $\lambda = 0.5$ is the optimum choice. With Tukey's transformation, the distribution of features in target tasks becomes more aligned to the assumed Gaussian distribution, and thus the distribution assumption error becomes smaller, benefiting the classifier which is trained on features sampled from the calibrated distribution.

The distribution approximation error. The distribution approximation error $D_{\mathcal{F}}(\mathcal{N}, \mathcal{G})$ comes from the inaccurate distribution approximation (calibration in our case) of the assumed distribution. In our distribution calibration, we utilize k base class statistics to calibrate the novel class distribution in Equation 5. The α in Equation 6 is a constant added on each element of the estimated covariance matrix, which can determine the degree of dispersion of features sampled from the calibrated distributions. Both k and α affect the distribution approximation error. Figure 5 shows the effect of different values of k and α. We observe that in each dataset, the performance of the validation set and the novel (testing) set generally has the same tendency, which indicates that the choice of hyper-parameters is dataset-dependent and is not overfitting to a specific set.

The estimation error. The right side of Figure 3 analyzes whether more generated features results in consistent improvement in both cases, namely when the features of support and query set are transformed by Tukey's transformation (red) and when they are not (blue). We found that when the number of generated features is below 500, both cases can benefit from more generated features, which corresponds to the estimation error asymptotically tending to 0 as the sample size tends to infinity. However, when more features are sampled, the performance of the classifier

tested on untransformed features begins to decline. This is caused by the inconsistency of the ground-truth distribution and the assumed distribution. Besides, the performances of 5shot are consistently better than that of 1shot in Table 2, Table 3 and Table 4, corresponding that larger N_s leads to smaller estimation error.

6 Conclusion

In this paper, a simple but effective distribution calibration strategy for few-shot learning is proposed. Compared to other generative-based methods, the proposed strategy doesn't involve any complex generative models and extra learnable parameters. A simple linear classifier trained with features generated by our strategy outperforms the current state-of-the-art methods by $\sim 5\%$ on *mini*ImageNet. The calibrated distribution is visualized and demonstrates an accurate estimation of the feature distribution. The established generalization error bound also identifies that the proposed method is promising to bridge the gap between few-shot learning and many-shot learning as it eliminates the distribution assumption error and the distribution approximation error. The theoretical framework also provides an insight to guide the future few-shot learning method.

7 Proofs

7.1 Proof of Theorem 4.1

First we introduce the basic generalization error bound (see [63] Theorem 5) with Rademacher complexity:

Lemma 7.1. *Let $\mathcal{F} \subseteq [a,b]$. For any $\delta > 0$, with probability at least $1-\delta$, there holds that for any $f \in \mathcal{F}$*

$$R(f) \leqslant \hat{R}(f) + 2\mathfrak{R}(\mathcal{F}) + \sqrt{\frac{(b-a)\ln(1/\delta)}{2N}} \tag{17}$$

$$\leqslant \hat{R}(f) + 2\hat{\mathfrak{R}}(\mathcal{F}) + 3\sqrt{\frac{(b-a)\ln(2/\delta)}{2N}} \tag{18}$$

Then we introduce the extended McDiamid's Inequality (see [36] Theorem C.2):

Lemma 7.2. *Given independent domains $\mathcal{Z}^{(S_k)}(1 \leqslant k \leqslant K)$, for any $1 \leqslant k \leqslant K$, let $\mathbf{Z}_1^{N_k} := \left\{\mathbf{z}_n^{(S_k)}\right\}_{n=1}^{N_k}$ be N_k independent random variables taking values from the domain $\mathcal{Z}^{(S_k)}$. Assume that the function $H : \left(\mathcal{Z}^{(S_1)}\right)^{N_1} \times \cdots \times \left(\mathcal{Z}^{(S_K)}\right)^{N_K} \to \mathbb{R}$ satisfies the condition of bounded difference: for all $1 \leqslant k \leqslant K$ and $1 \leqslant n \leqslant N_k$,*

$$\sup_{\mathbf{Z}_1^{N_1},\cdots,\mathbf{Z}_1^{N_K},\mathbf{z}_n^{(S_k)}} |\, \mathcal{H} - \mathcal{H}' \,| \leqslant c_n^{(k)}, \tag{19}$$

