Low-Rank Pairwise Alignment Bilinear Network For Few-Shot Fine-Grained Image Classification

Huaxi Huang, *Student Member, IEEE,* Junjie Zhang, Jian Zhang, *Senior Member, IEEE,*
Jingsong Xu, Qiang Wu, *Senior Member, IEEE*

Abstract—Deep neural networks have demonstrated advanced abilities on various visual classification tasks, which heavily rely on the large-scale training samples with annotated ground-truth. However, it is unrealistic always to require such annotation in real-world applications. Recently, Few-Shot learning (FS), as an attempt to address the shortage of training samples, has made significant progress in generic classification tasks. Nonetheless, it is still challenging for current FS models to distinguish the subtle differences between fine-grained categories given limited training data. To filling the classification gap, in this paper, we address the Few-Shot Fine-Grained (FSFG) classification problem, which focuses on tackling the fine-grained classification under the challenging few-shot learning setting. A novel low-rank pairwise bilinear pooling operation is proposed to capture the nuanced differences between the support and query images for learning an effective distance metric. Moreover, a feature alignment layer is designed to match the support image features with query ones before the comparison. We name the proposed model Low-Rank Pairwise Alignment Bilinear Network (LRPABN), which is trained in an end-to-end fashion. Comprehensive experimental results on four widely used fine-grained classification data sets demonstrate that our LRPABN model achieves the superior performances compared to state-of-the-art methods.

Index Terms—Few-Shot, Fine-Grained, Low-Rank, Pairwise, Bilinear Pooling, Feature Alignment.

I. INTRODUCTION

FINE-GRAINED image classification aims to distinguish different sub-categories belong to the same entry-level category such as birds [2], [3], dogs [4], and cars [5]. This problem is particularly challenging due to the low inter-category variance yet high intra-category discordance caused by various object postures, illumination conditions and distances from the cameras, *etc.* In general, the majority of fine-grained classification approaches need to be fed with a large amount of training data before obtaining a trustworthy classifier [6]–[11]. However, labeling the fine-grained data requires strong domain knowledge, *e.g.*, only ornithologists can accurately identify different bird species, which is significantly expensive compared to the generic object classification task. Moreover, in some fine-grained data sets such as the Wild-fish [12] and iNaturalist [13], the data distributions are usually imbalanced and follow the long-tail distribution, and in some of the categories, the well-labeled training samples are limited,

Huaxi Huang and Junjie Zhang are co-first authors. Corresponding author: Jian Zhang, email: Jian.Zhang@uts.edu.au.

Huaxi Huang, Jian Zhang, Qiang Wu and Jingsong Xu are with the Faculty of Engineering and Information Technology, University of Technology Sydney, Sydney NSW 2007, Australia. Junjie Zhang is with the School of Computer Science, The University of Adelaide, Adelaide SA 5005, Australia.

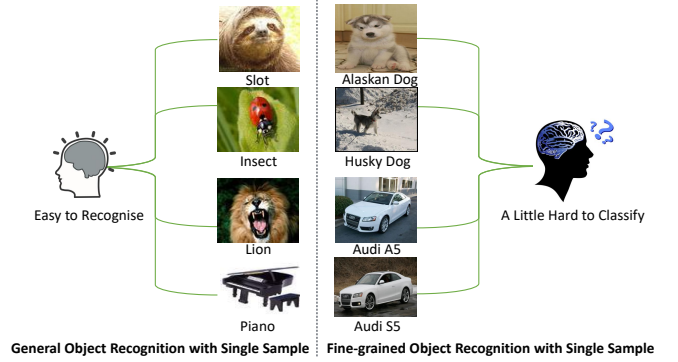The preliminary version of this work is accepted at IEEE ICME 2019 [1].

Fig. 1. An example of general one-shot learning (Left) and fine-grained one-shot learning (Right). For general one-shot learning, it is easy to learn the concepts of objects with only a single image. However, it is more difficult to distinguish the sub-classes of specific categories with one sample.

e.g., it is hard to collect large-scale samples of endangered species. How to tackle the fine-grained image classification with limited training data remains an open problem.

