Few-Shot Classification With Feature Reconstruction Bias

Zhen Li*, Lang Wang*, Shuo Ding*, Xiaochen Yang†, Xiaoxu Li*
* Lanzhou University of Technology, Lanzhou, China
E-mail: lixiaoxu@lut.edu.cn
† School of Mathematics and Statistics, University of Glasgow, UK.
E-mail: xiaochen.yang@glasgow.ac.uk

***Abstract*—Few-shot classification aims to classify unseen samples by learning from very few labeled samples. Very recently, reconstruction-based methods have been proposed and shown superior performance on few-shot fine-grained image classification, which, on top of the challenge of few labeled samples, faces the difficulty of identifying subtle differences between subcategories. In essence, these methods reconstruct unseen samples from few seen samples and use the distance between the original unseen samples and their reconstruction as the criterion for classification. However, as pointed out in this paper, a bias is introduced in the overall distribution between the reconstructed features and original features, which consequently affects the distance calculation and subsequent classification. To address this issue, we propose a new concept of Feature Reconstruction Bias (FRB), which can be computed easily in the training stage without introducing any new parameters. Moreover, we propose to use this bias to correct query features in the test stage, which is shown to increase inter-class distances and decrease intra-class distances. Experiments on four fine-grained benchmarks demonstrate the effectiveness of our approach, with state-of-the-art performance achieved in most scenarios.**

I. Introduction

Image classification is the core problem of computer vision research, which has broad prospects and extraordinary significance in many fields. However, traditional image classification tasks rely on a large amount of labeled data, which is in stark contrast with human intelligence which has the ability to learn from very few examples. Moreover, it can be difficult to acquire sufficient images, and labeling these images can be time-consuming and even more challenging, sometimes requiring expert knowledge. This is particularly the case for fine-grained image classification, whose task is to distinguish images of the same basic-level category, such as classifying dog images into breeds and flower images into species. Therefore, the problem of few-shot fine-grained image classification has gradually attracted attention, and more and more researchers have started to investigate how to learn discriminative features to identify the subtle differences in fine-grained images based on a small number of labeled samples.

Among existing research on few-shot image classification, feature-based methods and metric-based methods are the mainstream. Feature-based methods focus on learning transferable features from an auxiliary dataset with a large amount of labeled images [2, 3]. To further improve the discriminability for the novel task, features may be further fine-tuned on few labeled samples [4, 5]. Metric-based methods aim to find an effective metric space based on the auxiliary dataset with the goal that samples of the same class, even unseen before, are close to each other and samples of different classes are far apart [6, 7]. For fine-grained image classification, one distinction from traditional image classification is that useful information for distinguishing between classes may only lie in a small region of the image and at different positions across images. Therefore, methods based on local descriptors have been proposed, which extract and pool together local features from different parts of the labeled image to serve as the class representation and classify an unseen image by comparing it local features against this pool [8, 9]. Building on local features, Wertheimer et al. [1] recently proposed the feature

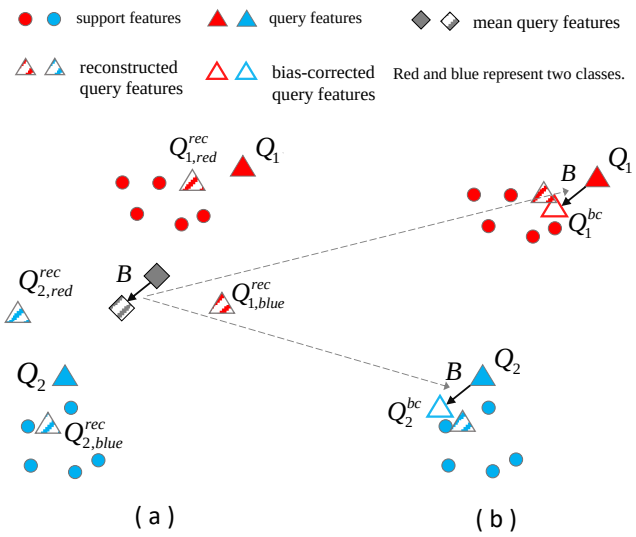

Fig. 1. (a) is the existing feature map reconstruction network (FRN) [1]. There exists a bias B between the distribution of query features, represented by the mean function (the solid diamond), and the distribution of reconstructed features (represented by the shaded diamond). (b) is the proposed feature reconstruction bias (FRB) module. The bias B is transferred and used to correct query features, which consequently reduces the distance between reconstructed features (shaded triangles) and bias-corrected features (hollow triangles).

map reconstruction network (FRN). One key novelty of FRN is the idea of reconstructing unseen query features from seen support features and then using the distance between original query features and reconstructed query features as the classification criterion. The reconstruction step, based on a mathematically elegant formula, introduces only two parameters and permits a closed-form solution, thereby having the virtues of computational efficiency and generalizability.