where

$$\mathcal{H} = H(\mathbf{Z}_1^{N_1},\cdots,\mathbf{Z}_1^{N_{k-1}},\mathbf{z}_1^{(S_k)},\cdots,\mathbf{z}_n^{(S_k)},\cdots \cdots,\mathbf{z}_{N_k}^{(S_k)},\mathbf{Z}_1^{N_{k+1}},\cdots,\mathbf{Z}_1^{N_K}),$$

$$\mathcal{H}' = H(\mathbf{Z}_1^{N_1},\cdots,\mathbf{Z}_1^{N_{k-1}},\mathbf{z}_1^{(S_k)},\cdots,\mathbf{z}_n'^{(S_k)},\cdots \cdots,\mathbf{z}_{N_k}^{(S_k)},\mathbf{Z}_1^{N_{k+1}},\cdots,\mathbf{Z}_1^{N_K}).$$

Then, for any $\xi > 0$

$$\Pr\left\{H\left(\mathbf{Z}_1^{N_1},\cdots,\mathbf{Z}_1^{N_K}\right) - \mathrm{E}\left\{H\left(\mathbf{Z}_1^{N_1},\cdots,\mathbf{Z}_1^{N_K}\right)\right\} \geqslant \xi\right\}$$
$$\leqslant \exp\left\{-2\xi^2 / \sum_{k=1}^{K}\sum_{n=1}^{N_k}\left(c_n^{(k)}\right)^2\right\} \tag{20}$$

Let

$$H(\mathbf{Z}_1^{N_s},\mathbf{Z}_1^{N_g}) = \sup_{f\in\mathcal{F}} |\, \hat{R}_{(s+g)}(f) - R_s(f) \,|\,. \tag{21}$$

Then by Equation 15, we have

$$H(\mathbf{Z}_1^{N_s},\mathbf{Z}_1^{N_g}) = \sup_{f\in\mathcal{F}}\left|\tau\hat{R}_s(f) + (1-\tau)\hat{R}_g(f) - R_s(f)\right|. \tag{22}$$

It is obvious that such $H(\mathbf{Z}_1^{N_s},\mathbf{Z}_1^{N_g})$ satisfies the condition of bounded difference with

$$c_n^{(s)} = \frac{(b-a)\tau}{N_s}, c_n^{(g)} = \frac{(b-a)(1-\tau)}{N_g}$$

According to Lemma 7.2, we have for any $\xi > 0$,

$$\Pr\left\{H\left(\mathbf{Z}_1^{N_s},\cdots,\mathbf{Z}_1^{N_g}\right) - \mathrm{E}\left\{H\left(\mathbf{Z}_1^{N_s},\cdots,\mathbf{Z}_1^{N_g}\right)\right\} \geqslant \xi\right\}$$
$$\leqslant \exp\left\{\frac{-2\xi^2}{(b-a)^2\left(\frac{\tau^2}{N_s} + \frac{(1-\tau)^2}{N_g}\right)}\right\}. \tag{23}$$

Equivalently, with probability at least $1-(\delta/4)$,

$$\begin{aligned}
&H(\mathbf{Z}_1^{N_s},\mathbf{Z}_1^{N_g})\\
&\leqslant \mathrm{E}\left\{H(\mathbf{Z}_1^{N_s},\mathbf{Z}_1^{N_g})\right\}\\
&\quad + \sqrt{\frac{(b-a)^2\ln(4/\delta)}{2}\left(\frac{\tau^2}{N_s} + \frac{(1-\tau)^2}{N_g}\right)}\\
&\leqslant \tau\sup_{f\in\mathcal{F}}\left|\hat{R}_s(f) - R_s(f)\right| + (1-\tau)\mathrm{E}\left\{\sup_{f\in\mathcal{F}}\left|\hat{R}_g(f) - R_s(f)\right|\right\}\\
&\quad + \sqrt{\frac{(b-a)^2\ln(4/\delta)}{2}\left(\frac{\tau^2}{N_s} + \frac{(1-\tau)^2}{N_g}\right)}\\
&= \tau\sup_{f\in\mathcal{F}}\left|\hat{R}_s(f) - R_s(f)\right|\\
&\quad + (1-\tau)\mathrm{E}\left\{\sup_{f\in\mathcal{F}}\left|\hat{R}_g(f) - R_g(f) + R_g(f) - R_s(f)\right|\right\}\\
&\quad + \sqrt{\frac{(b-a)^2\ln(4/\delta)}{2}\left(\frac{\tau^2}{N_s} + \frac{(1-\tau)^2}{N_g}\right)}\\
&\leqslant \tau\sup_{f\in\mathcal{F}}\left|\hat{R}_s(f) - R_s(f)\right|\\
&\quad + (1-\tau)\mathrm{E}\left\{\sup_{f\in\mathcal{F}}\left|\hat{R}_g(f) - R_g(f))\right|\right\}\\
&\quad + (1-\tau)\sup_{f\in\mathcal{F}}|R_g(f) - R_s(f)|\\
&\quad + \sqrt{\frac{(b-a)^2\ln(4/\delta)}{2}\left(\frac{\tau^2}{N_s} + \frac{(1-\tau)^2}{N_g}\right)}
\end{aligned} \tag{24}$$

