

# Bi-directional Feature Reconstruction Network for Fine-Grained Few-Shot Image Classification

Jijie Wu<sup>1</sup>, Dongliang Chang<sup>2</sup>, Aneeshan Sain<sup>3</sup>,  
Xiaoxu Li<sup>1\*</sup>, Zhanyu Ma<sup>2</sup>, Jie Cao<sup>1</sup>, Jun Guo<sup>2</sup>, and Yi-Zhe Song<sup>3</sup>

<sup>1</sup>School of Computer and Communications, Lanzhou University of Technology, Lanzhou, China

<sup>2</sup>School of Artificial Intelligence, Beijing University of Posts and Telecommunications, Beijing, China

<sup>3</sup>SketchX, CVSSP, University of Surrey, United Kingdom

{jijie, lixiaoxu, caoj}@lut.edu.cn, {changdongliang, mazhanyu, guojun}@bupt.edu.cn, {a.sain, y.song}@surrey.ac.uk

## Abstract

The main challenge for fine-grained few-shot image classification is to learn feature representations with higher inter-class and lower intra-class variations, with a mere few labelled samples. Conventional few-shot learning methods however cannot be naively adopted for this fine-grained setting – a quick pilot study reveals that they in fact push for the opposite (i.e., lower inter-class variations and higher intra-class variations). To alleviate this problem, prior works predominately use a support set to reconstruct the query image and then utilize metric learning to determine its category. Upon careful inspection, we further reveal that such unidirectional reconstruction methods only help to increase inter-class variations and are not effective in tackling intra-class variations. In this paper, we for the first time introduce a bi-reconstruction mechanism that can simultaneously accommodate for inter-class and intra-class variations. In addition to using the support set to reconstruct the query set for increasing inter-class variations, we further use the query set to reconstruct the support set for reducing intra-class variations. This design effectively helps the model to explore more subtle and discriminative features which is key for the fine-grained problem in hand. Furthermore, we also construct a self-reconstruction module to work alongside the bi-directional module to make the features even more discriminative. Experimental results on three widely used fine-grained image classification datasets consistently show considerable improvements compared with other methods. Codes are available at: <https://github.com/PRIS-CV/Bi-FRN>.

## Introduction

Few-shot fine-grained image classification (Huang et al. 2021; Li et al. 2021a) has emerged very recently as an important means to tackle the data scarcity problem widely facing fine-grained analysis (Wei et al. 2021). As per the conventional few-shot setting, it asks for effective model transfer given just a few support samples. The differences however lies in the unique characteristics brought by the fine-grained nature of the problem – the model needs to focus on learning subtle and discriminative features to discriminate not only on the category level (as per conventional few-shot), but importantly also on intra-class level differentiating

\*indicates corresponding author.

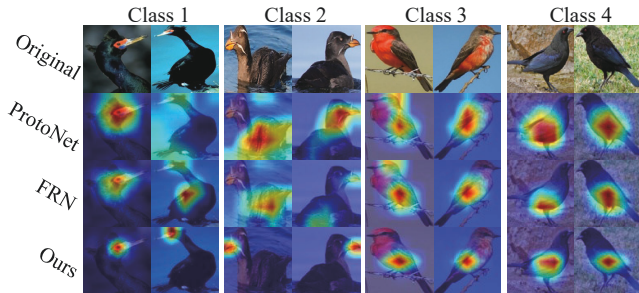


Figure 1: Visualization of the localized regions returned from Eigen-CAM (Muhammad and Yeasin 2020) based on a ProtoNet (Snell, Swersky, and Zemel 2017) model (trained on CUB-200-211 dataset) optimized by the traditional method, the FRN (Wertheimer, Tang, and Hariharan 2021), and the proposed method. The higher energy region denotes the more discriminative part in the image.

fine-grained visual differences amongst class instances.

It is therefore not surprising that traditional few-shot methods, when applied to the fine-grained problem, can no longer hold their promises. This is clearly seen in Figure 1, where we apply ProtoNet (Snell, Swersky, and Zemel 2017) on the “birds” dataset (Wah et al. 2011) – the model tends to encourage lower inter-class variations and higher intra-class variations, which is the very opposite of our goal of fine-grained image classification.

Attempts have been made on adapting traditional few-shot learning methods to the fine-grained scenario. Early attempts (Huang et al. 2021; Sun et al. 2020; Zhu, Liu, and Jiang 2020) have however focused on devising complex architectural designs which resulted in marginal gains over their vanilla counterparts. It was not until very recently that reconstruction-based methods have gained popularity (Wertheimer, Tang, and Hariharan 2021; Doersch, Gupta, and Zisserman 2020) and consequently achieved state-of-the-art performance. Through specifically engineering for support-query feature alignment, they naturally encourage fine-grained transfer. Effects of this can be observed in Figure 1, where feature regions learned by FRN (Wertheimer, Tang, and Hariharan 2021) tend to be more fine-grained when compared with ProtoNet (Snell, Swersky, and Zemel

2017).

However upon careful inspection, we importantly observe that although the model can focus on more subtle and discriminative regions, the semantic information they represented is nonetheless disparate between any two samples belonging to the same class (e.g., “Class 1” FRN focused on different parts of a bird). This seems to suggest that these reconstruction-based methods still suffer from having large intra-class variations, which is counter-productive for overall fine-grained learning (large inter-class variations and small intra-class variations). In other words, such methods only help to increase inter-class variations and do not reduce intra-class variations very well. The reason, we conjecture, is that the existing reconstruction-based methods (Wertheimer, Tang, and Hariharan 2021; Doersch, Gupta, and Zisserman 2020) (as shown in Figure 2(b)) works in a unidirectional fashion, i.e., reconstruction only happens one way from support features to query features. Consequently, it can only constrain the relationship between reconstructed query features and original query features to increase inter-class variations. This however fails to constrain the relationship between samples within each class in the support set, which is pivotal in decreasing intra-class variations.

As such, in this paper, we for the first time introduce a bi-directional reconstruction mechanism for few-shot fine-grained classification. Instead of using the support set to reconstruct the query set to increase inter-class variations only (as shown in Figure 2(b)), we additionally use the query set to reconstruct the support set to simultaneously reduce intra-class variations (as shown in Figure 2(c)). This modification might sound overly simple at first sight, it however importantly fulfills both desired learning outcomes for the fine-grained setting – support to query to encourage large inter-class variations, and query to support to encourage small intra-class variations (see Section for a detailed explanation).