However, despite the aforementioned benefits and empirically strong performance, we notice from experiments that, there exists a bias between the distributions of original query features and the reconstructed features in FRN. This issue is illustrated in Figure 1(a), where the arrow B indicates the gap between mean query features (denoted by the solid diamond) and mean reconstructed features (denoted by the shaded diamond); here the mean function is used to represent the overall distribution of features. This bias is a result of insufficient labeled samples and it will be passed to the distance calculation and affect the subsequent classification. Again, in Figure 1(a), we see that the query sample Q_2 has similar distances to the correct-class reconstruction, $Q^{rec}_{2,blue}$ and incorrect-class reconstruction, $Q^{rec}_{2,red}$, implying the difficulty to classify Q_2 correctly. To address this bias, we introduce a novel concept of Feature Reconstruction Bias (FRB), which can be computed easily in the training without introducing any new parameter. Based on this concept, we propose a simple yet effective remedy to correct query samples. As illustrated in Figure 1(b), after adding the bias B to the query features Q_1 and Q_2, the bias-corrected query features Q^{bc}_1 and Q^{bc}_2 locate closer to their correct-class reconstruction, thus improving the classification performance.

To summarize, the contribution of our work are three-fold:

- We propose a novel concept called feature reconstruction bias (FRB), which represents the distributional difference between the original query features and the reconstructed query features.
- We propose to transfer the optimal bias learned in the training stage to query features in the test stage, which effectively reduces the intra-class distances and results in better classification performance.
- We verify the effectiveness of the proposed approach on four fine-grained image classification datasets. Results show that our method achieves the state-of-the-art performance in most scenarios.

II. Related Work

In recent years, many methods for few-shot image classification have been proposed. In this section, we provide a brief review of metric-based methods and feature learning methods.

A. Metric-Based Few-shot Methods

As an important method to study few-shot learning, the metric-based method calculates the distance between features of the samples to be classified and features of labeled samples, and then determines the class of the samples to be classified by this distance.

Koch et al. [10] were the first to use Siamese neural networks for few-shot image classification tasks. The model maps the input to the target space through an embedding function and uses a simple distance function to calculate the similarity between samples. Matching Network [6] introduced the attention mechanism and memory mechanism LSTM for one-shot learning for fast learning, and finally used the cosine similarity for class prediction. Snell et al. [7] proposed the Prototypical Network, which calculates the distance between the sample to be classified and the prototype representation of the support set by the Euclidean distance, and realizes the classification according to this distance.

The above methods all use a predefined function to calculate the similarity score between samples. In addition, there are methods using neural networks for measurement. Relation Network [11] obtains a learnable non-linear similarity measure by training a neural network to compare the feature vectors extracted by the embedding module. Li et al. [8] proposed a Deep Nearest Neighbor Neural Network (DN4), which replaces simple image feature vectors with local descriptors and finds the class closest to the input image by comparing the local descriptors between the samples to be classified and the known samples. The method is particularly suitable for fine-grained image classification.

B. Feature Learning Methods

Feature learning methods mainly use related information or auxiliary information of the image to extract more discriminative features.

Oreshkin et al. [12] noticed that in classification task, there is a strong correlation between multiple classes of the same task, and adjusted the feature extractor according to the common features of multiple classes in the task. Gidaris et al. [13] proposed an approach for improving few-shot learning through self-supervision. This method combines an additional self-supervised learning task with a few-shot classification task, both sharing a feature extraction network, with the hope that the self-supervised learning task can improve the representational ability of feature extraction networks. In order to establish the connection between training and novel class, Zhu et al. [14] proposed an attribute-guided learning framework, which in addition to learning general feature representations can also establish an attribute-guided learning mechanism, thereby obtaining a more discriminative feature representations.

In addition to the above works, there are many studies aiming to improve feature learning by augmenting data or features. Ji et al. [15] proposed an unsupervised feature learning method. The method consists of two alternating processes of progressive clustering and episodic training. By generating pseudo-labeled training examples and optimizing the feature representation of the data, it obtains a high-performance few-shot learner. Li et al. [16] proposed

