Feature Distribution Fitting with Direction-Driven Weighting for Few-Shot Images Classification

Xin Wei[1], Wei Du[1], Huan Wan[2], Weidong Min[3, 4*]

[1] School of Software, Nanchang University
[2] School of Computer and Information Engineering, Jiangxi Normal University
[3] School of Mathematics and Computer Science, Institute of Metaverse, Nanchang University
[4] Jiangxi Key Laboratory of Smart City, Nanchang University
{xinwei, minweidong}@ncu.edu.cn, duwei@email.ncu.edu.cn, huanwan@jxnu.edu.cn

Abstract

Few-shot learning has received increasing attention and witnessed significant advances in recent years. However, most of the few-shot learning methods focus on the optimization of training process, and the learning of metric and sample generating networks. They ignore the importance of learning the ground-truth feature distributions of few-shot classes. This paper proposes a direction-driven weighting method to make the feature distributions of few-shot classes precisely fit the ground-truth distributions. The learned feature distributions can generate an unlimited number of training samples for the few-shot classes to avoid overfitting. Specifically, the proposed method consists of two optimization strategies. The direction-driven strategy is for capturing more complete direction information that can describe the feature distributions. The similarity-weighting strategy is proposed to estimate the impact of different classes in the fitting procedure and assign corresponding weights. Our method outperforms the current state-of-the-art performance by an average of 3% for 1-shot on standard few-shot learning benchmarks like *mini*ImageNet, CIFAR-FS, and CUB. The excellent performance and compelling visualization show that our method can more accurately estimate the ground-truth distributions.

Introduction

With the enormous amount of data and the rapid development of deep learning, supervised image classification has achieved ultra-high precision. However, building large-scale labeled datasets is expensive, and it is impossible to provide a large amount of training data in many application scenarios. Therefore, few-shot learning develops significantly in this context. Many few-shot learning methods have been proposed to improve the performance of models in few-shot image classification. Finn et al. (Finn, Abbeel, and Levine 2017) propose an algorithm based on meta-learning to make the model quickly adapt to new tasks with only a small number of training samples. The method proposed by Vinyals et al. (Vinyals et al. 2016) combines metric learning and attLSTM to build networks for classifying the query images, improving the classification ability of the models. Chen et al. (Chen et al. 2019a) train the feature extractor on base class data, and train the specific-task classifiers on the labeled samples in novel classes, surprisingly achieving outperformance. Li et al. (Li et al. 2020) construct conditional Wasserstein Generative Adversarial Networks to synthesize more training data to alleviate the data insufficiency problem. Most of the existing methods improve the generalization ability of the models through the optimization of training process, the learning of metric and sample generating networks. They do not focus on the ground-truth distribution of samples in the feature space. The ground-truth distribution is hard to estimate precisely through only a few feature vectors of samples. Yang et al. (Yang et al. 2021) propose a distribution calibration strategy (DC) to estimate the ground-truth distributions of the few-shot classes. The strategy measures the matching strength between the base classes and the samples in the support set (i.e., the target samples), and then transfers the statistics from the similar base classes to the few-shot classes in support sets based on the matching strength.

When measuring the matching strength between a base class and a target sample, a feasible way is to match all base class samples and the within-class variation with the target sample. This manner is similar to sparse representation, which makes full use of the location and sample distribution information of the base class. We call this manner Manner 1. As the computational cost of Manner 1 is too high, DC (Yang et al. 2021) calculates the sample mean of each base class and uses it to compare the similarity with the target sample, namely only keeps location information, thereby remarkably reducing the computational cost while maintaining a certain level of effectiveness. We observe that Manner 1 performs the matching with target samples through Euclidean distance, i.e., only measures the difference in each dimension separately. Each dimension corresponds to a direction in the feature space. The observation means Manner 1 ignores the location and distribution information in other directions, as shown in Fig. 1. Thus, it is necessary to add descriptive information in other directions.

As a simple and efficient way, more directional information can be obtained by calculating the cross product of the feature vector and its transpose to construct the feature description matrix of each base class sample and the description matrix of the different within-class variations, which is called Manner 2. Values in new dimensions/directions

*Corresponding author

are obtained by multiplying the values in two different dimensions/directions. However, considering that the individual calculation of each description matrix faces extremely high computational costs, it is simplified to calculate the feature description matrix of the mean value of each base class. This simplification is our proposed direction-driven strategy. It is worth noting that the proposed strategy has an equivalent transformation relation with Manner 2 in mathematics. Mathematical proofs are presented in the detailed description of the proposed method. In addition, different base classes contribute differently to the distribution fitting process of novel classes. Thus we also propose a similarity-weighting strategy to estimate the impact of the base classes in the fitting procedure and assign corresponding weights. Fig. 2 illustrates the two strategies.

The direction-driven strategy and the similarity-weighting strategy constitute the Direction-Driven Weighting Method (DDWM) that we propose in this paper. DDWM is applied for feature distribution fitting and few-shot image classification. It allows the estimated biased feature distributions of few-shot classes to more accurately fit the corresponding ground-truth distributions. With DDWM, more samples of the novel classes can be generated from the learned feature distribution, transferring the few-shot classification problem to the general classification problem.

The main contributions of this work are as follows:

- A direction-driven strategy is proposed to better describe the characteristics of the feature distributions in other directions. And we theoretically explained and analyzed it.
- A similarity-weighting strategy is proposed to measure the influence of different base classes and assign corresponding weights in the distribution fitting process.
- We propose the Direction-Driven Weighting Method (DDWM) to make the biased feature distributions precisely fit the corresponding ground-truth distributions and improve the few-shot classification performance.
- Extensive experiments are conducted to verify the performance of our method, including the comparison with SOTA methods, ablation study, and visualization verification. Results show that the proposed method can significantly improve the classification accuracy and achieve state-of-the-art accuracy on three datasets.

Related Work

Recently, few-shot image classification methods follow the task mechanism (Vinyals et al. 2016) to construct massive tasks from related datasets to simulate the condition of a few samples for training. Generally, these typical few-shot learning methods can be roughly divided into four categories: optimization-based methods, metric-based methods, fine-tuning-based methods, and generation-based methods.