Our bi-directional feature reconstruction framework mainly contains four modules: (i) a feature extraction module, (ii) a self-reconstruction module, (iii) a mutual bi-directional reconstruction module, and (iv) a distance metric module. (i) and (iv) are common to a reconstruction-based approach (Wertheimer, Tang, and Hariharan 2021). In addition to the proposed bi-directional module (iii), we also find use of a self-reconstruction module (ii) to benefit fine-grained feature learning, and works well with the former in terms of increasing inter-class variations and reducing intra-class variations. Ablative studies in Section show both to be effective and indispensable for few-shot fine-grained learning.

In summary, our contributions are three-fold: 1) We reveal the key problem in current reconstruction-based few-shot fine-grained classification lies with its inability in minimising intra-class variations. 2) We for the first time propose a bi-directional reconstruction network that simultaneously increase inter-class variations while reducing intra-class variations by way of a mutual support-query and query-support reconstruction. 3) Experimental results and ablative analyses on three fine-grained few-shot image datasets con-

sistently demonstrate the superiority of the proposed method and reveal insights on why the bi-directional approach is effective.

## Related Works

**Metric-Based Few-Shot Learning:** In few-shot learning methods, metric-based methods have received extensive attention owing to their simplicity and efficiency (Li et al. 2021b). A plethora of earlier works have adopted fixed metric or learnable module to learn a good metric embedding (Snell, Swersky, and Zemel 2017; Vinyals et al. 2016; Sung et al. 2018). These methods classify samples according to distance or similarity. Recently, GNN-based few-shot methods (Satorras and Estrach 2018; Kim et al. 2019; Yang et al. 2020) adopted graph neural networks (GNN) to model the similarity measurement – their advantage being that samples have a rich relational structure. In addition to these classic ones, some metric-based methods for fine-grained image classification have also emerged. While DN4 (Li et al. 2019) proposed a local descriptor-based image-to-class measure, using local features of samples to learn feature metric, BSNet (Li et al. 2021a) used a bi-similarity network to learn fewer but more discriminative regions using a combination of two different metrics. NDPNet (Zhang et al. 2021) designed a feature re-abstracting embedding network that projects the local features into the proposed similarity metric learning network to learn discriminative projection factors. Our method adopts a fixed euclidean distance to measure the error between construction features and origin features.

**Alignment-Based Few-Shot Learning:** Alignment-based fine-grained few-shot methods pay more attention to align the spatial positions of objects in images and then classify them based on a similarity measure. Unlike PARN (Wu et al. 2019) which is a position-aware relational network that aligns similar objects in spatial position and learns more flexible and robust metric capabilities, Semantic Alignment Metric Learning (SAML) (Hao et al. 2019) aligns the semantically relevant dominant objects in fine-grained images using a collect-and-select strategy. DeepEMD (Zhang et al. 2020) uses the Earth Mover’s Distance to generate the optimal matching flows between two local feature sets and computes the distance between two images based on the optimal matching flows and matching costs. In addition to these alignment-based methods above, reconstruction-based methods are also good for aligning spatial positions of objects from different fine-grained images. CTX (Doersch, Gupta, and Zisserman 2020) constructs a novel transformer-based neural network, which can find a coarse spatial relationship between the query and the labelled images, and then compute the distances between spatially-corresponding features to predict the label of samples. Alleviating the need for any new modules or large-scale learnable parameters FRN (Wertheimer, Tang, and Hariharan 2021) obtains the optimal reconstruction weights in the closed form solution to reconstruct query features from support features. Unlike these existing reconstruction based methods, we introduce bi-directional reconstruction method, that not only reconstructs query samples based on support samples for increas-

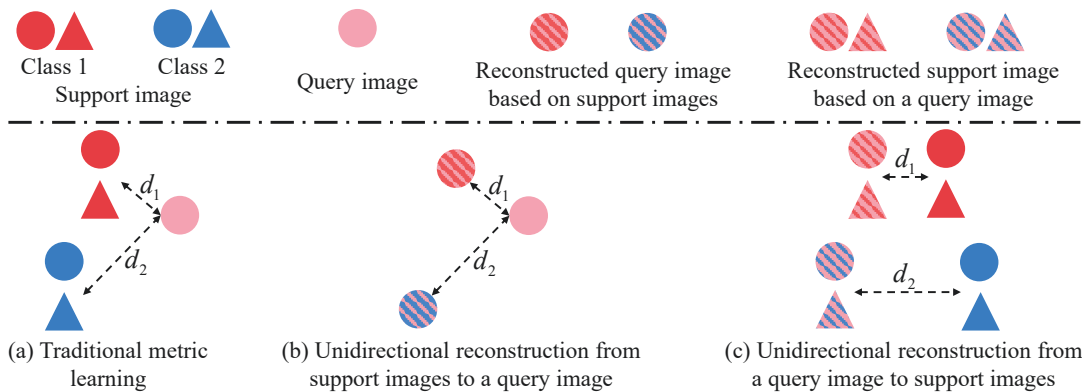


Figure 2: (a) is the traditional metric based method. (b) is the method proposed in (Wertheimer, Tang, and Hariharan 2021). (b) + (c) is the method proposed in this paper. (b) can help the model increase the inter-class variations, and (c) can help the model decrease the intra-class variations.

ing the inter-class variations, but also reconstructs support samples based on a query sample for reducing the intra-class variations.

**Attention Mechanism:** Vaswani et al. (Vaswani et al. 2017) first proposed the self-attention mechanism and then used it to build a new simple network architecture, namely Transformer. Besides achieving success in natural language processing (Vaswani et al. 2017; Kenton and Toutanova 2019; Brown et al. 2020), this architecture also works well in computer vision tasks (Dosovitskiy et al. 2021; Hassani et al. 2021; Chen, Fan, and Panda 2021). Recently, some few-shot learning works have gradually begun to adopt the self-attention mechanism. Han-Jia Ye et al. (Ye et al. 2020) proposed a Few-shot Embedding Adaptation with Transformer (FEAT) for few-shot learning. Here authors constructed set-to-set functions using a variety of approximators and found Transformer to be the most efficient option that can model interactions between images in a set and hence enable co-adaptation of each image. Unlike FEAT where the transformer structure is only used for support samples, CTX (Dorsch, Gupta, and Zisserman 2020) uses self-attention to calculate the spatial attention weights of query sample and support samples and learn a query-aligned class prototype. It then calculates the distance between the query sample and the aligned class prototype to classify the query sample. The proposed few-shot classification method also introduced the self-attention mechanism adopted in the transformer to learn optimal reconstruction weights in our self-reconstruction module and mutual reconstruction module.