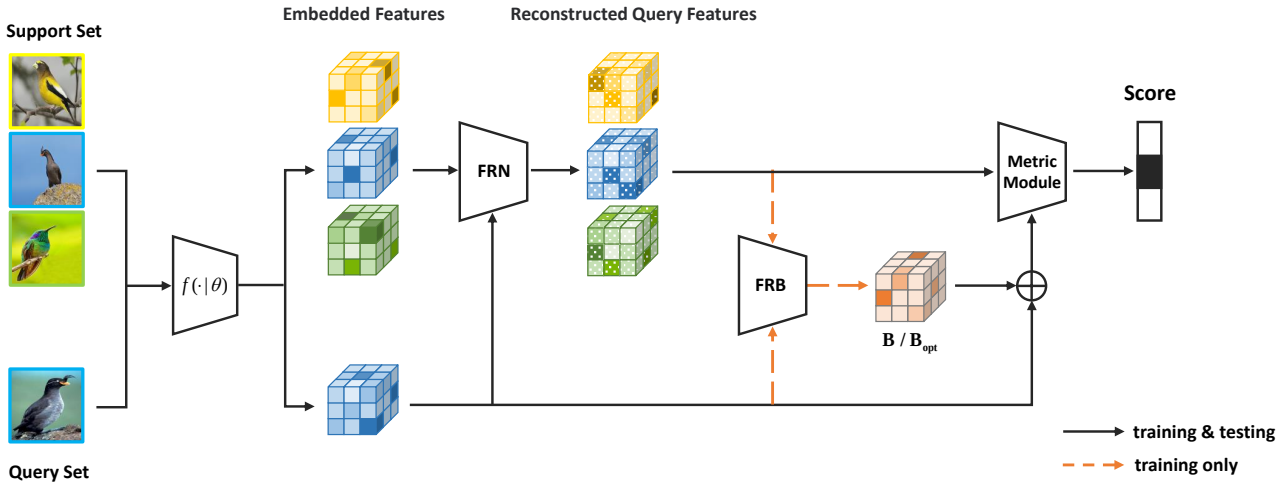

Fig. 2. Model architecture of the proposed method. Images are converted into embedded features by the embedding module $f(\cdot|\theta)$, and the query features are reconstructed using support features from each class through the feature map reconstruction network (FRN) module. The distribution bias B between the embedded query features and the reconstructed features is obtained by the feature reconstruction bias (FRB) module and used to correct query features. The corrected features and reconstructed features are compared in the metric module to generate a prediction score. In the training phase, the bias B is computed with the optimal one saved as B_{opt}; the testing phase will use B_{opt} directly instead of retraining through FRB.

Adversarial Feature Hallucination Networks (AFHN), which is based on conditional Wasserstein Generative Adversarial networks (cWGAN) under the condition of a small number of labeled samples to hallucinate diverse and discriminative features.

By combining metric-based and feature learning methods in recent years, various few-shot classification methods have emerged, bringing many breakthroughs to the few-shot research. However, there are still numerous problems to be solved.

III. Proposed Method

In this section, we present our method on identifying and correcting the bias in the distribution of embedded query features and reconstructed query features. We will start with a formulation of few-shot classification, followed by an overview of the proposed network and a detailed description of each component.

A. Problem Formulation

Few-shot classification typically involves three datasets: a support set $\mathcal{S} = \{(x_i, y_i)\}_{i=1}^{n} (n = C \times K)$, a query set $\mathcal{Q} = \{(x_j, y_j)\}_{j=1}^{m} (m = C \times M)$, and an auxiliary set $\mathcal{A}$. The support and query sets serve as the training and test sets in the traditional machine learning setting and they share the same label space. In addition, the support set contains K labeled images for each of the C classes, and thus few-shot classification is generally regarded as an C-way K-shot classification problem (K is a small number, e.g., 1 or 5); the query set contains $N \times M$ images from the same classes for testing. Since the support set has very limited labeled images and training on it can easily cause overfitting, an auxiliary set with a large number of labeled samples is typically used to learn transferable knowledge. Note that the label space of the auxiliary set is disjoint with the support and query sets.

A widely adopted approach to learn transferable knowledge is through episodic training [6]. Each episode mimics the C-way K-shot classification task, i.e., randomly selecting K and M images of C classes from the auxiliary set to form the support and query sets respectively. Moreover, the auxiliary set can be further split into training and validation sets with disjoint label space. Episodes formed from the training set is used to train the model, and the learned model is evaluated on episodes formed from the validation set for selecting hyperparameters. Once the optimal model is selected, it will be tested on the actual support and query sets to report the final performance.

B. Architecture Overview

As shown in Figure 2, the proposed method includes four modules. The first is the feature embedding module $f(\cdot \mid \theta)$, which extracts features from the input images. Networks such as a convolutional neural network and a residual network can be used for this purpose. The second is the feature map reconstruction network (FRN) module, which reconstructs query features from support features. The third module is the feature reconstruction bias (FRB) module, which calculates the overall distribution bias between reconstructed query features and embedded query features. In the training phase, the bias, B, is calculated for each episode and the one which obtains the optimal performance on the validation set is saved as B_{opt}. In the testing phase, the saved optimal bias B_{opt} will be added to each query feature, producing bias-corrected query features. The fourth module is a metric module, which performs classification by measuring the distance between reconstructed query features and bias-corrected query features.