The optimization-based methods learn to optimize the gradient descent procedure of the network by an alternate optimization strategy, such as Model-Agnostic Meta-Learning (MAML) (Finn, Abbeel, and Levine 2017) and Almost No Inner Loop (ANIL) (Raghu et al. 2020). The network with optimization-based methods can have a good

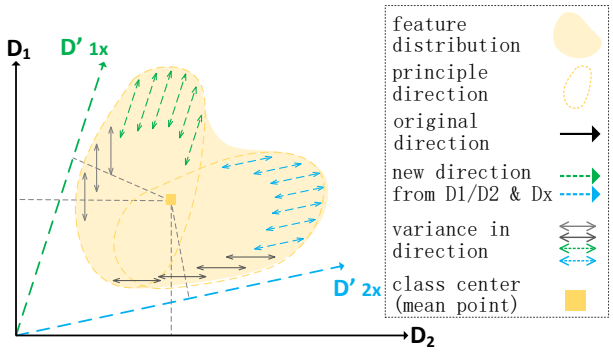

Figure 1: The example of feature distribution described in multiple directions. Previous work describes the distribution only in limited dimension directions (e.g., D_1 and D_2), which does not effectively describe the characteristics of the distribution in other directions. We describe the distribution in both original dimension directions and other dimension directions (e.g., D'_{1x} and D'_{2x}) for more location and distribution information.

initialization, updated direction, and learning rate to adapt quickly on tasks. The metric-based methods, such as ProtoNet (Snell, Swersky, and Zemel 2017) and MatchNet (Vinyals et al. 2016), learn to classify samples by calculating the distance between the query images and the representatives of each support class. The fine-tuning-based methods, such as baseline++ (Chen et al. 2019a) and RFS-simple (Tian et al. 2020), perform model pre-training with the base classes and then fine-tune a new classifier which will be relearned every time with little novel class data. Especially, S2M2$_R$ (Mangla et al. 2020) fine-tunes the backbone with the Manifold Mixup method for a few more epochs.

The generation-based methods (ZHANG et al. 2018; Schwartz et al. 2018; Zhang et al. 2019; Li et al. 2020; Hong et al. 2020; Bendre, Desai, and Najafirad 2021) build complex networks for generating samples or features based on Generative Adversarial Networks (GANs) (Goodfellow et al. 2014) and auto-encoder (Rumelhart, Hinton, and Williams 1986). Li et al. (Li et al. 2020) introduce a conditional Wasserstein GAN (WGAN) to synthesize fake features. Bendre et al. (Bendre, Desai, and Najafirad 2021) leverage a variational auto-encoder to reconstruct features by employing a multimodal strategy including semantic and image information. By contrast, Hong et al. (Hong et al. 2020) combine the matching procedure with GANs to generate image samples rather than feature samples. These methods can efficiently synthesize samples while requiring complex networks and algorithms.

The other generation-based methods (Yang et al. 2021; Zhang et al. 2021) transfer the information from the similar base classes to the few-shot classes. Yang et al. (Yang et al. 2021) propose a distribution calibration strategy to calibrate the biased feature distributions of few-shot classes to the corresponding ground-truth distributions. They transfer the statistics of feature distributions in similar base classes to the biased feature distributions. Compared with the cal-

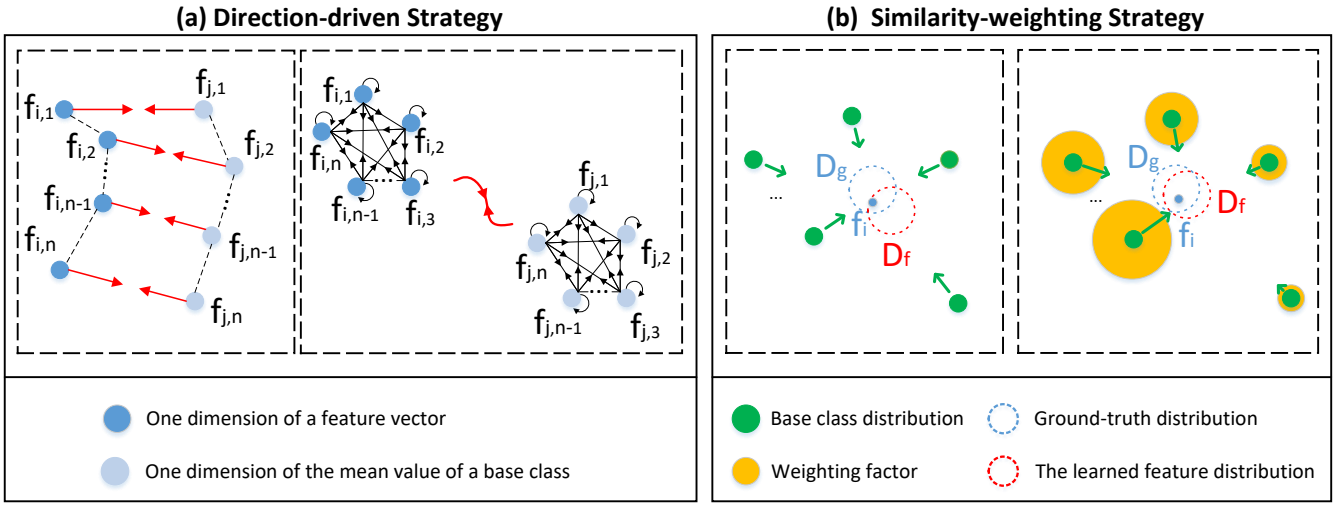

Figure 2: The illustration of the Direction-Driven Weighting Method: (a) direction-driven strategy, and (b) similarity-weighting strategy. The direction-driven strategy measures distribution information by the feature description matrix. Feature description matrix builds the relationship between different dimensions, creating new directions. The similarity-weighting strategy estimates the impact of the base classes in the fitting procedure and assigns corresponding weights to different base classes.

ibration strategy, our method can describe feature distributions in multiple directions for more information and estimate ground-truth distributions more accurately.

Methodology

This section gives a detailed description of the proposed method.