## Methodology

In this section, we describe our proposed method, starting with a formulation of the proposed method followed by an overview and an in-depth description of each component.

### The Problem Formulation

Given a dataset  $D = \{(x_i, y_i), y_i \in Y\}$ , we divide it into three parts, that is,  $D_{base} = \{(x_i, y_i), y_i \in Y_{base}\}$ ,  $D_{val} = \{(x_i, y_i), y_i \in Y_{val}\}$  and  $D_{novel} = \{(x_i, y_i), y_i \in Y_{novel}\}$ ,

where  $x_i$  and  $y_i$  are the original feature vector and class label of the  $i^{\text{th}}$  image, respectively. The categories of these three parts are disjoint, i.e.,  $Y_{base} \cap Y_{val} \cap Y_{novel} = \emptyset$ , and  $Y_{base} \cup Y_{val} \cup Y_{novel} = Y$ . Few-shot classification aims to improve a  $C$ -way  $K$ -shot classification performance on  $D_{novel}$  by learning knowledge from  $D_{base}$  and  $D_{val}$ . For such a task, there are  $C$  classes sampled randomly from  $D_{novel}$ , and each class only has  $K$  randomly sampled labelled (support) samples  $S$ , and  $M$  randomly sampled unlabelled (query) samples  $Q$ .

Many few-shot methods adopt the meta-learning paradigm (Finn, Abbeel, and Levine 2017; Ye and Chao 2022). Specifically, the training process of these methods on  $D_{base}$  is the same as the prediction process on  $D_{novel}$ . Their purpose is to learn the meta-knowledge of learning a category concept based on a few labelled samples. In the meta-training phase, the few-shot classification model learns the categories of query samples based on few support samples in each task on  $D_{base}$ . The optimal model is selected by evaluating performance of the few-shot classification model on multiple tasks on  $D_{val}$ . In meta-test phase, the final performance of the optimal model are commonly evaluated by the average accuracy on all sampled tasks of  $D_{novel}$ .

### Overview

Learning subtle and discriminative features is crucial for fine-grained few-shot image classification. Considering that the existing reconstruction-based methods, which mostly reconstruct query features based on support features, fail to reduce intra-class variations adequately, we propose a bi-directional reconstruction network.

As shown in Figure 3, our model consists of four modules: the first is embedding module  $f_\theta$  for extracting deep convolutional image features. This can be a traditional convolutional neural network or a residual network. The second is a feature self-reconstruction module  $g_\phi$ , in which the convolutional features of each image are reconstructed by themselves based on a self-attention mechanism. This module can make the similar local features become more sim-

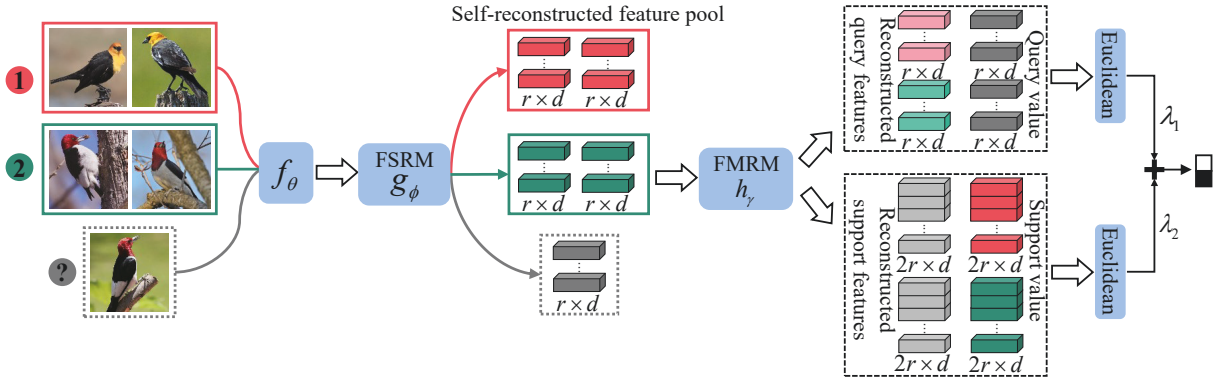


Figure 3: The proposed Bi-Directional feature reconstruction network. FSRM refers to Feature Self-reconstruction Module and FMRM refers to Feature Mutual Reconstruction Module.

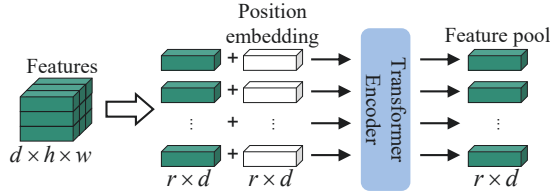


Figure 4: Feature self-reconstruction module.

ilar, while dissimilar ones even more dissimilar, and additionally benefits the following feature mutual reconstruction. The third is a feature mutual reconstruction module,  $h_\gamma$ , which reconstructs sample features in a bidirectional form. This module not only uses the support sample to reconstruct the query sample but also reconstructs the support sample from the query sample. Compared with the existing unidirectional reconstruction which only focuses on increasing inter-class variations of features, the bi-directional reconstruction adds another function – reducing the intra-class variations of features. Finally, the fourth is a Euclidean metric module, which is in charge of calculating the distance between query sample and reconstructed query sample, as well as support samples and reconstructed support samples. The weighted sum of the two distances are used for classifying query samples.

### Feature Self-Reconstruction Module (FSRM)

We constructed the feature self-reconstruction module (FSRM), as shown in Figure 4. For a  $C$ -way  $K$ -shot classification task, we input  $C \times (K + M)$  samples  $x_i$  into the embedding module  $f_\theta$  to extract features  $\hat{x}_i = f_\theta(x_i) \in \mathbb{R}^{d \times h \times w}$ , where  $d$  is the channel number,  $h$  and  $w$  represents the height and the width of features, respectively. Each image feature  $\hat{x}_i$  is the input of the FSRM,  $g_\phi$ , and the output of the FSRM is recorded as  $\hat{z}_i \in \mathbb{R}^{r \times d}$ .