The quantity $\sup_{f\in\mathcal{F}}\left|\hat{R}_s(f) - R_s(f)\right|$ is termed as $D_{\mathcal{F}}(\mathcal{S},\mathcal{G})$ [36], which measures the difference of two domains. Note that $D_{\mathcal{F}}(\mathcal{S},\mathcal{G}) \leqslant D_{\mathcal{F}}(\mathcal{S},\mathcal{N}) + D_{\mathcal{F}}(\mathcal{N},\mathcal{G})$ due to the triangle inequality, where $\mathcal{N}$ is the calibrated Gaussian distribution.

According to Lemma 7.1, with probability $1-\delta/2$ the following holds:

$$\sup_{f\in\mathcal{F}}\left|\hat{R}_s(f) - R_s(f)\right| \leqslant 2\hat{\mathfrak{R}}_s(\mathcal{F}) + 3\sqrt{\frac{(b-a)\ln(4/\delta)}{2N_s}} \tag{25}$$

Then, according to Definition 4.1, we have

$$\begin{aligned}
&\mathrm{E}\left\{\sup_{f\in\mathcal{F}}\left|\hat{R}_g(f)-R_g(f))\right|\right\}\\
&=\mathrm{E}\sup_{f\in\mathcal{F}}\left|\hat{R}_g(f)-\mathrm{E}'\hat{R}'_g(f))\right|\\
&\leqslant \mathrm{E}\mathrm{E}'\sup_{f\in\mathcal{F}}\left|\hat{R}_g(f)-\hat{R}'_g(f))\right|\\
&=\mathrm{E}\mathrm{E}'\sup_{f\in\mathcal{F}}\left|\frac{1}{N_g}\sum_{n=1}^{N_g}\left(f\left(\mathbf{z}_n^{(g)}\right)-f\left(\mathbf{z}'^{(g)}_n\right)\right)\right|\\
&=\mathrm{E}\mathrm{E}'\mathrm{E}_\sigma\sup_{f\in\mathcal{F}}\left|\frac{1}{N_g}\sum_{n=1}^{N_g}\sigma_n\left(f\left(\mathbf{z}_n^{(g)}\right)-f\left(\mathbf{z}'^{(g)}_n\right)\right)\right|\\
&\leqslant 2\mathrm{E}\mathrm{E}_\sigma\sup_{f\in\mathcal{F}}\left|\frac{1}{N_g}\sum_{n=1}^{N_g}\sigma_n f\left(\mathbf{z}_n^{(g)}\right)\right|\\
&=2\mathfrak{R}_g(\mathcal{F})
\end{aligned}\tag{26}$$

Then, using again McDiarmid's inequality, with at least probability $1-\delta/3$ the following holds:

$$\mathrm{E}\left\{\sup_{f\in\mathcal{F}}\left|\hat{R}_g(f)-R_g(f))\right|\right\}\leqslant 2\hat{\mathfrak{R}}_g(\mathcal{F})+3\sqrt{\frac{(b-a)\ln(4/\delta)}{2N_g}}\tag{27}$$

Before further bounding the Rademacher complexity $\hat{\mathfrak{R}}(\mathcal{F})$, we discuss the *Lipschitz continuity* of the loss function (cross-entropy loss) w.r.t $h_k(X), k=\{1,\ldots,c\}$.