Problem Definition

Following the standard setting of few-shot image classification, the whole dataset consists of data-label pairs $\mathcal{D} = \{(X_i, y_i)\}$, where $X_i \in \mathbb{R}^{H\times W\times 3}$ is the ith image, $y_i \in \mathcal{C}$ is the class label of X_i, and $\mathcal{C}$ denotes the label set of all classes. $\mathcal{C}$ can be divided into the label set of base classes $\mathcal{C}_b$ and the label set of novel classes $\mathcal{C}_n$, where $\mathcal{C}_b \cup \mathcal{C}_n = \mathcal{C}$ and $\mathcal{C}_b \cap \mathcal{C}_n = \varnothing$. The dataset $\mathcal{D}$ is accordingly divided into two subsets, namely, the base class data $\mathcal{D}_b$ and the novel class data $\mathcal{D}_n$. In the first training stage, a deep neural network is trained on base classes to obtain a parameter-fixed feature extractor for feature extraction on base and novel classes. In the second training stage, the M-way-K-shot task (Vinyals et al. 2016) is built based on novel classes, where M classes are selected from the novel classes $\mathcal{C}_n$, and only K labeled samples in each selected novel class are used in the task. Each task $\mathcal{T}$ has its own support set $\mathcal{S} = \{(x_i, y_i)\}_{i=1}^{M\times K}$ and query set $\mathcal{Q} = \{(x_i, y_i)\}_{i=1}^{M\times q}$, where $x_i \in \mathbb{R}^d$ is the feature vector of the ith image, $y_i \in \mathcal{C}_n$ is the class label of x_i, and q is the number of testing samples of each novel class. The model uses the support set $\mathcal{S}$ to build a classifier for training and uses the query set $\mathcal{Q}$ for evaluation.

Distribution Fitting Procedure

Base classes have sufficient labeled data, so it is convenient to obtain statistical information from the feature distributions of base class samples. However, the support set involved in task $\mathcal{T}$ only contains a few samples, which makes it difficult to estimate the ground-truth distributions of few-shot classes. The Direction-Driven Weighting Method (DDWM) is proposed to make the biased feature distributions of few-shot classes fit the corresponding ground-truth distributions. The algorithm of DDWM is shown in Algorithm 1. The following sections describe the whole procedure of our method and how to use it to improve the generalization ability of the model.

Equivalent Transformation Relation. The mathematical proof about the equivalent transformation relation is given in this section. The cross product of the feature vector and its transpose is calculated as the feature description matrix. In a base class, the sum of feature description matrices of all samples can be formulated as:

$$v_1 v_1^T + v_2 v_2^T + ... + v_n v_n^T = \sum_{i=1}^{n} v_i v_i^T, \tag{1}$$

where v_i is a column feature vector of the ith sample in a base class and n is the sample number of this base class. For each pair of samples in a base class, the cross product of their feature vectors is used to reflect the within-class variation. In this base class, the sum of description matrices of the different within-class variations is calculated as:

$$v_1 v_2^T + v_1 v_3^T + ... + v_n v_{n-1}^T = \sum_{i,j=1, i\neq j}^{n} v_i v_j^T. \tag{2}$$

The mean feature vector of a base class is obtained by computing the mean of each dimension. The sum of feature description matrices of the mean vector is calculated as:

$$(\frac{1}{n}\sum_{i=1}^{n} v_i)(\frac{1}{n}\sum_{j=1}^{n} v_j)^T = \frac{1}{n^2}(\sum_{i,j=1}^{n} v_i v_j^T). \tag{3}$$

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

Input: support set $\mathcal{S} = \{(x_i, y_i)\}_{i=1}^{M\times K}$, the mean of the jth base class μ_j $(j = 1, 2, ..., |\mathcal{C}_b|)$, and the covariance matrix of the jth base class Σ_j $(j = 1, 2, ..., |\mathcal{C}_b|)$.
Output: the optimal parameters of the classifier f_θ.

```
Transform {x_i}_{i=1}^{M×K} to {x̂_i}_{i=1}^{M×K} with Tukey's Ladder of Power transformation;
Update the support set S = {(x̂_i, y_i)}_{i=1}^{M×K};
Calculate feature description matrices {M'_j}_{j=1}^{|C_b|} of each base class using μ_j as Equation 8;
for (x̂_i, y_i) in S do
    Calculate the feature description matrix M_i of the feature vector x̂_i as Equation 7;
    Build a set F containing Frobenius norm values of the difference between M_i and {M'_j}_{j=1}^{|C_b|} as Equation 9;
    Select k nearest base classes to build a set of class labels I_k based on F as Equation 10;
    Calculate weight factors {λ_j | j ∈ I_k} of k nearest base classes based on {μ_j | j ∈ I_k} and x̂_i as Equation 11;
    Calculate the mean μ̃ and the covariance Σ̃ of the learned feature distribution for class y_i as Equation 12 and Equation 13;
    Sample feature vectors for class y_i from the learned distribution as Equation 14
end for
Train a classifier f_θ using both sampled features and support set features as Equation 15.
```

It can be observed that Eq. 1 + Eq. 2 = n^2 * Eq. 3, which indicates that the feature description matrix of base class samples and within-class variations can be simplified as the feature description matrix of mean vector, reducing the compuational complexity from $O(n^2)$ to $O(1)$.

Statistics of the Base Classes. According to the method proposed by Yang et al. (Yang et al. 2021), the data of every feature dimension from the same class follows a Gaussian distribution in the feature space, and similar classes usually have similar statistics of the feature representations, like mean and variance. As base classes have sufficient samples, their statistics can be estimated more accurately. The mean of feature vectors from the jth base class and the covariance matrix of the feature vectors from the jth base class are calculated as:

$$\mu_j = \frac{1}{n_j}\sum_{i=1}^{n_j}(x_i), \tag{4}$$

$$\Sigma_j = \frac{1}{n_j - 1}\sum_{i=1}^{n_j}(x_i - \mu_j)(x_i - \mu_j)^T, \tag{5}$$

where x_i is a feature vector of the ith sample in the jth base class, and n_j is the total number of the samples in the jth base class.

Tukey's Ladder of Power Transformation. To make the feature distribution more consistent with the Gaussian distribution, Tukey's Ladder of Power transformation (Tukey 1977) is used to transform the feature vectors in the support set and query set as:

$$\hat{x}_i = \begin{cases} x_i^\beta & \beta \neq 0 \\ \log x_i & \beta = 0 \end{cases}, \tag{6}$$

where x_i is the ith feature vector in the support set and query set, and β is a hyperparameter to adjust the skewness of a distribution. After the transformation, the support set and the query set are updated to $\mathcal{S} = \{(\hat{x}_i, y_i)\}_{i=1}^{M\times K}$ and $\mathcal{Q} = \{(\hat{x}_i, y_i)\}_{i=1}^{M\times q}$.