First, we reshape the feature  $\hat{x}_i$  as  $r$  local features in spatial positions  $[\hat{x}_i^1, \hat{x}_i^2, \dots, \hat{x}_i^r]$ , where  $r = h \times w$ . The traditional vision transformers (Dosovitskiy et al. 2021) build on standard transformers (Vaswani et al. 2017), taking the sequence of image patches as input. We however compute the

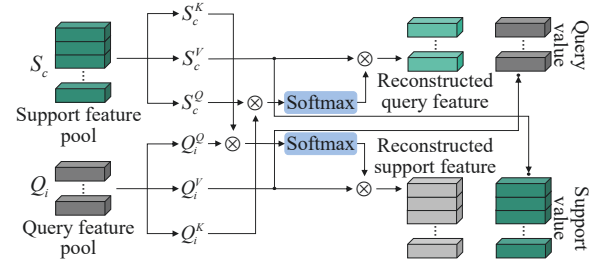


Figure 5: Feature mutual reconstruction module.

summation of the sequence of local features  $\hat{x}_i^j$  and the corresponding spatial position embedding  $E_{pos} \in \mathbb{R}^{r \times d}$  as the input of the transformer, i.e.,  $z_i = [\hat{x}_i^1, \hat{x}_i^2, \dots, \hat{x}_i^r] + E_{pos}$ , where  $E_{pos}$  adopts sinusoidal position encoding (Vaswani et al. 2017).

The output of FSRM module is computed based on the standard self-attention operation in Transformer Encoder, and the computing operation is shown as follows:

$$Attention(Q, K, V) = Softmax\left(\frac{QK^T}{\sqrt{d_k}}\right)V. \quad (1)$$

Therefore, we can obtain  $\hat{z}_i$  as,

$$\hat{z}_i = Attention(z_i W_\phi^Q, z_i W_\phi^K, z_i W_\phi^V), \hat{z}_i \in \mathbb{R}^{r \times d}, \quad (2)$$

where  $W_\phi^Q$ ,  $W_\phi^K$ , and  $W_\phi^V$  are a set of learnable weight parameters with  $d \times d$  size. Next, the  $\hat{z}_i$  are calculated continually by a Layer Normalization (LN) and a Multi-Layer Perceptron (MLP).

$$\hat{z}_i = MLP(LN(z_i + \hat{z}_i)). \quad (3)$$

### Feature Mutual Reconstruction Module (FMRM)

We propose Feature Mutual Reconstruction Module (FMRM), as shown in Figure 5, which contains two operations: reconstructing support features in one class given a query feature and reconstructing query feature given support features in one class.

For  $C$ -way  $K$ -shot classification task, after going through FSRM, we can obtain reconstructed support features of the

$c^{\text{th}}$  class, i.e.  $S_c = [\hat{z}_k^c] \in \mathbb{R}^{kr \times d}$ , where  $k \in [1, \dots, K]$  and  $c \in [1, \dots, C]$ , and reconstructed query feature  $\hat{Q}_i = \hat{z}_i \in \mathbb{R}^{r \times d}$ , where  $i \in [1, \dots, C \times M]$ .  $S_c$  multiplies by weights  $W_\gamma^Q, W_\gamma^K$  and  $W_\gamma^V$ , respectively, obtaining  $S_c^Q, S_c^K$  and  $S_c^V$ , where  $W_\gamma^Q, W_\gamma^K, W_\gamma^V \in \mathbb{R}^{d \times d}$ . Similarly,  $Q_i$  multiplies by weights  $W_\gamma^Q, W_\gamma^K$  and  $W_\gamma^V$ , respectively, obtaining  $Q_i^Q, Q_i^K$  and  $Q_i^V$ .

We calculate the reconstructed  $i^{\text{th}}$  query feature  $\hat{Q}_{(c,i)}$  from support features  $S_c^V$  in the  $c^{\text{th}}$  class and the reconstructed support features  $\hat{S}_{(i,c)}$  in the  $c^{\text{th}}$  class from  $i^{\text{th}}$  query using the two equations below.

$$\hat{Q}_{(c,i)} = \text{Attention}(Q_i^Q, S_c^K, S_c^V), \hat{Q}_{(c,i)} \in \mathbb{R}^{r \times d}, \quad (4)$$

$$\hat{S}_{(i,c)} = \text{Attention}(S_c^Q, Q_i^K, Q_i^V), \hat{S}_{(i,c)} \in \mathbb{R}^{kr \times d}. \quad (5)$$

where  $\text{Attention}(\cdot, \cdot, \cdot)$  is shown in Equation 1.

To our best knowledge, the existing reconstruction methods only use unidirectional reconstruction – using support features to reconstruct the query feature. Building on existing methods, we add the reverse reconstruction shown in Equation 5 – using query feature to reconstruct support features.

## Learning Objectives

After FMRM, we use the Euclidean metric to compute the distance from this query sample  $Q_i$  to the support samples in the  $c^{\text{th}}$  class (Figure 2(b)) as:

$$d_{Q_i \rightarrow S_c} = \|Q_i^V - \hat{Q}_{(c,i)}\|^2, \quad (6)$$

and compute the distance from the support samples in the  $c^{\text{th}}$  class to the query sample  $Q_i$  (Figure 2(c)) as:

$$d_{S_c \rightarrow Q_i} = \|S_c^V - \hat{S}_{(i,c)}\|^2. \quad (7)$$

The total distance can be obtained by weighted summation of two distances  $d_{Q_i \rightarrow S_c}$  and  $d_{S_c \rightarrow Q_i}$ , as,

$$d_i^c = \tau(\lambda_1 d_{Q_i \rightarrow S_c} + \lambda_2 d_{S_c \rightarrow Q_i}), \quad (8)$$

where  $\lambda_1$  and  $\lambda_2$  are learnable weight parameters, and both of them are initialized as 0.5.  $\tau$  is a learnable temperature factor, following (Wertheimer, Tang, and Hariharan 2021; Ye et al. 2020; Chen et al. 2020; Gidaris and Komodakis 2018). We normalize  $d_i^c$  to obtain  $\hat{d}_i^c$  as:

$$\hat{d}_i^c = \frac{e^{-d_i^c}}{\sum_{c=1}^C e^{-d_i^c}}. \quad (9)$$

Based on  $\hat{d}_i^c$ , the total loss in one  $C$ -way  $K$ -shot task can be calculated as:

$$\mathcal{L} = -\frac{1}{M \times C} \sum_{i=1}^{M \times C} \sum_{c=1}^C \mathbf{1}(y_i == c) \log(\hat{d}_i^c), \quad (10)$$

where  $\mathbf{1}(y_i == c)$  equals 1 when  $y_i$  and  $c$  are equal, otherwise 0.