C. Feature Map Reconstruction Network (FRN) Module

The FRN module module mainly adopts the existing feature map reconstruction networks [1].

The input to the FRN module is the features extracted by the feature embedding module. Given an image x_i, the feature embedding module extracts features and outputs $\hat{x}_i = f(x_i|\theta) \in \mathbb{R}^{d\times h\times w}$, where d denotes the number of channels, and h and w denote the height and width of feature maps respectively.

The FRN module first reshapes the features into r ($r = h \times w$) d-dimensional local descriptors, i.e., $\hat{x}_i = [\hat{x}_{i,(1)}, \ldots, \hat{x}_{i,(r)}]$, and pools together local descriptors of the same class to represent a class. In other words, for a C-way K-shot classification task, each class c is represented by a single matrix $S_c \in \mathbb{R}^{Kr\times d}$. Similarly, a query image is transformed into a matrix $Q \in \mathbb{R}^{r\times d}$.

Next, the FRN module reconstructs the query feature from the support features by utilizing ridge regression. According to the following two formulae, we can reconstruct query features Q_i using support features from class c (details can be found in [1]):

$$Q_{c,i}^{\text{rec}} = \rho Q_i S_c^T \left(S_c S_c^T + \lambda I\right)^{-1} S_c \tag{1}$$

$$Q_{c,i}^{\text{rec}} = \rho Q \left(S_c^T S_c + \lambda I\right)^{-1} S_c^T S_c, \tag{2}$$

where ρ and λ are learnable parameters. Meanwhile, in order to ensure their non-negative properties, ρ and λ are represented as follows:

$$\lambda = \frac{Kr}{d} e^{\alpha} \tag{3}$$

$$\rho = e^{\beta}, \tag{4}$$

where α and β are learnable hyperparameters with an initial value of zero. When $d > Kr$, it is recommended to use Eq. (3) to calculate the reconstructed feature for computational efficiency; otherwise, Eq. (4) is recommended.

Once the query features are reconstructed, they are reshaped back to a tensor with the same shape as before, i.e., a tensor of size $h \times w \times d$. These reconstructed features will be used in the FRB module and the metric module.

D. Feature Reconstruction Bias (FRB) Module

As discussed in the introduction and shown in Figure 1, there exists a bias between the distribution of reconstructed query features and the distribution of the embedded query features. Such bias will be brought forward in the calculation of the Euclidean distance between the embedded query features and class-wise reconstructed features, which is then used to make predictions. Therefore, to overcome the influence of this bias, we propose the FRB module which aims to learn the bias from training episodes and use the learned bias to correct embedded query features in the testing phase.

To represent the feature distribution, we follow the prototypical network [7] and compute the first-order statistic. In other words, we define the bias between the distribution of reconstructed query features and embedded query features to be the difference between their mean functions:

$$\begin{aligned} B &= \bar{Q}^{\text{rec}} - \bar{Q} \\ &= \frac{1}{Cm}\sum_{c=1}^{C}\sum_{i=1}^{m} Q_{c,i}^{\text{rec}} - \frac{1}{Cm}\sum_{c=1}^{C}\sum_{i=1}^{m} Q_{c,i} \end{aligned} \tag{5}$$

After computing the bias, it is added directly to the original query features to generate bias-corrected query features:

$$Q_i^{\text{bc}} = Q_i + B \tag{6}$$

In the training phase, the bias B is calculated using the reconstructed and embedded query features from the current task as in Eq. (5). Meanwhile, we evaluate the current model performance on the validation set so as to save the bias that generates the highest validation accuracy; this optimal bias is denoted by B_{opt}. During the testing phase, instead of recalculating B, we add this optimal bias to the embedded query features, i.e., replacing B by B_{opt} in Eq. (6). The reason is that, in reality, we may not have the query images available all at once and thus cannot calculate Eq. (5).

The final classification is performed based on the squared Euclidean distance between the reconstructed query features and bias-corrected query features:

$$d_{c,i} = \|Q_{c,i}^{\text{rec}} - Q_i^{\text{bc}}\|^2 \tag{7}$$

Applying the softmax function to the above distance, the final predicted probability is obtained as follows:

$$P\left(y_i = c \mid x_i\right) = \frac{\exp\left(-\tau d_{c,i}\right)}{\sum_{c'\in C}\exp\left(-\tau d_{c',i}\right)}, \tag{8}$$

where τ is a learnable temperature hyperparameter.