Recall that

$$\begin{aligned}
\ell(f(X),Y)&=-\sum_{i=1}^{c}1_{\{Y=i\}}\log(f_i(X))\\
&=-\log\left(\frac{\exp(h_Y(X))}{\sum_{i=1}^{c}\exp(h_i(X))}\right).
\end{aligned}\tag{28}$$

Taking the derivative of $\ell(f(X),Y)$ w.r.t. $h_i(X)$, if $i\neq Y$, we have

$$\frac{\partial\ell(f(X),Y)}{\partial h_i(X)}=\frac{\exp(h_i(X))}{\sum_{i=1}^{c}\exp(h_i(X))};\tag{29}$$

if $i=Y$, we have

$$\frac{\partial\ell(f(X),Y)}{\partial h_i(X)}=-1+\frac{\exp(h_i(X))}{\sum_{i=1}^{c}\exp(h_i(X))}.\tag{30}$$

According to Equation 29 and 30, it is clear that $-1\leqslant\frac{\partial\ell(f(X),\bar{Y})}{\partial h_i(X)}\leqslant 1$, indicating the loss function is 1-Lipschitz with respect to $h_i(X),\forall i\in\{1,\ldots,c\}$.

Lemma 7.3. *Assume that loss function $\ell(f(X_i),Y))$ is 1-Lipschitz with respect to $h_k(X_i), k=\{1,\ldots,c\}$, we have*

$$\begin{aligned}
\hat{\mathfrak{R}}(\mathcal{F})&=E\left[\sup_{f\in\mathcal{F}}\frac{1}{N}\sum_{i=1}^{N}\sigma_i\ell(f(X_i),Y_i)\right]\\
&\leqslant cE\left[\sup_{h\in H}\frac{1}{N}\sum_{i=1}^{N}\sigma_i h(X_i)\right],
\end{aligned}\tag{31}$$

where H is the function class induced by the deep neural network.

Proof is provided in Section 7.2.

Based on Lemma 7.3, we can bound the Rademacher complexity $\hat{\mathfrak{R}}(\mathcal{F})$ with the following lemma (see [64] Theorem 1):

Lemma 7.4. *Assume the Frobenius norm of the weight matrices $W_1,\ldots,W_d$ are at most $M_1,\ldots,M_d$. Let the activation functions be 1-Lipschitz, positive-homogeneous, and applied element-wise (such as the ReLU). Let X is upper bounded by B, i.e., for any X, $\|X\|\leqslant B$. Then,*

$$E\left[\sup_{h\in H}\frac{1}{N}\sum_{i=1}^{N}\sigma_i h(X_i)\right]\leqslant\frac{B(\sqrt{2d\log 2}+1)\Pi_{i=1}^{d}M_i}{\sqrt{N}}.\tag{32}$$

Thus,

$$\hat{\mathfrak{R}}(\mathcal{F})\leqslant\frac{cB(\sqrt{2d\log 2}+1)\Pi_{i=1}^{d}M_i}{\sqrt{N}}.\tag{33}$$

Overall, combining Equation 24, 25 and 27, we have with probability at least $1-\delta$,

$$\begin{aligned}
&R_q(f)-\hat{R}_{s+g}(f)\\
&\leqslant(1-\tau)D_{\mathcal{F}}(\mathcal{S},\mathcal{G})+3(1-\tau)\sqrt{\frac{(b-a)\ln(4/\delta)}{2N_g}}\\
&\quad+2(1-\tau)\hat{\mathfrak{R}}_g(\mathcal{F})+3\tau\sqrt{\frac{(b-a)\ln(4/\delta)}{2N_s}}\\
&\quad+2\tau\hat{\mathfrak{R}}_s(\mathcal{F})+\sqrt{\frac{(b-a)^2\ln(4/\delta)}{2}\left(\frac{\tau^2}{N_s}+\frac{(1-\tau)^2}{N_g}\right)}\\
&\leqslant(1-\tau)D_{\mathcal{F}}(\mathcal{S},\mathcal{N})+(1-\tau)D_{\mathcal{F}}(\mathcal{N},\mathcal{G})\\
&\quad+2(1-\tau)\frac{cB(\sqrt{2d\log 2}+1)\Pi_{i=1}^{d}M_i}{\sqrt{N_g}}\\
&\quad+2\tau\frac{cB(\sqrt{2d\log 2}+1)\Pi_{i=1}^{d}M_i}{\sqrt{N_s}}\\
&\quad+\sqrt{\frac{(b-a)^2\ln(4/\delta)}{2}\left(\frac{\tau^2}{N_s}+\frac{(1-\tau)^2}{N_g}\right)}.
\end{aligned}\tag{34}$$

This completes the proof.