Direction-Driven Strategy. For more location and distribution information, the feature description matrix is used to describe the distribution in both original dimension directions and other dimension directions. The feature description matrix $\mathcal{M}_i$ of the feature vector $\hat{x}_i$ in the support set and the feature description matrix $\mathcal{M}_j^{'}$ of the mean μ_j of the jth base class are calculated as:

$$\mathcal{M}_i = \hat{x}_i\hat{x}_i^T, \tag{7}$$

$$\mathcal{M}_j^{'} = \mu_j{\mu_j}^T. \tag{8}$$

After calculating the feature description matrices, we select k base classes that are highly matched with the target sample to transfer their statistics to the few-shot class. To measure the matching strength, the Frobenius norm is applied as follows:

$$\mathbb{F} = \{- \parallel \mathcal{M}_i - \mathcal{M}_j^{'} \parallel_F |\ j \in \mathcal{C}_b\}. \tag{9}$$

Based on Frobenius norm, the k most matched base classes are selected as follows:

$$\mathbb{I}_k = \{j \mid - \parallel \mathcal{M}_i - \mathcal{M}_j^{'} \parallel_F \in Topk(\mathbb{F})\}, \tag{10}$$

where $Topk(\cdot)$ is the operator to select the top elements. $\mathbb{F}$ is the set of Frobenius norm values, and $\mathbb{I}_k$ is the class label set containing the labels of k nearest base classes with respect to the feature vector $\hat{x}_i$ from the support set $\mathcal{S}$.

Similarity-Weighting Strategy. Nearer classes have more similar statistics of the feature representations (Yang et al. 2021). Therefore, different base classes contribute differently to the distribution fitting process. To measure their impact more accurately, different weights are assigned to the k base classes. The weight factor is formulated as:

$$\lambda_j = \frac{1}{(1+ \parallel \hat{x}_i - \mu_j \parallel_2)^\gamma}, j \in \mathbb{I}_k, \tag{11}$$

where γ, as a weight controller, is a hyperparameter used to control the magnitude of the weight factor. The weight factor λ_j is assigned to the corresponding mean and covariance of the k nearest base classes. The mean and covariance of the learned feature distribution are given as:

$$\tilde{\mu} = \frac{1}{\sum_{j\in\mathbb{I}_k}\lambda_j + 1} * (\sum_{j\in\mathbb{I}_k}\lambda_j\mu_j + \hat{x}_i), \tag{12}$$

$$\tilde{\Sigma} = \frac{1}{\sum_{j\in\mathbb{I}_k}\lambda_j + 1} * (\sum_{j\in\mathbb{I}_k}\lambda_j\Sigma_j + \alpha), \tag{13}$$

where α is the compensation for the within-class variation in few-shot classes. The above distribution fitting procedure is performed on each feature vector from the support set $\mathcal{S}$ for more diverse and accurate distribution estimation.

	Method	*mini*ImageNet		CUB	
		5-way-1-shot	**5-way-5-shot**	**5-way-1-shot**	**5-way-5-shot**
Optimization-based	MAML (Finn, Abbeel, and Levine 2017)	48.70 ± 1.75	63.11 ± 0.92	50.45 ± 0.97	59.60 ± 0.84
	Meta-SGD (Li et al. 2017)	50.47 ± 1.87	64.03 ± 0.94	53.34 ± 0.97	67.59 ± 0.82
	adaResNet (Munkhdalai et al. 2018)	56.88 ± 0.62	71.94 ± 0.57	-	-
	LEO (Rusu et al. 2019)	61.76 ± 0.08	77.59 ± 0.12	-	-
	E3BM (Liu, Schiele, and Sun 2020)	63.8 ± 0.4	80.29 ± 0.25	-	-
	Meta Transfer Learning (Sun et al. 2019)	64.3 ± 1.7	80.9 ± 0.8	-	-
Metric-based	MatchingNet (Vinyals et al. 2016)	43.44 ± 0.77	55.31 ± 0.73	73.49 ± 0.89	84.45 ± 0.58
	ProtoNet (Snell, Swersky, and Zemel 2017)	49.42 ± 0.78	68.20 ± 0.66	72.99 ± 0.88	86.64 ± 0.51
	RelationNet (Sung et al. 2018)	50.44 ± 0.82	65.32 ± 0.70	68.65 ± 0.91	81.12 ± 0.63
	CAN (Hou et al. 2019)	63.85 ± 0.48	79.44 ± 0.34	-	-
	AAP2S (Ma et al. 2022)	64.82 ± 0.12	81.31 ± 0.22	77.64 ± 0.19	90.43 ± 0.18
	Sum-min (Afrasiyabi et al. 2022)	68.32 ± 0.62	82.71 ± 0.46	79.60 ± 0.80	90.48 ± 0.44
Fine-tuning-based	Baseline++ (Chen et al. 2019a)	53.97 ± 0.79	75.90 ± 0.61	69.55 ± 0.89	85.17 ± 0.50
	RFS-simple (Tian et al. 2020)	62.02 ± 0.63	79.64 ± 0.44	-	-
	Negative-Cosine (Liu et al. 2020)	62.33 ± 0.82	80.94 ± 0.59	72.66 ± 0.85	89.40 ± 0.43
	S2M2 (Mangla et al. 2020)	64.93 ± 0.18	83.18 ± 0.11	-	-
Generation-based	MetaGAN (ZHANG et al. 2018)	52.71 ± 0.64	68.63 ± 0.67	-	-
	Delta-Encoder (Schwartz et al. 2018)	59.9	69.7	69.8	82.6
	TriNet (Chen et al. 2019b)	58.12 ± 1.37	76.92 ± 0.69	69.61 ± 0.46	84.10 ± 0.35
	Meta Variance Transfer (Park et al. 2020)	-	67.67 ± 0.70	-	80.33 ± 0.6
	DC (Yang et al. 2021)	68.12 ± 0.20	83.08 ± 0.14	78.29 ± 0.22	88.92 ± 0.12
Ours	DDWM	$\mathbf{68.58 \pm 0.20}$	$\mathbf{84.65 \pm 0.13}$	$\mathbf{80.40 \pm 0.20}$	$\mathbf{90.75 \pm 0.11}$

Table 1: 5-way-1-shot and 5-way-5-shot classification accuracy (%) on *mini*ImageNet and CUB with 95% confidence intervals. The numbers in bold have intersecting confidence intervals with the most accurate method.