We train the model by minimizing the cross-entropy loss:

$$L = -\frac{1}{m}\sum_{i=1}^{m}\left(\boldsymbol{y}_i^T \log\left(\boldsymbol{p}_i\right)\right), \tag{9}$$

where $\boldsymbol{y}_i$ denotes the one-hot vector and $\boldsymbol{p}_i$ denotes the vector of predicted probability.

IV. EXPERIMENTS

A. Datasets and Preprocessing

To test the effectiveness of the proposed method, we use four benchmark fine-grained datasets: CUB-200-2011, Stanford-Cars, Stanford-Dogs and Oxford 102 Flower Dataset. A brief description of the datasets is given as follows:

CUB-200-2011 dataset (CUB) [26]: a widely-used fine-grained dataset for bird classification. Number of categories: 200; number of images: 11788; annotations per image: 15 part locations, 312 binary attributes, 1 bounding box. The size of images in CUB ranges from about 200×200 to about 500×500 pixels.

Stanford-Cars dataset (Cars) [27]: a car image dataset,

TABLE I

RESULTS OF 5-WAY 1-SHOT AND 5-WAY 5-SHOT CLASSIFICATION WITH THE CONV-4 BACKBONE ON THE CUB, DOGS, CARS, AND FLOWERS DATA. MEAN ACCURACY AND 95% CONFIDENCE INTERVALS COMPUTED OVER 10,000 TASKS ARE REPORTED. THE BEST (SECOND-BEST, RESP.) METHOD IS HIGHLIGHTED IN BOLD (UNDERLINED, RESP.). METHODS LABELED BY † DENOTE OUR IMPLEMENTATIONS.

Method	CUB		Dogs		Cars		Flowers	
	1-shot	5-shot	1-shot	5-shot	1-shot	5-shot	1-shot	5-shot
MatchingNet(NerulPS '16) [6]†	60.06±0.88	74.57±0.73	46.10±0.86	59.79±0.72	44.73±0.77	64.74±0.72	71.89±0.90	85.46±0.59
ProtoNet(NerulPS '17) [7]†	63.64±0.23	84.23±0.15	45.12±0.21	69.16±0.16	48.42±0.22	71.38±0.18	64.23±0.23	84.97±0.16
RelationNet(CVPR '19) [11]†	63.94±0.92	77.87±0.64	47.35±0.88	66.20±0.74	46.04±0.91	68.52±0.78	69.50±0.96	83.91±0.63
DN4(CVPR '19) [8]	57.45±0.89	84.41±0.58	39.08±0.76	69.81±0.69	34.12±0.68	87.47±0.47	71.15±0.94	88.86±0.56
Baseline++(CVPR '19) [17]†	62.36±0.84	79.08±0.61	44.49±0.70	64.48±0.66	46.82±0.76	68.20±0.72	70.54±0.84	86.63±0.58
DeepEMD(CVPR '20) [18]	64.08±0.50	80.55±0.71	46.73±0.49	65.74±0.63	61.63±0.27	72.95±0.38	-	-
DSN(CVPR '20) [19]†	71.57±0.92	83.51±0.60	44.33±0.81	60.04±0.68	48.16±0.86	60.77±0.75	67.71±0.92	84.58±0.70
LRPABN(TMM '21) [9]	63.63±0.77	76.06±0.58	45.72±0.75	60.94±0.66	60.28±0.76	73.29±0.58	-	-
TOAN(TCSVT '21) [20]	65.34±0.75	80.43±0.60	49.30±0.77	67.16±0.49	**65.90±0.72**	84.24±0.48	-	-
BSNet(TIP '21) [21]†	62.84±0.95	85.39±0.56	43.42±0.89	71.90±0.68	40.89±0.77	86.88±0.50	72.79±0.91	84.93±0.64
FRN(CVPR '21) [1]†	73.63±0.21	**88.50±0.13**	58.23±0.22	76.50±0.17	64.42±0.22	84.17±0.13	73.61±0.22	88.93±0.14
MixFSL(ICCV '21) [22]†	53.61±0.88	73.24±0.75	43.96±0.77	64.43±0.68	44.56±0.80	59.63±0.79	68.01±0.90	85.10±0.62
DLG(IJMLC '22) [23]	64.77±0.90	83.31±0.55	47.77±0.86	67.07±0.72	62.56±0.82	**88.98±0.47**	-	-
Ours	**73.80±0.21**	88.21±0.13	**58.40±0.22**	**76.86±0.15**	65.00±0.22	84.17±0.13	**74.49±0.23**	**89.59±0.13**

TABLE II

RESULTS OF 5-WAY 1-SHOT AND 5-WAY 5-SHOT CLASSIFICATION WITH THE RESNET-12 BACKBONE ON THE CUB, DOGS, CARS, AND FLOWERS DATA. MEAN ACCURACY AND 95% CONFIDENCE INTERVALS COMPUTED OVER 10,000 TASKS ARE REPORTED. THE BEST (SECOND-BEST, RESP.) METHOD IS HIGHLIGHTED IN BOLD (UNDERLINED, RESP.). METHODS LABELED BY † DENOTE OUR IMPLEMENTATIONS.