7.2 Proof of Lemma 7.3

Proof.

$$\begin{aligned}
&E\left[\sup_{f\in\mathcal{F}}\frac{1}{N}\sum_{i=1}^{N}\sigma_i\ell(f(X_i),Y_i)\right]\\
&=E\left[\sup_{\arg\max\{h_1,\ldots,h_c\}}\frac{1}{N}\sum_{i=1}^{N}\sigma_i\ell(f(X_i),Y_i)\right]\\
&=E\left[\sup_{\max\{h_1,\ldots,h_c\}}\frac{1}{N}\sum_{i=1}^{N}\sigma_i\ell(f(X_i),Y_i)\right]\\
&\leqslant E\left[\sum_{k=1}^{c}\sup_{h_k\in H}\frac{1}{N}\sum_{i=1}^{N}\sigma_i\ell(f(X_i),Y_i)\right]\\
&=\sum_{k=1}^{c}E\left[\sup_{h_k\in H}\frac{1}{N}\sum_{i=1}^{N}\sigma_i\ell(f(X_i),Y_i)\right]\\
&\leqslant cE\left[\sup_{h_k\in H}\frac{1}{N}\sum_{i=1}^{N}\sigma_i h_k(X_i)\right]\\
&=cE\left[\sup_{h\in H}\frac{1}{N}\sum_{i=1}^{N}\sigma_i h(X_i)\right].
\end{aligned}\tag{35}$$

The first two equations hold because f, $\arg\max\{h_1,\ldots,h_c\}$, and $\max\{h_1,\ldots,h_c\}$ give the same constraint on $h_i(X)$. The fifth inequality holds due to Talagrand Contraction Lemma [65]. □

REFERENCES

[1] C. Finn, P. Abbeel, and S. Levine, "Model-agnostic meta-learning for fast adaptation of deep networks," in *ICML*, 2017. 1, 2, 6, 7

[2] J. Snell, K. Swersky, and R. S. Zemel, "Prototypical networks for few-shot learning," in *NeurIPS*, 2017. 1, 2, 6, 7

[3] B. Hariharan and R. Girshick, "Low-shot visual recognition by shrinking and hallucinating features," in *ICCV*, 2017. 1, 2

[4] Y.-X. Wang, R. Girshick, M. Hebert, and B. Hariharan, "Low-shot learning from imaginary data," in *CVPR*, 2018. 1, 2

[5] M. Ren, E. Triantafillou, S. Ravi, J. Snell, K. Swersky, J. B. Tenenbaum, H. Larochelle, and R. S. Zemel, "Meta-learning for semi-supervised few-shot classification," in *ICLR*, 2018. 1, 5, 7

[6] Y. Xian, T. Lorenz, B. Schiele, and Z. Akata, "Feature generating networks for zero-shot learning," in *CVPR*, 2018. 1, 2

[7] R. Salakhutdinov, J. Tenenbaum, and A. Torralba, "One-shot learning with a hierarchical nonparametric bayesian model," in *ICML workshop*, 2012. 1

[8] S. Yang, L. Liu, and M. Xu, "Free lunch for few-shot learning: Distribution calibration," in *ICLR*, 2021. 2

[9] Z. Li, F. Zhou, F. Chen, and H. Li, "Meta-sgd: Learning to learn quickly for few shot learning," in *arxiv*, 2017. 2, 6

[10] X. Xia, T. Liu, N. Wang, B. Han, C. Gong, G. Niu, and M. Sugiyama, "Are anchor points really indispensable in label-noise learning?" in *NeurIPS*, 2019. 2

[11] X. Xia, T. Liu, B. Han, N. Wang, M. Gong, H. Liu, G. Niu, D. Tao, and M. Sugiyama, "Part-dependent label noise: Towards instance-dependent label noise," in *NeurIPS*, 2020. 2

[12] X. Xia, T. Liu, B. Han, M. Gong, J. Yu, G. Niu, and M. Sugiyama, "Instance correction for learning with open-set noisy labels," *arXiv*, 2021. 2