Subsequent Application of DDWM. After transferring the statistics from the k most matched base classes to the few-shot class, the learned feature distributions are denoted as a set $\mathbb{N}_{y_i} = \{\mathcal{N}(\tilde{\mu}_i, \tilde{\Sigma}_i) \mid i \in (1, ..., K)\}$, where $\tilde{\mu}_i$ and $\tilde{\Sigma}_i$ are the learned mean and covariance, respectively. Here, the size of the set is the value of K in an M-way-K-shot task. An unlimited number of feature vectors can be sampled from the learned feature distributions. The set of generated feature vectors for class y_i is formulated as:

$$\mathbb{D}_{y_i} = \{(x_i, y_i) \mid x_i \sim \mathcal{N}(\tilde{\mu}_i, \tilde{\Sigma}_i), \forall \mathcal{N}(\tilde{\mu}_i, \tilde{\Sigma}_i) \in \mathbb{N}_{y_i}\}. \quad (14)$$

The number of sampled features for label y_i is set as a hyperparameter N. A task-specific classifier is trained by minimizing loss on all $L2$-normalized generated features and original features of the support set. The loss function for classifier training is given by:

$$\mathcal{L} = \mathbb{E}_{(x,y) \in \mathcal{S} \cup \{\mathbb{D}_{y_i} | y_i \in \mathcal{Y}^{\mathcal{S}}\}} [L_{CE}(f_\theta(\frac{x}{\| x \|_2}), y)], \quad (15)$$

where $\mathcal{Y}^{\mathcal{S}}$ is the set of labels for the support set $\mathcal{S}$, L_{CE} is the cross-entropy loss, and the classifier model is parameterized by θ.

Experiments

Datasets and Evaluation Metric

The experiments are conducted on three widely used few-shot learning benchmarks, including *mini*ImageNet (Vinyals et al. 2016), CUB (Wah et al. 2011), and CIFAR-FS (Bertinetto et al. 2019). *mini*ImageNet is a mini-version of the ImageNet dataset (Russakovsky et al. 2015). It contains 100 classes with 600 images per class, and the image size is 84 × 84 × 3. *mini*ImageNet is divided into 64 base classes, 16 validation classes, and 20 novel classes in all experiments. CUB (Wah et al. 2011) is a fine-grained classification benchmark that includes different bird classes and contains about 11,788 images of size 84 × 84 × 3. CUB is split into 100 base classes, 50 validation classes, and 50 novel classes. CIFAR-FS is created by randomly splitting 100 classes of CIFAR-100 (Krizhevsky and Hinton 2009) into 64 base classes, 16 validation classes, and 20 novel classes. The images are of size 32 × 32 × 3.

Following the 5-way-1-shot and 5-way-5-shot settings, the top-1 accuracy is used as the evaluation metric to evaluate the performance of different methods on three datasets. The experiments are conducted on 10,000 tasks randomly sampled from novel classes. In each task, 15 query images per class are input to the classifier for evaluation. The reported result is the average accuracy of 10,000 tasks.

Implementation Details

Similar to the previous work (Mangla et al. 2020), for each dataset, a WideResNet model is trained on base classes as the feature extractor. For the task-specific classifier, we choose the logistic regression classifier with $L2$ regularization to test the performance on novel classes. The hyperparameters of our model contain the power of Tukey's transformation β, the weight controller γ, the number of matched base classes k, the compensation for the within-class variation α, and the number of generated features for each sup-

	Method	CIFAR-FS	
		5-way-1-shot	**5-way-5-shot**
Optimization-based	MAML (Finn, Abbeel, and Levine 2017)	58.9 ± 1.9	71.5 ± 1.0
	R2D2 (Bertinetto et al. 2019)	65.4 ± 0.2	79.4 ± 0.2
	MetaOptNet (Lee et al. 2019)	72.8 ± 0.7	85.0 ± 0.5
Metric-based	RelationNet (Sung et al. 2018)	55.0 ± 1.0	69.3 ± 0.8
	ProtoNet (Snell, Swersky, and Zemel 2017)	55.5 ± 0.7	72.0 ± 0.6
	Shot-Free (Ravichandran, Bhotika, and Soatto 2019)	69.15	84.70
	AAP2S (Ma et al. 2022)	73.12 ± 0.22	85.69 ± 0.16
	RENet (Kang et al. 2021)	74.51 ± 0.46	86.60 ± 0.32
Fine-tuning-based	Baseline++ (Chen et al. 2019a)	67.50 ± 0.64	80.08 ± 0.32
	RFS-simple (Tian et al. 2020)	71.5 ± 0.8	86.0 ± 0.5
Ours	DDWM	$\mathbf{75.26 \pm 0.21}$	$\mathbf{86.90 \pm 0.15}$

Table 2: 5-way-1-shot and 5-way-5-shot classification accuracy (%) on CIFAR-FS with 95% confidence intervals. The numbers in bold have intersecting confidence intervals with the most accurate method.

Notation / Dataset	β	γ	k	α	N
*mini*Imagenet	0.5	1.1	5	0.01	550
CIFAR-FS	0.5	0.5	4	0.06	500
CUB	0.5	1.1	6	0.01	550

Table 3: The hyperparameter setting of DDWM.

port set class N. Table 3 shows the setting of hyperparameters in detail. All experiments are conducted with the configuration of Nvidia Quadro RTX 5000 (16GB), RAM 64GB, Ubuntu 20.04 and torch 1.7.1.

Comparison with State-of-the-art Methods

The performance of DDWM is compared with other state-of-the-art methods. Table 1 presents the 5-way-1-shot and 5-way-5-shot classification results on *mini*ImageNet and CUB, and the results on CIFAR-FS are shown in Table 2. We can observe that DDWM outperforms the current state-of-the-art performance by an average of 3% on three datasets for 1-shot setting. From these experiments, we conclude that the proposed method achieves the best performance on 1-shot and 5-shot settings of *mini*ImageNet, CIFAR-FS, and CUB.