During the training process, for a  $C$ -way  $K$ -shot task on  $D_{base}$ , we minimize  $\mathcal{L}$  to update the proposed network, and repeat this process on all randomly generated tasks.

## Experimental Results and Analysis

### Datasets and Implementation Details

**Datasets:** To evaluate the performance of the proposed method, we selected three benchmark fine-grained datasets, CUB-200-2011 (CUB) (Wah et al. 2011) is a classic fine-grained image classification dataset. It contains 11,788 images from 200 bird species. Following (Zhang et al. 2020; Ye et al. 2020), we crop each image to a human annotated bounding box. Stanford-Dogs (Dogs) (Khosla et al. 2011) is a challenging fine-grained image categorization dataset. The dataset includes 20,580 annotated images of 120 breeds of dogs from around the world. Stanford-Cars (Cars) (Krause et al. 2013) is also a commonly used benchmark dataset for fine-grained image classification. The dataset contains 16,185 images of 196 car-types. For each dataset, we divided it into  $D_{train}$ ,  $D_{val}$  and  $D_{test}$ . The ratio of  $D_{train}$ ,  $D_{val}$  and  $D_{test}$  is same as the literature (Zhu, Liu, and Jiang 2020), and all images are resized to  $84 \times 84$ .

**Implementation Details:** We conducted experiments on two widely used backbone architectures: Conv-4 and ResNet-12. The architectures of Conv-4 and ResNet-12 are the same as that of (Wertheimer, Tang, and Hariharan 2021; Ye et al. 2020). We implemented all our experiments on NVIDIA 3090Ti GPUs via Pytorch (Paszke et al. 2019). In our experiments, we train all Conv-4 and ResNet-12 models for 1,200 epochs using SGD with Nesterov momentum of 0.9. The initial learning rate is set to 0.1 and weight decay to  $5e-4$ . Learning rate is decreased by a scaling factor of 10 after every 400 epochs. For Conv-4 models, we train the proposed models using 30-way 5-shot episodes, and test for 1-shot and 5-shot episodes. We use 15 query images per class in both settings. Furthermore, for ResNet-12 models we train our proposed model using 15-way 5-shot episodes, in order to save memory. We employ standard data augmentation, including center crop, random horizontal flip and colour jitter for better training stability. Thereafter, we select the best-performing model based on the validation set, and validate every 20 epochs. For all experiments, we report the mean accuracy of 10,000 randomly generated tasks on  $D_{test}$  with 95% confidence intervals on the standard 5-way, 1-shot and 5-shot settings.

### Comparison with State-of-the-Arts

To validate the efficiency of our method for fine-grained few-shot image classification, we conducted experiments on the three fine-grained image classification datasets discussed earlier. The results of Relation (Sung et al. 2018), DN4 (Li et al. 2019) and BSNet (Li et al. 2021a) are from literature BSNet (Li et al. 2021a), and the results of SAML (Hao et al. 2019) and DeepEMD (Zhang et al. 2020) are from DLG (Cao et al. 2022), and the results of LRPABN (Huang et al. 2021) are from MattML (Zhu, Liu, and Jiang 2020). The results of methods marked with †, such as ProtoNet (Snell, Swersky, and Zemel 2017), PARN (Wu et al. 2019), CTX (Doersch, Gupta, and Zisserman 2020), FRN (Wertheimer, Tang, and Hariharan 2021), FRN+TDM (Lee, Moon, and Heo 2022) and DeepEMD (Zhang et al. 2020) are obtained via the code officially

| <i>Method</i>            | <i>CUB</i>        |                   | <i>Dogs</i>       |                   | <i>Cars</i>       |                   |
|--------------------------|-------------------|-------------------|-------------------|-------------------|-------------------|-------------------|
|                          | 1-shot            | 5-shot            | 1-shot            | 5-shot            | 1-shot            | 5-shot            |
| ProtoNet† (NeurIPS 2017) | 64.82±0.23        | 85.74±0.14        | 46.66±0.21        | 70.77±0.16        | 50.88±0.23        | 74.89±0.18        |
| Relation (CVPR 2018)     | 63.94±0.92        | 77.87±0.64        | 47.35±0.88        | 66.20±0.74        | 46.04±0.91        | 68.52±0.78        |
| DN4 (CVPR 2019)          | 57.45±0.89        | 84.41±0.58        | 39.08±0.76        | 69.81±0.69        | 34.12±0.68        | 87.47±0.47        |
| PARN† (ICCV 2019)        | 74.43±0.95        | 83.11±0.67        | 55.86±0.97        | 68.06±0.72        | 66.01±0.94        | 73.74±0.70        |
| SAML (ICCV 2019)         | 65.35±0.65        | 78.47±0.41        | 45.46±0.36        | 59.65±0.51        | 61.07±0.47        | 88.73±0.49        |
| DeepEMD (CVPR 2020)      | 64.08±0.50        | 80.55±0.71        | 46.73±0.49        | 65.74±0.63        | 61.63±0.27        | 72.95±0.38        |
| LRPABN (TMM 2021)        | 63.63±0.77        | 76.06±0.58        | 45.72±0.75        | 60.94±0.66        | 60.28±0.76        | 73.29±0.58        |
| BSNet(D&C) (TIP 2021)    | 62.84±0.95        | 85.39±0.56        | 43.42±0.86        | 71.90±0.68        | 40.89±0.77        | 86.88±0.50        |
| CTX† (NeurIPS 2020)      | 72.61±0.21        | 86.23±0.14        | 57.86±0.21        | 73.59±0.16        | 66.35±0.21        | 82.25±0.14        |
| FRN† (CVPR 2021)         | 74.90±0.21        | 89.39±0.12        | 60.41±0.21        | 79.26±0.15        | 67.48±0.22        | 87.97±0.11        |
| FRN+TDM† (CVPR 2022)     | 72.01±0.22        | 89.05±0.12        | 51.57±0.23        | 75.25±0.16        | 65.67±0.22        | 86.44±0.12        |
| Ours                     | <b>79.08±0.20</b> | <b>92.22±0.10</b> | <b>64.74±0.22</b> | <b>81.29±0.14</b> | <b>75.74±0.20</b> | <b>91.58±0.09</b> |
| ProtoNet† (NeurIPS 2017) | 81.02±0.20        | 91.93±0.11        | 73.81±0.21        | 87.39±0.12        | 85.46±0.19        | 95.08±0.08        |
| CTX† (NeurIPS 2020)      | 80.39±0.20        | 91.01±0.11        | 73.22±0.22        | 85.90±0.13        | 85.03±0.19        | 92.63±0.11        |
| DeepEMD† (CVPR 2020)     | 75.59±0.30        | 88.23±0.18        | 70.38±0.30        | 85.24±0.18        | 80.62±0.26        | 92.63±0.13        |
| FRN† (CVPR 2021)         | 84.30±0.18        | 93.34±0.10        | 76.76±0.21        | 88.74±0.12        | 88.01±0.17        | 95.75±0.07        |
| FRN+TDM† (CVPR 2022)     | 85.15±0.18        | 93.99±0.09        | <b>78.02±0.20</b> | <b>89.85±0.11</b> | 88.92±0.16        | 96.88±0.06        |
| Ours                     | <b>85.44±0.18</b> | <b>94.73±0.09</b> | 76.89±0.21        | 88.27±0.12        | <b>90.44±0.15</b> | <b>97.49±0.05</b> |