[13] X. Xia, T. Liu, B. Han, N. Wang, J. Deng, J. Li, and Y. Mao, "Extended t: Learning with mixed closed-set and open-set noisy labels," *arXiv*, 2020. 2

[14] S. Wu, X. Xia, T. Liu, B. Han, M. Gong, N. Wang, H. Liu, and G. Niu, "Class2simi: A noise reduction perspective on learning with noisy labels," in *ICML*, 2021. 2

[15] S. Yang, E. Yang, B. Han, Y. Liu, M. Xu, G. Niu, and T. Liu, "Estimating instance-dependent label-noise transition matrix using dnns," *arXiv*, 2021. 2

[16] S. Yang, P. Sun, Y. Jiang, X. Xia, R. Zhang, Z. Yuan, C. Wang, P. Luo, and M. Xu, "Objects in semantic topology," *arXiv*, 2021. 2

[17] O. Vinyals, C. Blundell, T. Lillicrap, K. Kavukcuoglu, and D. Wierstra, "Matching networks for one shot learning," in *NeurIPS*, 2016. 2, 3, 6

[18] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial nets," in *NeurIPS*, 2014. 2

[19] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, "Learning Representations by Back-propagating Errors," *Nature*, vol. 323, pp. 533–536, 1986. 2

[20] R. Zhang, T. Che, Z. Ghahramani, Y. Bengio, and Y. Song, "Metagan: An adversarial approach to few-shot learning," in *NeurIPS*, 2018. 2, 6

[21] Z. Chen, Y. Fu, Y. Zhang, Y. Jiang, X. Xue, and L. Sigal, "Multi-level semantic feature augmentation for one-shot learning," *TIP*, vol. 28, no. 9, pp. 4594–4605, 2019. 2, 6

[22] E. Schwartz, L. Karlinsky, J. Shtok, S. Harary, M. Marder, A. Kumar, R. Feris, R. Giryes, and A. Bronstein, "Delta-encoder: an effective sample synthesis method for few-shot object recognition," in *NeurIPS*, 2018. 2, 6

[23] H. Gao, Z. Shou, A. Zareian, H. Zhang, and S.-F. Chang, "Low-shot learning via covariance-preserving adversarial augmentation networks," in *NeurIPS*, 2018. 2

[24] J. Zhang, C. Zhao, B. Ni, M. Xu, and X. Yang, "Variational few-shot learning," in *ICCV*, 2019. 2, 6

[25] S. Yang, W. Yu, Y. Zheng, H. Yao, and T. Mei, "Adaptive semantic-visual tree for hierarchical embeddings," in *ACM MM*, 2019. 2

[26] Y. S. Tiexin Qin, Wenbin Li and G. Yang, "Unsupervised few-shot learning via distribution shift-based augmentation," in *arxiv*, 2020. 2

[27] A. Antoniou and A. J. Storkey, "Assume, augment and learn: Unsupervised few-shot meta-learning via random labels and data augmentation," *arXiv*, 2019. 2

[28] S. Yang, M. Xu, H. Xie, S. Perry, and J. Xia, "Single-view 3d object reconstruction from shape priors in memory," in *CVPR*, June 2021, pp. 3152–3161. 2

[29] S.-J. Park, S. Han, J.-w. Baek, I. Kim, J. Song, H. B. Lee, J.-J. Han, and S. J. Hwang, "Meta variance transfer: Learning to augment from the others," in *ICML*, 2020. 2, 6

[30] P. Mangla, N. Kumari, A. Sinha, M. Singh, B. Krishnamurthy, and V. N. Balasubramanian, "Charting the right manifold: Manifold mixup for few-shot learning," in *WACV*, 2020. 3, 5, 7, 8

[31] J. W. Tukey, *Exploratory data analysis*, ser. Addison-Wesley Series in Behavioral Science. Reading, MA: Addison-Wesley, 1977. [Online]. Available: https://cds.cern.ch/record/107005 3

[32] M. Mohri, A. Rostamizadeh, and A. Talwalkar, *Foundations of Machine Learning*. MIT Press, 2018. 4

[33] P. L. Bartlett and S. Mendelson, "Rademacher and gaussian complexities: Risk bounds and structural results," *JMLR*, vol. 3, no. Nov, pp. 463–482, 2002. 4