Human beings can learn novel generic concepts with only one or a few samples easily. To simulate this intelligent ability, machine few-shot learning is initially identified by Li *et al.* [14]. They propose to utilize probabilistic models to represent object categories and update them with a few training examples. Most recently, inspired by the advanced representation learning ability of deep neural networks, deep machine few-shot learning [15]–[20] revives and achieves significant improvements against previous methods. However, considering the cognitive process of human beings, preschool students can easily distinguish the difference between generic concepts like the 'Cat' and 'Dog' after seeing a few exemplary images of these animals, but they may be confused about fine-grained dog categories such as the 'Husky' and 'Alaskan' with limited samples. The undeveloped classification ability of children in processing information compared to adults [21], [22] indicates that generic few-shot methods cannot cope with the few-shot fine-grained classification task admirably. To this end, in this paper, we focus on dealing with the Few-Shot Fine-Gained (FSFG) classification in a 'developed' way.

Wei *et al.* [23] recently introduce the FSFG task. Besides establishing the FSFG problem, they propose a deep neural network model named Piece-wise Classifier Mapping (PCM). By adopting the meta-learning strategy on the auxiliary data set, their model can classify different samples in the testing data set with a few labeled samples. The most critical issue
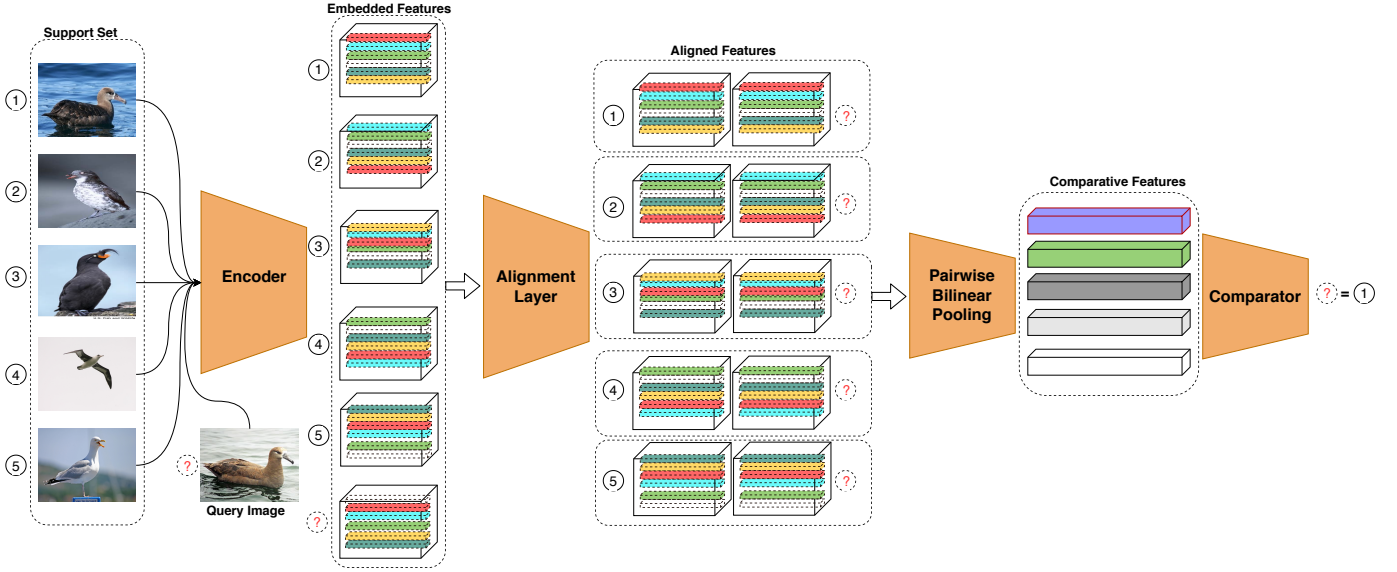
Fig. 2. The framework of LRPABN under the 5-way-1-shot fine-grained image classification setting. The support set contains five labeled samples for each category (marked with numbers) and the query image labeled with a question mark. The LRPPABN model can be divided into four components: Encoder, Alignment Layer, Pairwise Bilinear Pooling, and Comparator. The Encoder extracts coarse features from raw images. Alignment Layer matches the pairs of support and query. Pairwise Bilinear Pooling acts as a fine-grained extractor that captures the subtle features. The Comparator generates the final results.

in FSFG is to acquire subtle and informative image features. In PCM, the authors adopt the naive self-bilinear pooling to extract image representations, which widely used in the state-of-the-art fine-grained object classification [8], [9], [24]. Then with the operation of bilinear feature grouping, the PCM model can generate low-rank subtle descriptors of the original image. Most recently, Li *et al.* [19] propose a covariance pooling [10] to distillate the image representation of each category. These matrix-outer-product based bilinear pooling operations [19], [23] could extract the second-order image features and contains more information than traditional first-order features [24], and thus achieve better performance on FSFG tasks than generic ones.