Method	CUB		Dogs		Cars		Flowers	
	1-shot	5-shot	1-shot	5-shot	1-shot	5-shot	1-shot	5-shot
ProtoNet(NerulPS '17) [7]†	79.64±0.20	91.15±0.11	72.85±0.22	86.54±0.13	82.29±0.20	93.11±0.10	75.41±0.22	89.46±0.14
DeepEMD(CVPR '20) [18]†	71.11±0.31	86.30±0.19	67.59±0.30	81.13±0.20	73.30±0.29	88.37±0.17	70.00±0.35	83.63±0.26
RENet(ICCV '21) [24]†	79.49±0.44	91.11±0.24	71.69±0.47	85.60±0.30	79.66±0.44	91.95±0.22	79.91±0.42	92.33±0.22
MixFSL(ICCV '21) [22]†	67.86±0.94	82.12±0.66	67.26±0.90	82.05±0.56	58.15±0.87	80.54±0.63	72.60±0.91	86.52±0.65
FRN(CVPR '21) [1]†	**83.16±0.19**	92.59±0.11	76.49±0.21	88.22±0.12	86.48±0.18	94.78±0.08	81.07±0.20	92.52±0.11
VFD(CVPR '21) [25]	79.12±0.83	91.48±0.39	76.24±0.87	88.00±0.47	-	-	-	-
Ours	83.06±0.19	**92.92±0.10**	**76.55±0.21**	**88.27±0.12**	**86.94±0.18**	**95.59±0.07**	**82.77±0.19**	**93.96±0.10**

which is also widely used for fine-grained classification. Number of categories: 196, number of images: 16185, containing make, model and year information.

Stanford-Dogs dataset (Dogs) [28]: is a dataset of dog images, which was originally collected for fine-grained image classification. Number of categories: 120, number of pictures: 20580. This dataset is an image of 120 species of dogs from around the world, built using ImageNet images and annotations.

Oxford 102 Flower Dataset (Flowers) [29]: a dataset of flower images. Number of categories: 102, number of pictures: 8189. Each flower consists of 40 to 258 images, and these flowers are dominated by common British flowers.

For all datasets, images are resized to a widely used size: 84×84.

B. Implementation Details

Following best practices in few-shot classification, we adopt both conv4 and ResNet-12 backbone for our experiments. All experiments are implemented by using Pytorch on one NVIDIA RTX 3090 GPU.

Conv-4 consists of four convolution blocks with each block composed of 3×3 convolutions with 64 channels, batch normalization, a ReLU nonlinearity, and 2×2 max-pooling. The shape of the output feature maps for input images with size 84×84 is 64×5×5. ResNet-12 consists of four residual blocks. Each residual block is composed of three convolutional layers. The shape of output feature map is 640×5×5 for input images with a size of 84×84.

We perform experiments on the standard 5-way 1-shot and 5-way 5-shot setups on four datasets [26–29]. Finally, we

TABLE III

ABLATION STUDIES ON THE CONSTRUCTION OF THE OPTIMAL BIAS, B_{OPT} – LEARNED VIA META-LEARNING *vs* SELECTED FROM THE BEST MODEL. 5-WAY 1-SHOT AND 5-SHOT CLASSIFICATION ARE PERFORMED ON CUB, DOGS, AND FLOWERS WITH THE RESNET-12 BACKBONE.

Method	CUB		Dogs		Flowers	
	1-shot	5-shot	1-shot	5-shot	1-shot	5-shot
meta-learned	80.57	90.86	73.74	86.45	77.93	90.97
best model	83.06	92.92	76.55	88.27	82.77	93.96

report the mean accuracy and the 95% confidence intervals computed over 10,000 randomly selected tasks.

In our experiments, we also used standard data enhancements to augment labeled data to alleviate the overfitting problem of the model [1, 11].

C. Experimental Results

In this section, we compare the proposed method with 15 few-shot classification and fine-grained few-shot methods on 5-way 1-shot and 5-way 5-shot tasks. Results with Conv-4 as the backbone network are listed in Table I, and results with the ResNet-12 backbone are listed in Table II. Some methods are not evaluated on Flowers and Dogs datasets in their original papers and thus their results are missing in our tables.