[34] S. Ben-David, J. Blitzer, K. Crammer, A. Kulesza, F. Pereira, and J. W. Vaughan, "A theory of learning from different domains," *Machine learning*, vol. 79, no. 1, pp. 151–175, 2010. 4

[35] J. Blitzer, K. Crammer, A. Kulesza, F. Pereira, and J. Wortman, "Learning bounds for domain adaptation," *NeurIPS*, 2008. 4

[36] C. Zhang, L. Zhang, and J. Ye, "Generalization bounds for domain adaptation," *NeurIPS*, 2012. 4, 10

[37] S. Ravi and H. Larochelle, "Optimization as a model for few-shot learning," in *ICLR*, 2017. 5, 6

[38] E. Grant, C. Finn, S. Levine, T. Darrell, and T. L. Griffiths, "Recasting gradient-based meta-learning as hierarchical bayes," in *ICLR*, 2018. 6

[39] L. Franceschi, P. Frasconi, S. Salzo, R. Grazzi, and M. Pontil, "Bilevel programming for hyperparameter optimization and meta-learning," in *ICML*, 2018. 6

[40] T. Munkhdalai, X. Yuan, S. Mehri, and A. Trischler, "Rapid adaptation with conditionally shifted neurons," in *ICML*, 2018. 6

[41] K. Lee, S. Maji, A. Ravichandran, and S. Soatto, "Meta-learning with differentiable convex optimization," in *CVPR*, 2019. 6

[42] A. A. Rusu, D. Rao, J. Sygnowski, O. Vinyals, R. Pascanu, S. Osindero, and R. Hadsell, "Meta-learning with latent embedding optimization," in *ICLR*, 2019. 6, 7

[43] Q. Sun, Y. Liu, T.-S. Chua, and B. Schiele, "Meta-transfer learning for few-shot learning," in *CVPR*, 2019. 6

[44] H. Li, D. Eigen, S. Dodge, M. Zeiler, and X. Wang, "Finding task-relevant features for few-shot learning by category traversal," in *CVPR*, 2019. 6

[45] Y. Liu, B. Schiele, and Q. Sun, "An ensemble of epoch-wise empirical bayes for few-shot learning," in *ECCV*, 2020. 6, 7

[46] F. Sung, Y. Yang, L. Zhang, T. Xiang, P. H. Torr, and T. M. Hospedales, "Learning to compare: Relation network for few-shot learning," in *CVPR*, 2018. 6, 7

[47] V. G. Satorras and J. B. Estrach, "Few-shot learning with graph neural networks," in *ICLR*, 2018. 6

[48] J. Kim, T. Kim, S. Kim, and C. D. Yoo, "Edge-labeling graph neural network for few-shot learning," in *CVPR*, 2019. 6

[49] L. Bertinetto, J. F. Henriques, P. H. S. Torr, and A. Vedaldi, "Meta-learning with differentiable closed-form solvers," in *ICLR*, 2019. 6

[50] Y. Liu, J. Lee, M. Park, S. Kim, E. Yang, S. Hwang, and Y. Yang, "Learning to propagate labels: transductive propagation network for few-shot learning," in *ICLR*, 2019. 6, 7

[51] B. Liu, Y. Cao, Y. Lin, Q. Li, Z. Zhang, M. Long, and H. Hu, "Negative margin matters: Understanding margin in few-shot classification," in *ECCV*, 2020. 6

[52] H. Li, D. Eigen, S. Dodge, M. Zeiler, and X. Wang, "Finding task-relevant features for few-shot learning by category traversal," in *CVPR*, 2019. 7

[53] Q. Sun, Y. Liu, T.-S. Chua, and B. Schiele, "Meta-transfer learning for few-shot learning," in *CVPR*, 2019. 7

[54] C. Zhang, Y. Cai, G. Lin, and C. Shen, "Deepemd: Few-shot image classification with differentiable earth mover's distance and structured classifiers," in *CVPR*, 2020. 7

[55] P. Welinder, S. Branson, T. Mita, C. Wah, F. Schroff, S. Belongie, and P. Perona, "Caltech-UCSD Birds 200," California Institute of Technology, Tech. Rep. CNS-TR-2010-001, 2010. 5