It is worth noting that both [19] and [23] employ bilinear pooling on the input image itself to enhance the information of original features, which noted as the self-bilinear pooling operation. However, when a human identifies the similar objects, she/he tends to compare them thoroughly in a pairwise way, *e.g.,* comparing the heads of two birds first, then the wings and feet last. Therefore, it is natural to enhance the information during the comparing process when dealing with FSFG classification tasks. Based on this motivation, we propose a novel pairwise bilinear pooling operation on the support and query images to extract the comparative second-order images descriptors for FSFG.

There are a series of works that address the generic few-shot classification by learning to compare [15]–[17], among which the RelationNet [17] achieves state-of-the-art performance by combining a feature encoder and a non-linear relation comparator. However, the matching feature extraction in the RelationNet only concatenates the support and query feature maps in depth (channel) dimension and fails to capture nuanced features for the fine-grained classification.

To address the above issues, we propose a novel end-

to-end FSFG model that captures the fine-grained relations among different classes. This subtle comparative ability of our models is inherently more intelligent than merely modeling the data distribution [17], [19], [23]. The main contributions are summarized as follows:

- *Pairwise Bilinear Pooling*. Existing second-order based FSFG methods [19], [23] enhance the individual encoded features by directly applying the self-bilinear pooling operation. However, such an operation fails to capture more nuanced relations between similar objects. Instead, we uncover the fine-grained relations between different support and query image pairs by using matrix outer product operation, which is called pairwise bilinear pooling. Based on the explicit elicitation of correlative information of pair samples, the proposed operation can extract more discriminate features than existing approaches [1], [17], [23]. More importantly, we introduce a low-rank approximation for the comparative second-order feature, where a set of co-variance low-rank transform matrices are learned to reduce the complexity of the operation.

- *Effective Feature Alignment*. The main advantage of self-bilinear based FSFG methods is the enhancement of depth information for individual spatial positions in the image, which is achieved by the matrix outer product operation on convolved feature maps. Inspired by the self-bilinear pooling operation, we design a simple yet effective alignment mechanism to match the pairwise convolved image features. By exploiting the compact image features alignment, the ablation study shows that the proposed alignment mechanism is crucial for the significant improvements against the baseline model, where only the alignment loss is applied [1].

- *Performance*. By incorporating the feature alignment mechanism and pairwise bilinear pooling operation,