Visualization of Generated Samples

In Fig. 3, a 5-way-1-shot task sampled from CUB is randomly selected to show the t-SNE (van der Maaten and Hinton 2008) visualization of the ground-truth distributions and feature samples generated by DDWM. For the 5-way-1-shot task, the support set only contains five samples. We show the support set in Fig. 3 (a), the feature distributions generated by DDWM in Fig. 3 (b), the feature distribution generated by DC method (Yang et al. 2021) in Fig. 3 (c), and the ground-truth feature distributions in Fig. 3 (d). Fig. 3 shows that the feature distributions generated by DDWM almost cover the corresponding ground-truth distributions. As a comparison, there is little overlap between the feature distributions of the DC method and the ground-truth feature distributions. Fig. 3 illustrates the inherent plausibility of our method in improving the accuracy of few-shot classification tasks.

Tukey Trans.	Training with G.F.	$L2$ Norm.	*mini*ImageNet	
			5-way-1-shot	5-way-5-shot
×	×	×	60.06 ± 0.20	81.43 ± 0.14
✓	×	×	64.66 ± 0.20	83.43 ± 0.13
×	✓	×	64.93 ± 0.20	82.65 ± 0.14
×	×	✓	66.64 ± 0.20	83.51 ± 0.13
✓	✓	×	68.14 ± 0.20	84.29 ± 0.13
✓	×	✓	66.07 ± 0.19	83.52 ± 0.13
×	✓	✓	66.18 ± 0.20	82.41 ± 0.14
✓	✓	✓	$\mathbf{68.58 \pm 0.20}$	$\mathbf{84.65 \pm 0.13}$

Table 4: Ablation study for 5-way-1-shot and 5-way-5-shot settings on *mini*ImageNet. Note that transformation, generated features, and normalization are abbreviated as Trans., G.F., and Norm. in this table.

Ablation Study

In order to prove the performance of multiple ingredients in the proposed method, we report the results of ablation studies in Table 4. It shows the effects of Tukey's Ladder of Powers transformation, the generated features, and the $L2$ normalization. The baseline is the performance of the model trained without any ingredients of the proposed method. Comparing the baseline, the Tukey transformation, the generated features, and $L2$ normalization improve the performance for 1-shot by 4.6%, 4.9%, and 6.6%, respectively. There is a significant improvement of over 8% when using both the Tukey transformation and the generated features. Note that $L2$ normalization can only slightly improve baseline performance when using Tukey's transformation and the generated features simultaneously.

Hyperparameter Tuning

The hyperparameters are tuned on the validation sets, and we choose the hyperparameter values that lead to the highest

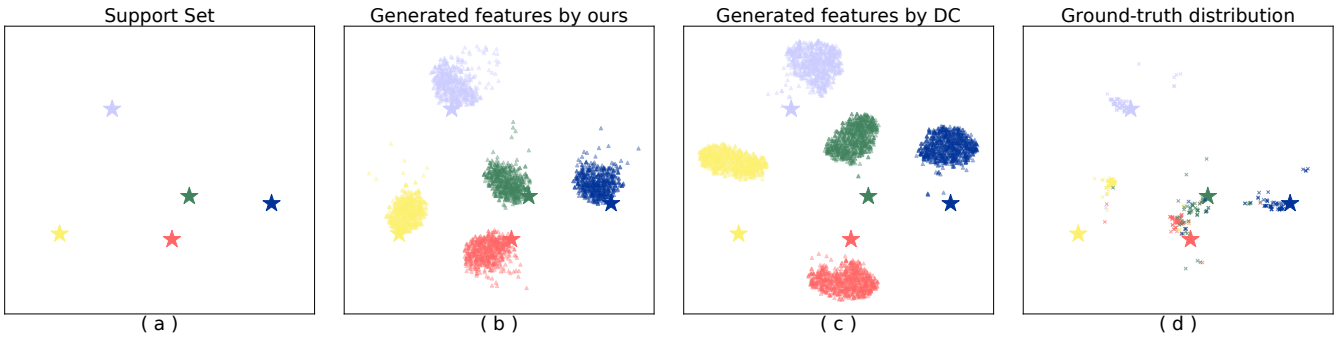

Figure 3: The t-SNE visualization of feature distributions learned by DDWM and ground-truth distributions on a random 5-way-1-shot task sampled from CUB. ★ represents support set features, ▲ in figure (b) (c) represents the generated features, × in figure (d) represents the features of ground-truth distributions.

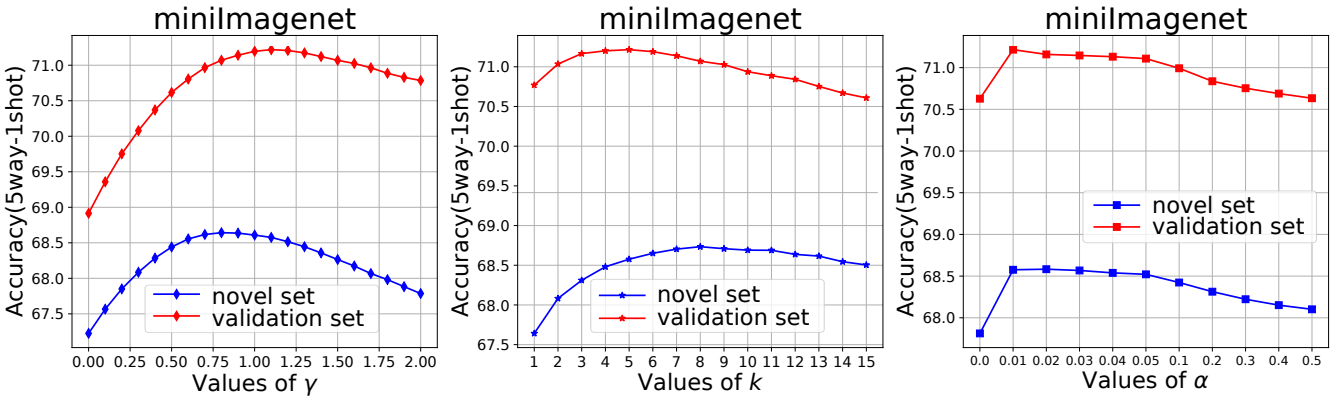

Figure 4: Left: accuracy of the model tested on the validation set (red) and novel set (blue) when increasing the weight controller. Middle: accuracy of the model tested on the validation set (red) and novel set (blue) when increasing the number of matched base classes. Right: accuracy of the model tested on the validation set (red) and novel set (blue) when increasing the compensation for the within-class variation in few-shot classes.