Table 1: 5-way few-shot classification performance on the *CUB*, *Dogs* and *Cars* datasets. The top block uses Conv-4 backbone and the bottom block uses ResNet-12 backbone. Methods labeled by † denote our implementations.

| <i>Backbone</i> | <i>Method</i>       | <i>CUB</i>        |                   | <i>Dogs</i>       |                   | <i>Cars</i>       |                   |
|-----------------|---------------------|-------------------|-------------------|-------------------|-------------------|-------------------|-------------------|
|                 |                     | 1-shot            | 5-shot            | 1-shot            | 5-shot            | 1-shot            | 5-shot            |
| Conv-4          | Baseline (ProtoNet) | 64.82±0.23        | 85.74±0.14        | 46.66±0.21        | 70.77±0.16        | 50.88±0.23        | 74.89±0.18        |
|                 | (FSRM)              | 75.37±0.21        | 88.61±0.12        | <b>65.10±0.22</b> | 79.94±0.15        | 71.61±0.22        | 84.70±0.14        |
|                 | (FMRM)              | 74.92±0.21        | 89.97±0.11        | 61.28±0.21        | 80.07±0.14        | 70.22±0.21        | 88.45±0.11        |
|                 | (FSRM+FMRM)         | <b>79.08±0.20</b> | <b>92.22±0.10</b> | 64.74±0.22        | <b>81.29±0.14</b> | <b>75.74±0.20</b> | <b>91.58±0.09</b> |
| ResNet-12       | Baseline (ProtoNet) | 81.02±0.20        | 91.93±0.11        | 73.81±0.21        | 87.39±0.12        | 85.46±0.19        | 95.08±0.08        |
|                 | (FSRM)              | 82.53±0.19        | 92.43±0.10        | 75.64±0.21        | 87.44±0.12        | 85.95±0.18        | 94.44±0.08        |
|                 | (FMRM)              | 84.33±0.18        | 94.25±0.09        | 76.29±0.21        | <b>89.06±0.11</b> | 89.62±0.15        | 97.45±0.05        |
|                 | (FSRM+FMRM)         | <b>85.44±0.18</b> | <b>94.73±0.09</b> | <b>76.89±0.21</b> | 88.27±0.12        | <b>90.44±0.15</b> | <b>97.49±0.05</b> |

Table 2: Ablation studies using only FSRM module or FMRM module.

| <i>Backbone</i> | <i>Method</i>       | <i>CUB</i>        |                   | <i>Dogs</i>       |                   | <i>Cars</i>       |                   |
|-----------------|---------------------|-------------------|-------------------|-------------------|-------------------|-------------------|-------------------|
|                 |                     | 1-shot            | 5-shot            | 1-shot            | 5-shot            | 1-shot            | 5-shot            |
| Conv-4          | Baseline (ProtoNet) | 64.82±0.23        | 85.74±0.14        | 46.66±0.21        | 70.77±0.16        | 50.88±0.23        | 74.89±0.18        |
|                 | Ours (Q→S)          | <b>79.88±0.20</b> | 91.76±0.11        | <b>65.26±0.22</b> | 80.81±0.14        | 75.61±0.20        | 90.49±0.10        |
|                 | Ours (S→Q)          | 76.54±0.21        | 88.03±0.14        | 64.39±0.22        | 78.36±0.15        | 72.71±0.22        | 85.11±0.14        |
|                 | Ours (Mutual)       | 79.08±0.20        | <b>92.22±0.10</b> | 64.74±0.22        | <b>81.29±0.14</b> | <b>75.74±0.20</b> | <b>91.58±0.09</b> |
| ResNet-12       | Baseline (ProtoNet) | 81.02±0.20        | 91.93±0.11        | 73.81±0.21        | 87.39±0.12        | 85.46±0.19        | 95.08±0.08        |
|                 | Ours (Q→S)          | 83.72±0.19        | 93.31±0.09        | 76.50±0.21        | 87.95±0.12        | 87.37±0.17        | 95.10±0.08        |
|                 | Ours (S→Q)          | 81.72±0.19        | 90.83±0.11        | 75.62±0.22        | 86.47±0.13        | 85.90±0.18        | 93.17±0.10        |
|                 | Ours (Mutual)       | <b>85.44±0.18</b> | <b>94.73±0.09</b> | <b>76.89±0.21</b> | <b>88.27±0.12</b> | <b>90.44±0.15</b> | <b>97.49±0.05</b> |

Table 3: Ablation on reconstruction designs of FMRM.

provided by the author, which is replaced by the dataset used in this paper.