(a) Embedding Network   **(b) Low-rank Bilinear Pooling**   **(c) Comparator Network**
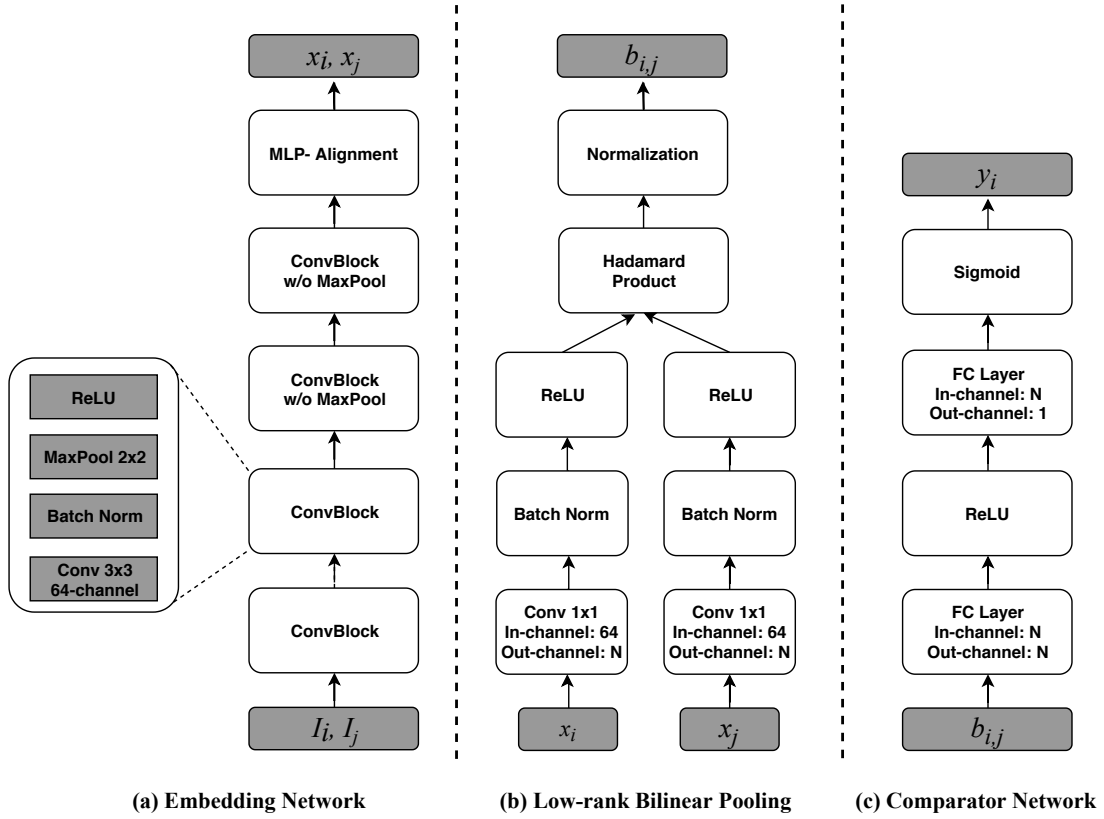
Fig. 3. Detailed network architectures used in LRPABN. (a) The Embedding network with Alignment Layer. (b) Low-Rank Pairwise Bilinear Pooling Layer. (c) The Comparator Network. I_i represents the query image, while I_j is the support image, x_i, x_j are the embedded image features and $b_{i,j}$ represents the comparative bilinear feature. y_i is the predicted label by the comparator.

the proposed model achieves the state-of-the-art performances on four benchmark data sets.

The preliminary version of the proposed model was published at IEEE ICME-19 [1]. The differences between the preliminary version and the new materials are mainly from three aspects:

- A more advanced pairwise pooling operation with a low-rank constraint is proposed. Instead of directly operating the matrix-outer-product as the previous version, we propose to learn multiple transformations for fusing the input image features. By applying these transformations, the proposed model generates more compact and discriminative bilinear features than previous ones, which is verified by the coding-pooling theory [25], [26]. Moreover, we introduce a low-rank approximation of the new bilinear model as our final model to further reduce the computation complexity.
- A novel alignment mechanism is introduced to encourage the input feature pairs of the bilinear operation are matched. Instead of solely relying on the alignment losses, we incorporate a feature position re-arrangement layer with the alignment loss to boost the matching performance.
- More comprehensive experimental results analysis and ablation studies are conducted, and the proposed model achieves superior performances against compared models.

The rest of this paper is organized as follows: Section II

gives a brief introduction of related works on Fine-grained Object classification, Generic Deep Few-shot Learning as well as recent progress in Fine-grained Few-shot Learning. Section III presents the proposed LRPABN method, then Section IV offers the data sets description, experiment setup, and experimental results analysis. Section V concludes the whole paper in the last.

II. RELATED WORK

A. Fine-Grained Object Classification

Fine-grained object classification has been a trending topic in the computer vision research area for years, and most traditional fine-grained approaches use hand-crafted features as image representations [27]–[29]. However, due to the limited representative capacity of hand-crafted features, the performance of this type of method is moderate. In recent years, deep neural networks have developed advanced abilities in the feature extraction and function approximation [30]–[36], bringing significant progress in the fine-grained image classification task [6]–[10], [24], [37], [37]–[48].

Deep fine-grained classification approaches can be roughly divided into two groups: regional feature-based methods [6], [7], [37], [37]–[44] and global feature-based methods [8]–[10], [24], [45]–[48]. In fine-grained image classification, the most informative information generally lies in the discriminate parts of the object. Therefore, regional feature-based approaches

tend to detect such parts first and then fuse them to form a robustness representation of the object. For instance, Zhang *et al.* [39] firstly combine the R-CNN [49] into the fine-grained classifier with a geometric prior, in which the modified R-CNN generates thousands of proposals, the most discriminate ones are then selected for the object classification. In [42], Peng *et al.* adopt two attention modules to localize objects and choose the discriminate parts simultaneously. A spectral clustering method is then employed to align the parts with the same semantic meaning for the prediction. However, the classification performance of these models relies heavily on the parts localization. Getting a well-trained part detector needs the input of a large amount of subtle annotated samples, which is infeasible to obtain. Moreover, the sophisticated regional feature fusion mechanism leads to the increasing complexity of the fine-grained classifier.