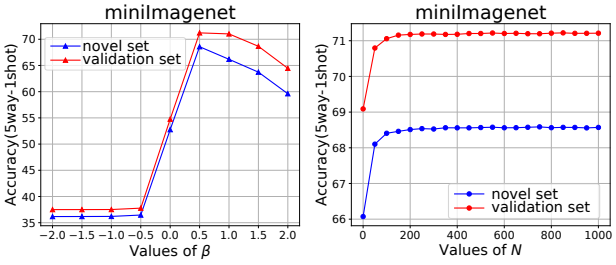

Figure 5: Left: accuracy of the model tested on the validation set (red) and novel set (blue) when increasing the power in Tukey's transformation. Right: accuracy of the model tested on the validation set (red) and novel set (blue) when increasing the number of generated features.

accuracies. An example of hyperparameter tuning on mini-Imagenet is shown in Fig. 4 and Fig. 5. It can be seen that the curves of the validation set (red) and novel set (blue) share the same α ($\alpha = 0.01$), β ($\beta = 0.5$), and N ($N = 550$) that reach the highest accuracy. In the right slide of Fig. 5, the performance consistently improves as the number of generated features grows on both the validation set (red) and novel set (blue). And the accuracy remains stable when the number of generated features exceeds 200. In Fig. 4 and Fig. 5, the curves of each hyperparameter have a similar tendency on the validation set and novel (testing) set, indicating that our model does not overfit on the validation set.

Conclusion

This paper proposes the Direction-Driven Weighting Method (DDWM) for few-shot image classification, which makes the biased feature distributions of few-shot classes precisely fit the corresponding ground-truth distributions. DDWM can describe the distribution in multiple directions for more distribution information like the within-class variation. Theoretical explanation and analysis are given in the paper. The experimental results show that DDWM outperforms the state-of-the-art result by an average of 3% for the 5-way-1-shot setting on *mini*ImageNet, CIFAR-FS, and CUB. The visualization of the generated features proves that the DDWM can more accurately estimate the ground-truth distribution.

Acknowledgments

This work was supported by the National Natural Science Foundation of China (Grant No. 62106093, 62076117, 62106090), Jiangxi Key Laboratory of Smart City (Grant No. 20192BCD40002), and the Urgent Need for Overseas Talent project (Grant No. 20223BCJ25040, 20223BCJ25026).

References

Afrasiyabi, A.; Larochelle, H.; Lalonde, J.-F.; and Gagné, C. 2022. Matching Feature Sets for Few-Shot Image Classification. In *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 9014–9024.

Bendre, N.; Desai, K.; and Najafirad, P. 2021. Generalized Zero-Shot Learning Using Multimodal Variational Auto-Encoder With Semantic Concepts. In *2021 IEEE International Conference on Image Processing*, 1284–1288.

Bertinetto, L.; Henriques, J. F.; Torr, P.; and Vedaldi, A. 2019. Meta-learning with differentiable closed-form solvers. In *Proceedings of the Seventh International Conference on Learning Representations*.

Chen, W.-Y.; Liu, Y.-C.; Kira, Z.; Wang, Y.-C. F.; and Huang, J.-B. 2019a. A Closer Look at Few-shot Classification. In *Proceedings of the Seventh International Conference on Learning Representations*.

Chen, Z.; Fu, Y.; Zhang, Y.; Jiang, Y.-G.; Xue, X.; and Sigal, L. 2019b. Multi-Level Semantic Feature Augmentation for One-Shot Learning. *IEEE Transactions on Image Processing*, 28(9): 4594–4605.

Finn, C.; Abbeel, P.; and Levine, S. 2017. Model-Agnostic Meta-Learning for Fast Adaptation of Deep Networks. In Precup, D.; and Teh, Y. W., eds., *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, 1126–1135. PMLR.

Goodfellow, I.; Pouget-Abadie, J.; Mirza, M.; Xu, B.; Warde-Farley, D.; Ozair, S.; Courville, A.; and Bengio, Y. 2014. Generative Adversarial Nets. In Ghahramani, Z.; Welling, M.; Cortes, C.; Lawrence, N.; and Weinberger, K., eds., *Advances in Neural Information Processing Systems*, volume 27. Curran Associates, Inc.

Hong, Y.; Niu, L.; Zhang, J.; and Zhang, L. 2020. Matchinggan: Matching-Based Few-Shot Image Generation. In *2020 IEEE International Conference on Multimedia and Expo*, 1–6.

Hou, R.; Chang, H.; MA, B.; Shan, S.; and Chen, X. 2019. Cross Attention Network for Few-shot Classification. In Wallach, H.; Larochelle, H.; Beygelzimer, A.; d'Alché-Buc, F.; Fox, E.; and Garnett, R., eds., *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc.

Kang, D.; Kwon, H.; Min, J.; and Cho, M. 2021. Relational Embedding for Few-Shot Classification. In *2021 IEEE/CVF International Conference on Computer Vision*, 8802–8813. Los Alamitos, CA, USA: IEEE Computer Society.

Krizhevsky, A.; and Hinton, G. 2009. Learning multiple layers of features from tiny images. Technical Report 0, University of Toronto, Toronto, Ontario.

Lee, K.; Maji, S.; Ravichandran, A.; and Soatto, S. 2019. Meta-Learning With Differentiable Convex Optimization. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 10649–10657. Los Alamitos, CA, USA: IEEE Computer Society.

Li, K.; Zhang, Y.; Li, K.; and Fu, Y. 2020. Adversarial Feature Hallucination Networks for Few-Shot Learning. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 13467–13476.

Li, Z.; Zhou, F.; Chen, F.; and Li, H. 2017. Meta-SGD: Learning to Learn Quickly for Few-Shot Learning. arXiv:1707.09835.

Liu, B.; Cao, Y.; Lin, Y.; Li, Q.; Zhang, Z.; Long, M.; and Hu, H. 2020. Negative Margin Matters: Understanding Margin in Few-Shot Classification. In Vedaldi, A.; Bischof, H.; Brox, T.; and Frahm, J.-M., eds., *Computer Vision – ECCV 2020*, 438–455. Cham: Springer International Publishing. ISBN 978-3-030-58548-8.

Liu, Y.; Schiele, B.; and Sun, Q. 2020. An Ensemble of Epoch-Wise Empirical Bayes for Few-Shot Learning. In Vedaldi, A.; Bischof, H.; Brox, T.; and Frahm, J.-M., eds., *Computer Vision – ECCV 2020*, 404–421. Cham: Springer International Publishing. ISBN 978-3-030-58517-4.