[56] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. S. Bernstein, A. C. Berg, and L. Fei-Fei, "Imagenet large scale visual recognition challenge," *Int. J. Comput. Vis.*, vol. 115, no. 3, pp. 211–252, 2015. 5

[57] W.-Y. Chen, Y.-C. Liu, Z. Kira, Y.-C. F. Wang, and J.-B. Huang, "A closer look at few-shot classification," in *ICLR*, 2019. 5

[58] C. Cortes and V. Vapnik, "Support-vector networks," *Mach. Learn.*, 1995. 5, 7

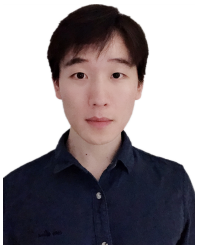
[59] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay, "Scikit-learn: Machine learning in Python," *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011. 5

[60] L. van der Maaten and G. Hinton, "Visualizing data using t-SNE," *Journal of Machine Learning Research*, 2008. 7, 9

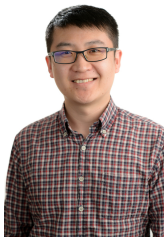
[61] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *CVPR*, 2016. 7, 8

[62] S. Zagoruyko and N. Komodakis, "Wide residual networks," in *BMVC*, 2016. 7, 8
[63] O. Bousquet, S. Boucheron, and G. Lugosi, "Introduction to statistical learning theory," in *Summer School on Machine Learning*. Springer, 2003, pp. 169–207. 10
[64] N. Golowich, A. Rakhlin, and O. Shamir, "Size-independent sample complexity of neural networks," in *COLT*, 2018. 11
[65] E. F. Beckenbach and R. Bellman, *Inequalities*. Springer Science & Business Media, 2012, vol. 30. 11

Shuo Yang received the B.E. degree in computer science and technology from the Harbin Institute of Technology, Harbin, China, in 2020. He is currently pursuing a Ph.D. degree in the School of Electrical and Data Engineering, Faculty of Engineering and Information Technology, University of Technology Sydney, advised by Prof.Min Xu. His research lies in computer vision and machine learning.



Songhua Wu received the B.E. degree in electronic science and technology from the University of Science and Technology of China, in 2019. He is currently pursuing a Ph.D. degree in the Trustworthy Machine Learning Lab with the School of Computer Science at the University of Sydney. His research interests include statistical learning theory, weakly supervised learning, and causal representation learning.



Tongliang Liu is currently a Senior Lecturer with School of Computer Science at the University of Sydney. He is heading the Trustworthy Machine Learning Laboratory and is also a Visiting Scientist at RIKEN AIP. He is broadly interested in the fields of trustworthy machine learning and its interdisciplinary applications, with a particular emphasis on learning with noisy labels, adversarial learning, transfer learning, unsupervised learning, and statistical deep learning theory. He has authored and co-authored more than 100 research articles including ICML, NeurIPS, ICLR, CVPR, ICCV, ECCV, AAAI, IJCAI, KDD, ICME, IEEE T-PAMI, T-NNLS, and T-IP, with best paper awards, e.g., the 2019 ICME Best Paper Award and the PacificVis 2021 Best VisNotes Paper Award. He is/was Area Chair for many conferences, such as NeurIPS, ICLR, UAI, AAAI, and IJCAI. He is a recipient of Discovery Early Career Researcher Award (DECRA) from Australian Research Council (ARC); the Cardiovascular Initiative Catalyst Award by the Cardiovascular Initiative; and was named in the Early Achievers Leaderboard of Engineering and Computer Science by The Australian in 2020.

Min Xu is currently an Associate Professor at University of Technology Sydney. She received the B.E. degree from the University of Science and Technology of China, Hefei, China, in 2000, the M.S. degree from National University of Singapore, Singapore, in 2004, and the Ph.D. degree from University of Newcastle, Callaghan NSW, Australia, in 2010. Her research interests include multimedia data analytics, computer vision and machine learning. She has published over 100 research papers in high quality international journals and conferences. She has been invited to be a member of the program committee for many international top conferences, including ACM Multimedia Conference and reviewers for various highlyrated international journals, such as IEEE Transactions on Multimedia, IEEE Transactions on Circuits and Systems for Video Technology and much more. She is an Associate Editor of Journal of Neurocomputing.