On the contrary, global feature-based fine-grained methods [8]–[10], [24], [45]–[48] extract the feature from the whole image without explicitly localize the object parts. Bilinear CNN model (BCNN) [8] is the first work that adopts matrix outer product operation on the original embedded features to generate a second-order representation for fine-grained classification. Li *et al.* [10] (iSQRT-COV) further improve the navie bilinear model by using covariance matrices over the last convolutional features as fine-grained features. iSQRT-COV obtains state-of-the-art performance on both generic and fine-grained datasets.

However, the feature dimensions of the second-order models are the square fold of the naive ones, to reduce the computation complexity, Gao *et al.* [45] propose a compact bilinear pooling operation, which applies Tensor Sketch [50] to reduce the dimensions. Kong *et al.* [46] introduce a low-rank co-decomposition of the covariance matrix that fatherly decreases the complexity, while Kim *et al.* [51] adopt Hadamard product to redefine the bilinear matrix outer product and proposes a factorized low-rank bilinear pooling for multimodal learning. Furthermore, Gao *et al.* [48] devise a hierarchical approach for fine-grained classification using a cross-layer factorized bilinear pooling operation. Inspired by the flexibility and effectiveness of the Hadamard product for extracting the second-order features between visual features and textual features in VQA tasks [51], in our LRPABN model, we propose to adopt the factorized bilinear pooling to approximate pairwise second-order statistics for FSFG task. LRPABN achieves better performance compared to the previous models.

B. Generic Deep Few-shot Learning

The majority of deep few-shot learning methods [52], [53] [15]–[20] follow the strategy of meta-learning [54], [55], which distills the meta-knowledge from batches of auxiliary few-shot tasks. Each auxiliary task mimics the target few-shot tasks with the same support and query images' split. After episodes of training on auxiliary tasks, the trained model can converge speedily to an appreciable local optimum on target data without suffering from the overfitting.

One of the most representative methods is by learning from fine-tuning [56], MAML [52] designs a meta-learning framework that determines the transferable weights for the initialization of the deep neural network. By fine-tuning the network with the limited training samples, the model can achieve reliable performance in a few gradient descent update steps. Moreover, Sachin *et al.* [53] propose a gradient-based method that learns well-initialized weights but also an effective LSTM-based optimizer. Different from this type of approach, our model is free from retraining during the meta-testing stage.

Another class of few-shot learning methods follows the idea of learning to compare [15]–[17], [20], [57]. In general, these approaches consist of two main components: a feature embedding network and a similarity metric. These methods aim to optimize the transferable embedding of both auxiliary data and target data. Consequently, the test images can be identified by the simple nearest neighbor classifier [15], [16], deep distance matrix based classifier [17], or cosine-distance based classifier [20], [57]. Considering the FSFG task requires the more advanced information processing ability, we propose to capture more nuanced features from the images pairs other than the first-order extraction used in leaning to compare approaches.

C. Few-shot Fine-grained Learning

Most recently, Wei *et al.* [23] propose the first FSFG model by employing two sub-networks to tackle the problem jointly. The first component is a self-bilinear encoder, which adopts the matrix outer product operation on convolved features to capture subtle image features, while the second one is a mapping network that learns the decision boundaries of the input data. Li *et al.* [19] further replace the naive self-bilinear pooing as the covariance pooling. Moreover, they design a covariance metric to generate relation scores. However, self-bilinear pooling [19], [23] cannot extract comparative features between pairs of images, and the dimension of pooled features is usually large. Pahde *et al.* [58] propose a cross-modality FSFG model, which embeds the textual annotations and image features into a common latent space. They also introduce a discriminative text-conditional GAN for the sample generation, which selects the representative samples from the auxiliary set. However, it is both computation and time consuming to obtain rich annotations for the fine-grained samples.

III. METHODOLOGY

In this section, we present the problem formulation of FSFG first. Then the proposed LRPABN model is introduced, including the Low-Rank Pairwise Bilinear Polling operation and Feature Alignment Layer, which are the core parts of LRPABN. The detailed network architecture of LRPABN is given at last.