Ma, R.; Fang, P.; Drummond, T.; and Harandi, M. 2022. Adaptive Poincaré Point to Set Distance for Few-Shot Classification. *Proceedings of the Thirty-Sixth AAAI Conference on Artificial Intelligence*, 36(2): 1926–1934.

Mangla, P.; Singh, M.; Sinha, A.; Kumari, N.; Balasubramanian, V. N.; and Krishnamurthy, B. 2020. Charting the Right Manifold: Manifold Mixup for Few-shot Learning. In *2020 IEEE Winter Conference on Applications of Computer Vision*, 2207–2216. Los Alamitos, CA, USA: IEEE Computer Society.

Munkhdalai, T.; Yuan, X.; Mehri, S.; and Trischler, A. 2018. Rapid Adaptation with Conditionally Shifted Neurons. In Dy, J.; and Krause, A., eds., *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, 3664–3673. PMLR.

Park, S.-J.; Han, S.; Baek, J.-W.; Kim, I.; Song, J.; Lee, H. B.; Han, J.-J.; and Hwang, S. J. 2020. Meta Variance Transfer: Learning to Augment from the Others. In III, H. D.; and Singh, A., eds., *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, 7510–7520. PMLR.

Raghu, A.; Raghu, M.; Bengio, S.; and Vinyals, O. 2020. Rapid Learning or Feature Reuse? Towards Understanding the Effectiveness of MAML. In *Proceedings of the Eighth International Conference on Learning Representations*.

Ravichandran, A.; Bhotika, R.; and Soatto, S. 2019. Few-Shot Learning With Embedded Class Models and Shot-Free Meta Training. In *2019 IEEE/CVF International Conference*

on Computer Vision, 331–339. Los Alamitos, CA, USA: IEEE Computer Society.

Rumelhart, D. E.; Hinton, G. E.; and Williams, R. J. 1986. Learning representations by back-propagating errors. *nature*, 323(6088): 533–536.

Russakovsky, O.; Deng, J.; Su, H.; Krause, J.; Satheesh, S.; Ma, S.; Huang, Z.; Karpathy, A.; Khosla, A.; Bernstein, M.; et al. 2015. ImageNet Large Scale Visual Recognition Challenge. *International Journal of Computer Vision*, 115(3): 211–252.

Rusu, A. A.; Rao, D.; Sygnowski, J.; Vinyals, O.; Pascanu, R.; Osindero, S.; and Hadsell, R. 2019. Meta-Learning with Latent Embedding Optimization. In *Proceedings of the Thirty-Sixth International Conference on Learning Representations*.

Schwartz, E.; Karlinsky, L.; Shtok, J.; Harary, S.; Marder, M.; Kumar, A.; Feris, R.; Giryes, R.; and Bronstein, A. 2018. Delta-encoder: an effective sample synthesis method for few-shot object recognition. In Bengio, S.; Wallach, H.; Larochelle, H.; Grauman, K.; Cesa-Bianchi, N.; and Garnett, R., eds., *Advances in Neural Information Processing Systems*, volume 31. Curran Associates, Inc.

Snell, J.; Swersky, K.; and Zemel, R. 2017. Prototypical Networks for Few-shot Learning. In Guyon, I.; Luxburg, U. V.; Bengio, S.; Wallach, H.; Fergus, R.; Vishwanathan, S.; and Garnett, R., eds., *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc.

Sun, Q.; Liu, Y.; Chua, T.; and Schiele, B. 2019. Meta-Transfer Learning for Few-Shot Learning. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 403–412. Los Alamitos, CA, USA: IEEE Computer Society.

Sung, F.; Yang, Y.; Zhang, L.; Xiang, T.; Torr, P. S.; and Hospedales, T. M. 2018. Learning to Compare: Relation Network for Few-Shot Learning. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 1199–1208. Los Alamitos, CA, USA: IEEE Computer Society.

Tian, Y.; Wang, Y.; Krishnan, D.; Tenenbaum, J. B.; and Isola, P. 2020. Rethinking Few-Shot Image Classification: A Good Embedding is All You Need? In Vedaldi, A.; Bischof, H.; Brox, T.; and Frahm, J.-M., eds., *Computer Vision – ECCV 2020*, 266–282. Cham: Springer International Publishing. ISBN 978-3-030-58568-6.

Tukey, J. W. 1977. *Exploratory data analysis*. Addison-Wesley Series in Behavioral Science: Quantitative Methods.

van der Maaten, L.; and Hinton, G. 2008. Visualizing Data using t-SNE. *Journal of Machine Learning Research*, 9(86): 2579–2605.

Vinyals, O.; Blundell, C.; Lillicrap, T.; kavukcuoglu, k.; and Wierstra, D. 2016. Matching Networks for One Shot Learning. In Lee, D.; Sugiyama, M.; Luxburg, U.; Guyon, I.; and Garnett, R., eds., *Advances in Neural Information Processing Systems*, volume 29. Curran Associates, Inc.

Wah, C.; Branson, S.; Welinder, P.; Perona, P.; and Belongie, S. 2011. The Caltech-ucsd Birds-200-2011 Dataset. Technical report, California Institute of Technology.

Yang, S.; Wu, S.; Liu, T.; and Xu, M. 2021. Bridging the Gap between Few-Shot and Many-Shot Learning via Distribution Calibration. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 1–1.

Zhang, J.; Liu, W.; Zhao, Z.; and Su, F. 2021. Distribution Estimation Based Pseudo-Feature Library Generation For Few-Shot Image Classification. In *2021 IEEE International Conference on Multimedia and Expo Workshops*, 1–6.

Zhang, J.; Zhao, C.; Ni, B.; Xu, M.; and Yang, X. 2019. Variational Few-Shot Learning. In *2019 IEEE/CVF International Conference on Computer Vision*, 1685–1694.

ZHANG, R.; Che, T.; Ghahramani, Z.; Bengio, Y.; and Song, Y. 2018. MetaGAN: An Adversarial Approach to Few-Shot Learning. In Bengio, S.; Wallach, H.; Larochelle, H.; Grauman, K.; Cesa-Bianchi, N.; and Garnett, R., eds., *Advances in Neural Information Processing Systems*, volume 31. Curran Associates, Inc.