We use Conv-4 and ResNet-12 as the backbone of all compared methods and test 5-way 1-shot and 5-way 5-shot

classification performance. As seen from Table 1, the proposed method achieves highest accuracy on all three datasets when the Conv-4 is adopted. Apart from the result on the Dogs (Khosla et al. 2011) dataset where performance falls

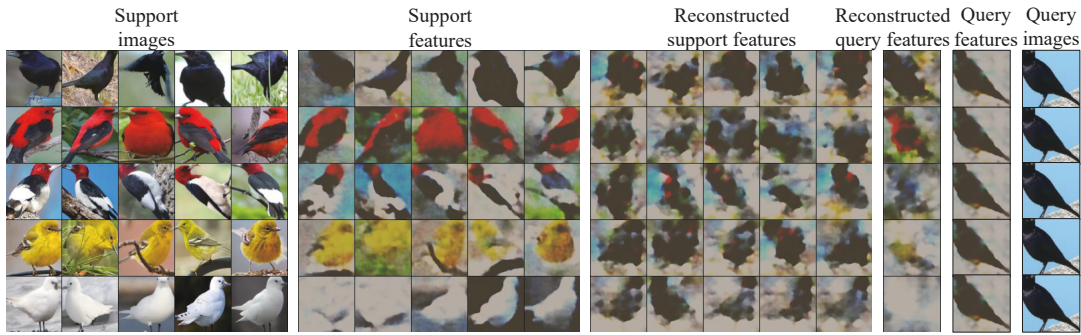


Figure 6: Recovered images of different features by our method for the CUB dataset.

slightly behind the FRN+TDM method when the ResNet-12 is adopted, our method achieves highest accuracy.

In a nutshell, compared with other newly proposed methods, our method achieves stable and excellent performance on the three fine-grained image datasets on both 5-way 1-shot and 5-way 5-shot classification tasks, majorly owing to our network structure. The two modules FSRM and FMRM, conditions the model to learn the subtle and discriminative features for fine-grained classification.

### Ablation Study

To further justify the design choices of our method and model components towards efficiency and accuracy, we perform a few ablation studies on three datasets using both Conv-4 and ResNet-12 as backbone networks.

**The Effectiveness of FSRM and FMRM:** We compare our method’s efficacy by removing components in a strip-down fashion. In other words, we conduct experiments and report in Table 2, where we remove FSRM module (*FMRM*), FMRM module (*FSRM*), and then both which becomes equivalent to ProtoNet (Snell, Swersky, and Zemel 2017) (*Baseline (ProtoNet)*). Evidently, performance improves further in our method where both FSRM and FMRM are used together (*FSRM+FMRM*). Therefore, FSRM and FMRM modules are indispensable and complementary.

**The Effectiveness of Mutual Reconstruction in FMRM:** For our method, we remove reconstruction of support samples based on query samples in FMRM, i.g.,  $\lambda_2$  becoming 0 in Equation 8, which is marked as Ours ( $Q \rightarrow S$ ). And we remove reconstruction of query samples based on support samples, i.g.,  $\lambda_1$  becoming 0 in Equation 8, which is marked as Ours ( $S \rightarrow Q$ ). As per Table 3, both unidirectional reconstruction methods score lower than the proposed feature mutual reconstruction method (FMRM) in most cases. Therefore, it can be concluded that the design of the Feature Mutual Reconstruction Module (FMRM) is reasonable and effective.

### Visualization Analysis

To demonstrate the efficiency of our proposed network better, we recovered the original and reconstructed features. We trained an inverse ResNet as a decoder, the input of which is the feature value in the mutual reconstruction module, and the output of which is a  $3 \times 84 \times 84$  recovered image. In

the training process, we use an Adam optimizer with an initial learning rate of 0.01, set batch size as 200 and train for 1,000 epochs, with an L1 loss to measure the prediction error.

As per Figure 6, the left-most block shows the support images of 5 classes, each of which occupies one row and contains 5 images. The right-most column is a query image, which we copied 5 times simply for convenient comparison and aesthetics. The second block from left is the recovered images of support features  $S_c^V$ , whereas second column from the right is the recovered images of the  $i^{\text{th}}$  query features  $Q_i^V$ . The third block from left is the recovered images of reconstructed support feature  $\hat{S}_{(i,c)}$  based on  $i^{\text{th}}$  query feature, while that from the right is the recovered images of reconstructed query features  $\hat{Q}_{(c,i)}$  based on each  $c$  class support feature.

From the first and second blocks (first being left-most), it can be seen that the decoder we trained is capable of recovering images from features. For the third and fourth ones, it can be seen that if a query sample is used to reconstruct the support samples which has same class as the query sample, the reconstructed features will be similar to the original support features. However if we use a query sample to reconstruct the support samples which has a class different from the query sample, the reconstructed features will be quite different from the original support features. Likewise, when we use support samples to reconstruct the query samples, similar results are obtained. This indicates that the proposed network can effectively alleviate inaccurate similarity measure of the unaligned fine-grained images.

### Conclusion

In this paper, we proposed a bi-directional feature reconstruction network for few-shot fine-grained image classification. Our major contribution is a mutual reconstruction module that works in both directions, i.e., support to test and test to support. Compared to the existing reconstruction-based methods, the proposed method can achieve larger inter-class variations and lower the intra-class variations which is crucial to fine-grained learning. Extensive experiments show that the proposed network can perform well on three fine-grained image datasets consistently, competing strongly and at times surpassing contemporary state-of-the-arts.

## Acknowledgments

This work was supported in part by Beijing Natural Science Foundation under Grant Z200002, in part by National Natural Science Foundation of China (NSFC) No. 62111530146, 62176110, 61906080, 61922015, U19B2036, 62225601, in part by the Young Doctoral Fund of Education Department of Gansu Province under Grant 2021QB-038, in part by Hong-liu Distinguished Young Talents Foundation of Lanzhou University of Technology, in part by a scholarship from China Scholarship Council (CSC) under Grant CSC No.202006470036, and in part by BUPT Excellent Ph.D. Students Foundation No.CX2020105.