A. Problem Definition

Given a Fine-Grained target data set $\mathcal{T}$:

$$\mathcal{T} = \left\{ \mathcal{B} = \{(\overline{x}_b, \overline{y}_b)\}_{b=1}^{K \times \tilde{C}} \right\} \cup \left\{ \mathcal{N} = \{(\overline{x}_v)\}_{v=1}^{V} \right\}, \quad (1)$$
$$\overline{y}_b \in \{1, \tilde{C}\}, \overline{x} \in \mathcal{R}^N, \mathcal{B} \cap \mathcal{N} = \emptyset, V \gg K \times \tilde{C}.$$

For the FSFG task, the target data set $\mathcal{T}$ contains two parts: the labeled subset $\mathcal{B}$ and the unlabeled subset $\mathcal{N}$, where samples from each subset are fine-grained images. The model needs to classify the unlabeled data $\overline{x}_v$ from $\mathcal{N}$ according to a few labeled samples from $\mathcal{B}$, where $\overline{y}_b$ is the ground-truth label of sample $\overline{x}_b$. If the labeled data in the target data set contains K labeled images for each of $\tilde{C}$ different categories, the problem is noted as $\tilde{C}$-way-K-shot.

In order to obtain an ideal model on such a data set, Few-Shot learning usually employs a fully annotated data set, which has similar property or data distribution with $\mathcal{T}$ as an auxiliary data set $\mathcal{A}$:

$$
\begin{aligned}
\mathcal{A} = \left\{ \mathcal{S} = \{(x_i, y_i)\}_{i=1}^{L} \right\} \cup \left\{ \mathcal{Q} = \{(x_j, y_j)\}_{j=1}^{J} \right\}, \\
y_i, y_j \in \{1, C\}, x \in \mathcal{R}^N, \mathcal{S} \cap \mathcal{Q} = \emptyset, \mathcal{A} \cap \mathcal{T} = \emptyset,
\end{aligned}
\tag{2}
$$

where x_i/y_i and x_j/y_j represent images and their corresponding labels. In each round of training, the auxiliary data set $\mathcal{A}$ is randomly separated into two parts: support data set $\mathcal{S}$, and query data set $\mathcal{Q}$. With setting $L = K \times \tilde{C}$, we can mimic the composition of the target data set in each iteration. Then $\mathcal{A}$ is employed to learn a meta-learner $\mathfrak{F}$, which can transfer the knowledge from $\mathcal{A}$ to target data $\mathcal{T}$. Once obtained meta-learner, it can be fine-tuned with labeled target data set $\mathcal{B}$, and finally, classify the samples from $\mathcal{N}$ into their corresponding categories [1], [15], [17]–[20], [23].

B. The proposed LRPABN

The whole framework of LRPABN is shown in Figure 2, and detailed architecture is given in Figure 3. Different from traditional few-shot embedding structures [15]–[17], we add the Low-Rank Pairwise Bilinear Pooling to construct the fine-grained image feature extractor. Moreover, we modify the non-linear comparator [17] and apply it to the fine-grained task. As the Figure 2 shows, given the support set consisting of five classes with one image per class, an Encoder that is trained with the auxiliary data $\mathcal{A}$ can extract the first-order image features from the raw images, then the Alignment Layer coordinates the embedded feature in support set with the query image feature in pairs. Next, the Low-Rank Bilinear Pooling is used to generate the comparative second-order image representation from the embedded feature pairs. Finally, the Comparator assigns the optimal label to the query from support labels in consonance with the similarity between the query and different support classes.

Pairwise bilinear pooling layer aims to capture the nuanced comparative features of image pairs by employing the bilinear pooling operation, which plays a crucial role in determining the relations between support and query pairs. However, it is natural that if a couple of inputs are not well-matched, the pooled features cannot result in the maximum classification performance gain. Therefore, we introduce an alignment layer which consists of a Multi-Layer Perceptron (MLP) and feature alignment losses to guarantee the registration of the pairs.