With the Conv-4 backbone, we first observe that the proposed method performs better than the baseline method FRN in all cases except 5-way 5-shot on CUB, indicating the importance of correcting the distribution bias. Secondly, the proposed method outperforms other methods on Dogs and Flowers datasets in both 5-way 1-shot and 5-way 5-shot settings. On CUB, it performs the best on the 1-shot task and is very close to the optimal method on the 5-shot task. On Cars, our method is competitive on the 1-shot task and slightly worse on the 5-shot task. Overall, the method achieves the best or second-best performance in almost all cases, demonstrating its efficacy.

With the ResNet-12 backbone, our method achieves the state-of-the-art performance in both 5-way 1-shot and 5-way 5-shot settings on Dogs, Cars, and Flowers datasets. This advantage is particularly evident on the Flowers dataset, where the accuracy of our method is 1.69% higher than the runner-up FRN. On the CUB dataset, it achieves the highest accuracy in the 1-shot setting and is slightly worse than FRN in the 5-shot setting, but the gap between is very small. Again, these results verify the effectiveness of the proposed method.

D. Ablation Studies

One important component in our method is the construction of the optimal bias, B_{opt}. Currently, we use a simple approach of selecting the bias B from episode that achieves the highest validation accuracy. An alternative approach would be parameterizing B and then meta-learning these parameters on training episodes. These two strategies are compared and the result is shown in Table III. We see that the approach of using the bias from the best model achieves higher classification accuracy in all cases, favoring a simple approach with no parameter introduced.

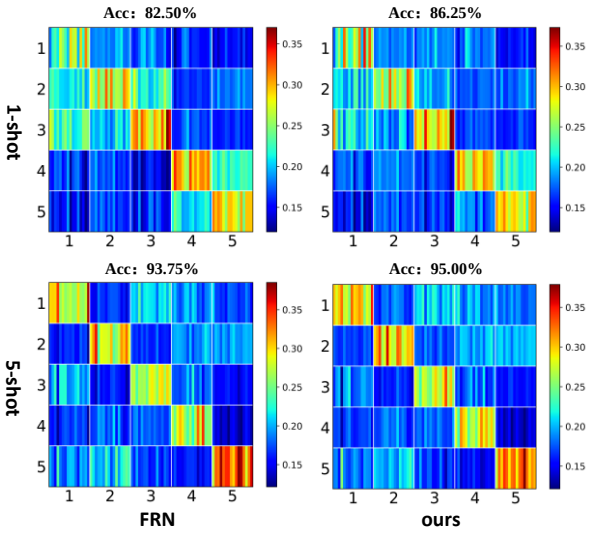

Fig. 3. Visualization of probability scores of FRN and the proposed method (ours) on Flowers dataset. Bars in the diagonal (off-diagonal, resp.) matrices indicate the probability of classifying into the correct (incorrect, resp.) class.

E. Visualization Analysis

Figure 3 provides the visualization of the probability score for FRN and the proposed method under the experimental settings of the 5-way 1-shot and 5-way 5-shot on the Flowers dataset. We randomly sample 15 query samples in the testing phase, calculate the probability score according to Eq. 8, and then display these scores in the confusion matrix. Each bar in the diagonal matrices denotes the probability of classifying a query image into the correct class and the one in the off-diagonal matrices denotes the probability of classifying into the wrong class; warmer color indicates a higher probability score. As we can see from the figure, with the proposed method, query images are classified into the correct class with higher probability. Recalling Eq. (8), this means that the distances between reconstructed and original query features are smaller in our method than FRN.

V. CONCLUSION

In this paper, we introduce a new concept to characterize the distributional bias in reconstruction-based methods. We also propose a simple approach to learn and save the optimal bias in the training phase and use it correct query features in the test phase. The resulting model can reach an advanced level with extensive experiments show that the proposed method performs competitively or outperforms existing few-shot classification methods on four fine-grained image datasets.

REFERENCES

[1] D. Wertheimer, L. Tang, and B. Hariharan, "Few-shot classification with feature map reconstruction networks,"

in *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021, pp. 8012–8021.

[2] Y. Tian, Y. Wang, D. Krishnan, J. B. Tenenbaum, and P. Isola, "Rethinking few-shot image classification: a good embedding is all you need?" in *European Conference on Computer Vision*. Springer, 2020, pp. 266–282.

[3] 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 *European Conference on Computer Vision*. Springer, 2020, pp. 438–455.

[4] X. Wang, F. Yu, R. Wang, T. Darrell, and J. E. Gonzalez, "TAFE-Net: Task-aware feature embeddings for low shot learning," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 1831–1840.

[5] H.-J. Ye, H. Hu, D.-C. Zhan, and F. Sha, "Few-shot learning via embedding adaptation with set-to-set functions," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 8808–8817.