## References

- Brown, T. B.; Mann, B.; Ryder, N.; Subbiah, M.; Kaplan, J.; Dhariwal, P.; Neelakantan, A.; Shyam, P.; Sastry, G.; Askell, A.; et al. 2020. Language Models are Few-Shot Learners. In *NeurIPS*.
- Cao, S.; Wang, W.; Zhang, J.; Zheng, M.; and Li, Q. 2022. A Few-Shot Fine-Grained Image Classification Method Leveraging Global and Local Structures. *International Journal of Machine Learning and Cybernetics*.
- Chen, C.-F.; Fan, Q.; and Panda, R. 2021. CrossViT: Cross-Attention Multi-Scale Vision Transformer for Image Classification. In *ICCV*.
- Chen, Y.; Wang, X.; Liu, Z.; Xu, H.; Darrell, T.; et al. 2020. A New Meta-Baseline for Few-Shot Learning. *arXiv preprint arXiv:2003.04390*.
- Doersch, C.; Gupta, A.; and Zisserman, A. 2020. CrossTransformers: Spatially-Aware Few-Shot Transfer. In *NerulPS*.
- Dosovitskiy, A.; Beyer, L.; Kolesnikov, A.; Weissenborn, D.; Zhai, X.; Unterthiner, T.; Dehghani, M.; Minderer, M.; Heigold, G.; Gelly, S.; et al. 2021. An Image Is Worth 16x16 Words: Transformers for Image Recognition at Scale. In *ICLR*.
- Finn, C.; Abbeel, P.; and Levine, S. 2017. Model-Agnostic Meta-Learning for Fast Adaptation of Deep Networks. In *ICML*.
- Gidaris, S.; and Komodakis, N. 2018. Dynamic Few-Shot Visual Learning Without Forgetting. In *CVPR*.
- Hao, F.; He, F.; Cheng, J.; Wang, L.; Cao, J.; and Tao, D. 2019. Collect and Select: Semantic Alignment Metric Learning for Few-Shot Learning. In *ICCV*.
- Hassani, A.; Walton, S.; Shah, N.; Abuduweili, A.; Li, J.; and Shi, H. 2021. Escaping the Big Data Paradigm with Compact Transformers. *arXiv preprint arXiv:2104.05704*.
- Huang, H.; Zhang, J.; Zhang, J.; Xu, J.; and Wu, Q. 2021. Low-Rank Pairwise Alignment Bilinear Network for Few-Shot Fine-Grained Image Classification. *IEEE Transactions on Multimedia*.
- Kenton, J. D. M.-W. C.; and Toutanova, L. K. 2019. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In *NAACL-HLT*.
- Khosla, A.; Jayadevaprakash, N.; Yao, B.; and Li, F.-F. 2011. Novel Dataset for Fine-Grained Image Categorization: Stanford Dogs. In *CVPR workshops*.
- Kim, J.; Kim, T.; Kim, S.; and Yoo, C. D. 2019. Edge-Labeling Graph Neural Network for Few-Shot Learning. In *CVPR*.
- Krause, J.; Stark, M.; Deng, J.; and Fei-Fei, L. 2013. 3D Object Representations for Fine-Grained Categorization. In *ICCV Workshops*.
- Lee, S.; Moon, W.; and Heo, J.-P. 2022. Task Discrepancy Maximization for Fine-Grained Few-Shot Classification. In *CVPR*.
- Li, W.; Wang, L.; Xu, J.; Huo, J.; Gao, Y.; and Luo, J. 2019. Revisiting Local Descriptor Based Image-To-Class Measure for Few-Shot Learning. In *CVPR*.
- Li, X.; Wu, J.; Sun, Z.; Ma, Z.; Cao, J.; and Xue, J.-H. 2021a. BSNet: Bi-similarity Network for Few-Shot Fine-Grained Image Classification. *IEEE Transactions on Image Processing*.
- Li, X.; Yang, X.; Ma, Z.; and Xue, J.-H. 2021b. Deep Metric Learning for Few-Shot Image Classification: A Review of Recent Developments. *Pattern Recognition*.
- Muhammad, M. B.; and Yeasin, M. 2020. Eigen-Cam: Class Activation Map Using Principal Components. In *IJCNN*.
- Paszke, A.; Gross, S.; Massa, F.; Lerer, A.; Bradbury, J.; Chanan, G.; Killeen, T.; Lin, Z.; Gimelshein, N.; Antiga, L.; et al. 2019. PyTorch: An Imperative Style, High-Performance Deep Learning Library. In *NeurIPS*.
- Satorras, V. G.; and Estrach, J. B. 2018. Few-Shot Learning with Graph Neural Networks. In *ICLR*.
- Snell, J.; Swersky, K.; and Zemel, R. 2017. Prototypical Networks for Few-Shot Learning. In *NerulPS*.
- Sun, X.; Xv, H.; Dong, J.; Zhou, H.; Chen, C.; and Li, Q. 2020. Few-Shot Learning for Domain-Specific Fine-Grained Image Classification. *IEEE Transactions on Industrial Electronics*.
- Sung, F.; Yang, Y.; Zhang, L.; Xiang, T.; Torr, P. H.; and Hospedales, T. M. 2018. Learning to Compare: Relation Network for Few-Shot Learning. In *CVPR*.
- Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A. N.; Kaiser, L. u.; and Polosukhin, I. 2017. Attention is All you Need. In *NerulPS*.
- Vinyals, O.; Blundell, C.; Lillicrap, T.; Kavukcuoglu, K.; and Wierstra, D. 2016. Matching Networks for One Shot Learning. In *NerulPS*.
- Wah, C.; Branson, S.; Welinder, P.; Perona, P.; and Belongie, S. J. 2011. The Caltech-UCSD Birds-200-2011 Dataset. Technical report, California Institute of Technology.
- Wei, X.-S.; Song, Y.-Z.; Mac Aodha, O.; Wu, J.; Peng, Y.; Tang, J.; Yang, J.; and Belongie, S. 2021. Fine-Grained Image Analysis with Deep Learning: A Survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*.
- Wertheimer, D.; Tang, L.; and Hariharan, B. 2021. Few-Shot Classification with Feature Map Reconstruction Networks. In *CVPR*.

- Wu, Z.; Li, Y.; Guo, L.; and Jia, K. 2019. PARN: Position-Aware Relation Networks for Few-Shot Learning. In *ICCV*.
- Yang, L.; Li, L.; Zhang, Z.; Zhou, X.; Zhou, E.; and Liu, Y. 2020. DPGN: Distribution Propagation Graph Network for Few-Shot Learning. In *CVPR*.
- Ye, H.-J.; and Chao, W.-L. 2022. How to Train Your MAML to Excel in Few-Shot Classification. In *ICLR*.
- Ye, H.-J.; Hu, H.; Zhan, D.-C.; and Sha, F. 2020. Few-Shot Learning via Embedding Adaptation with Set-to-Set Functions. In *CVPR*.
- Zhang, C.; Cai, Y.; Lin, G.; and Shen, C. 2020. DeepEMD: Few-Shot Image Classification with Differentiable Earth Mover’s Distance and Structured Classifiers. In *CVPR*.
- Zhang, W.; Liu, X.; Xue, Z.; Gao, Y.; and Sun, C. 2021. NDPNet: A Novel Non-linear Data Projection Network for Few-Shot Fine-Grained Image Classification. *arXiv preprint arXiv:2106.06988*.
- Zhu, Y.; Liu, C.; and Jiang, S. 2020. Multi-Attention Meta Learning for Few-Shot Fine-Grained Image Recognition. In *IJCAI*.