1) Pairwise Bilinear Pooling Layer: The Bilinear CNN [8] for the image classification can be defined as a quadruple:

$$
\begin{aligned}
&B\text{-}CNNs = (\mathfrak{E}_I, \mathfrak{E}_{II}, \mathfrak{f}_b, \mathcal{C}), \\
&\mathfrak{E} : \mathcal{I} \longrightarrow \mathcal{X} \in \mathcal{R}^{c \times h \times w}, \\
&\mathfrak{f}_b(\mathcal{I}, \mathfrak{E}_I, \mathfrak{E}_{II}) = \frac{1}{hw} \sum_{i=1}^{hw} f_{\alpha,i} f_{\beta,i}^T,
\end{aligned}
\tag{3}
$$

where $\mathfrak{E}_I$ and $\mathfrak{E}_{II}$ are encoders for each input stream, $\mathfrak{f}_b$ is the self-bilinear pooling operation, and $\mathcal{C}$ represents the classifier. $\mathcal{I} \in \mathcal{R}^{H \times W \times C}$ is the input image with H height, W width, and C color channels. Through encoder $\mathfrak{E}$, the input image is transformed into a tensor $\mathcal{M} \in \mathcal{R}^{h \times w \times c}$, which has c feature channels, and h, w indicate the height and width of the embedded feature map. Given two encoders $\mathfrak{E}_I : \mathcal{I} \longrightarrow \mathcal{X}_\alpha \in \mathcal{R}^{c_1 \times h \times w}$ and $\mathfrak{E}_{II} : \mathcal{I} \longrightarrow \mathcal{X}_\beta \in \mathcal{R}^{c_2 \times h \times w}$, $f_{\alpha,i} \in \mathcal{R}^{c_1 \times 1}$ and $f_{\beta,i} \in \mathcal{R}^{c_2 \times 1}$ denote feature vectors at specific spatial location i in each feature map $\mathcal{X}_\alpha$ and $\mathcal{X}_\beta$, where $i \in [1, hw]$. The pooled feature is a $c_1 \times c_2$ vector. $\mathcal{C}$ is a fully-connected layer trained with cross-entropy loss.

Different from the conventional self-bilinear pooling operates on pairs of embedded features from the same image, in our **pairwise** bilinear pooling layer, the input pair is generated from the different source sets, *i.e.*, $\mathcal{I}_\mathcal{A} \in \mathcal{S}$ and $\mathcal{I}_\mathcal{B} \in \mathcal{Q}$. With the encoder $\tilde{\mathfrak{E}}$, the pairwise bilinear pooling $\mathfrak{f}_{pb}$ can be defined as:

$$
\begin{aligned}
&\mathfrak{f}_{pb}(\mathcal{I}_\mathcal{A}, \mathcal{I}_\mathcal{B}, \tilde{\mathfrak{E}}) = \tilde{\mathfrak{E}}(\mathcal{I}_\mathcal{A}) \tilde{\mathfrak{E}}(\mathcal{I}_\mathcal{B})^T, \\
&\tilde{\mathfrak{E}} : \mathcal{I} \longrightarrow \mathcal{X} \in \mathcal{R}^{c \times hw}.
\end{aligned}
\tag{4}
$$

It is worth noting that in the pairwise bilinear pooling, we only have one shared embedding function $\tilde{\mathfrak{E}}$. Different from the self-bilinear pooling that operates on the same input image, pairwise bilinear pooling uses a matrix-outer-product on two different samples. Equation (4) is the pairwise bilinear pooling used in our previous work [1]. However, the pooled pairwise feature is a $c_1 \times c_2$ vector, which results in a square growth of the original feature dimension. For example, with an embedding network AlexNet [59], $c_1 = c_2 = 512$, the pairwise bilinear pooling generates a $512 \times 512 = 262{,}144$-d representation. As reported in [45], in such a high-dimensional feature space, less than 5% of dimensions are informative. Moreover, recent research [26] also indicates that the matrix-outer-product based bilinear pooling suffers from redundancy and burstiness issues because of the rank-one property of bilinear features. The dimensionality of matrix-outer-product based bilinear features incites the heavy computational loads as well as burstiness phenomenons.