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

[7] J. Snell, K. Swersky, and R. S. Zemel, "Prototypical networks for few-shot learning," *ArXiv*, vol. abs/1703.05175, 2017.

[8] W. Li, L. Wang, J. Xu, J. Huo, Y. Gao, and J. Luo, "Revisiting local descriptor based image-to-class measure for few-shot learning," *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 7253–7260, 2019.

[9] H. Huang, J. Zhang, J. Zhang, J. Xu, and Q. Wu, "Low-rank pairwise alignment bilinear network for few-shot fine-grained image classification," *IEEE Transactions on Multimedia*, vol. 23, pp. 1666–1680, 2021.

[10] G. R. Koch, "Siamese neural networks for one-shot image recognition," 2015.

[11] F. Sung, Y. Yang, L. Zhang, T. Xiang, P. H. S. Torr, and T. M. Hospedales, "Learning to compare: Relation network for few-shot learning," *CVPR*, pp. 1199–1208, 2018.

[12] B. N. Oreshkin, P. R. López, and A. Lacoste, "TADAM: Task dependent adaptive metric for improved few-shot learning," *ArXiv*, vol. abs/1805.10123, 2018.

[13] S. Gidaris, A. Bursuc, N. Komodakis, P. Pérez, and M. Cord, "Boosting few-shot visual learning with self-supervision," *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 8058–8067, 2019.

[14] Y. Zhu, W. Min, and S. Jiang, "Attribute-guided feature learning for few-shot image recognition," *IEEE Transactions on Multimedia*, vol. 23, pp. 1200–1209, 2021.

[15] Z. Ji, X. Zou, T. Huang, and S. Wu, "Unsupervised few-shot feature learning via self-supervised training," *Frontiers in Computational Neuroscience*, vol. 14, 2020.

[16] K. Li, Y. Zhang, K. Li, and Y. R. Fu, "Adversarial feature hallucination networks for few-shot learning," *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 13 467–13 476, 2020.

[17] W.-Y. Chen, Y.-C. Liu, Z. Kira, Y. Wang, and J.-B. Huang, "A closer look at few-shot classification," *ArXiv*, vol. abs/1904.04232, 2019.

[18] C. Zhang, Y. Cai, G. Lin, and C. Shen, "DeepEMD: Few-shot image classification with differentiable earth mover's distance and structured classifiers," *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 12 200–12 210, 2020.

[19] C. Simon, P. Koniusz, R. Nock, and M. T. Harandi, "Adaptive subspaces for few-shot learning," *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 4135–4144, 2020.

[20] H. Huang, J. Zhang, L. Yu, J. Zhang, Q. Wu, and C. Xu, "TOAN: Target-oriented alignment network for fine-grained image categorization with few labeled samples," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 32, no. 2, pp. 853–866, 2022.

[21] X. Li, J. Wu, Z. Sun, Z. Ma, J. Cao, and J.-H. Xue, "BSNet: Bi-similarity network for few-shot fine-grained image classification," *IEEE Transactions on Image Processing*, vol. 30, pp. 1318–1331, 2021.

[22] A. Afrasiyabi, J.-F. Lalonde, and C. Gagn'e, "Mixture-based feature space learning for few-shot image classification," *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 9021–9031, 2021.

[23] S. Cao, W. Wang, J. Zhang, M. Zheng, and Q. Li, "A few-shot fine-grained image classification method leveraging global and local structures," *Int. J. Mach. Learn. Cybern.*, vol. 13, pp. 2273–2281, 2022.

[24] D. Kang, H. Kwon, J. Min, and M. Cho, "Relational embedding for few-shot classification," in *IEEE/CVF International Conference on Computer Vision*, 2021.

[25] J. Xu, H. M. Le, M. Huang, S. Athar, and D. Samaras, "Variational feature disentangling for fine-grained few-shot classification," *IEEE/CVF International Conference on Computer Vision*, pp. 8792–8801, 2021.

[26] C. Wah, S. Branson, P. Welinder, P. Perona, and S. J. Belongie, "The Caltech-UCSD birds-200-2011 dataset," 2011.

[27] J. Krause, M. Stark, J. Deng, and L. Fei-Fei, "3D object representations for fine-grained categorization," *2013 IEEE International Conference on Computer Vision Workshops*, pp. 554–561, 2013.

[28] A. Khosla, N. Jayadevaprakash, B. Yao, and L. Fei-Fei, "Novel dataset for fine-grained image categorization : Stanford dogs," 2012.

[29] M.-E. Nilsback and A. Zisserman, "Automated flower classification over a large number of classes," *2008 Sixth Indian Conference on Computer Vision, Graphics & Image Processing*, pp. 722–729, 2008.