To overcome this shortcoming of previous proposed pairwise bilinear pooling, inspired by the Factorized Bilinear Pooling [51] applied in the visual-question-answer task, we further propose a **low-rank** pairwise bilinear pooling operation. For the given $\mathcal{X}_\mathcal{A} = \left[\mathbf{x}_1^A, \mathbf{x}_2^A, \cdots, \mathbf{x}_{hw}^A\right]$ and $\mathcal{X}_\mathcal{B} = \left[\mathbf{x}_1^B, \mathbf{x}_2^B, \cdots, \mathbf{x}_{hw}^B\right]$ from Equation (4), where $\mathbf{x}_j \in \mathcal{R}^{c \times 1}$ stands for any spatial feature vector in $\mathcal{X}$, $j \in [1, hw]$. The low-rank pairwise bilinear can be formulated as:

$$
z_j = \left(\mathbf{x}_j^A\right)^T W_i \mathbf{x}_j^B,
\tag{5}
$$

where $W_i \in \mathcal{R}^{c \times c}$ is a projection matrix, $\mathbf{x}_j^A$ and $\mathbf{x}_j^B$ are the feature vectors from $\mathcal{X}_A$ and $\mathcal{X}_B$ in the same position j, separately. Equation (5) fuses these feature vectors into a common scalar z_j. Given a set of projection matrices $\mathcal{W} = [W_1, W_2, \cdots, W_n] \in \mathcal{R}^{c \times c \times n}$, the redefined bilinear feature of any position j is $\mathbf{z}_j = [z_1, z_2, \cdots, z_n]^T$. n is the dimension of this bilinear feature. Then the comparative bilinear representation for the original pairs can be represented as $\mathcal{Z} = [\mathbf{z}_1, \mathbf{z}_2, \cdots, \mathbf{z}_{hw}]$. It is worth noticing that Equation (4) is different from Equation (5), which adopts projection matrix W_i in learning the bilinear feature. Moreover, in Equation (5), the dimension of comparative bilinear feature is n that can be far smaller than $c \times c$ in Equation (4). In this way, the model gets a low-rank approximation for the original comparative bilinear feature.

In Equation (5), the learned projection $\mathcal{W}$ requires $c \times c \times n$ parameters, where c = 64 and n = 512 in our implementation, *i.e.,* 2,097,152 parameters in total, which requires a large amount of memory footprint, inference time, and computational complexity. To solve this problem, we present a low-rank approximation of W_i:

$$
\begin{aligned}
z_j &= \left(\mathbf{x}_j^A\right)^T W_i \mathbf{x}_j^B \\
&= \left(\mathbf{x}_j^A\right)^T U_i V_i^T \mathbf{x}_j^B \\
&= U_i^T \mathbf{x}_j^A \circ V_i^T \mathbf{x}_j^B,
\end{aligned} \tag{6}
$$

where $U_i \in \mathcal{R}^{c \times 1}$ and $V_i \in \mathcal{R}^{c \times 1}$, $\circ$ denotes the Hadamard product. Equation (6) is the **final form** of low-rank pairwise bilinear pooling, which applies projection matrix and matrix factorization to approximate a full low-rank bilinear model (Equation (5)). In Equation (6), it needs $2nc$ parameters to generate the pairwise bilinear feature. Therefore, the spatial complexity of the required parameters is reduced from $\mathcal{O}(nc^2)$ to $\mathcal{O}(nc)$. It is worth noting that there are two low-rank approximations applied in the final form of our newly proposed model LRPABN. One is to tackle the information redundancy and burstiness issue of the matrix-outer-product based bilinear pooling (Equation (4) to (5)), the other is to apply the low-rank matrix factorization to approximate the learned transformations (Equation (5) to (6)). The proposed LRPABN is different from [48], [51], where [51] adopts the factorized bilinear pooling to fuse the multi-modal features, and [48] operates on convolutional features of the same image. Our method conducts on pairs of support and query images. To our best knowledge, LRPABN is the first work that extracts the low-rank bilinear feature from pairs of distinct images for FSFG tasks.

Theoretically, the previous proposed model [1] belongs to the category of matrix-outer-product bilinear pooling, which has been proved as a similarity-based coding-pooling [25], [26]. As [26] (Corollary 2) indicates that such bilinear pooling has the unstable dictionary, which is determined by the input pairs, therefore it is inconsistent for all data. This local dictionary can not capture the global geometry of the whole data space, which results in burstiness issues. However, the newly proposed low-rank pairwise bilinear model (6) is a type of factorized bilinear coding (Equation (24) in [26]), which learns a global dictionary from the entire data space

in a scalable way, thus achieves better performance than the previous one.

2) Feature Alignment Layer: The self-bilinear pooling operates on the same image, which means in any spatial location of the embedded feature pairs, the operating features are entirely aligned. However, since the proposed pairwise bilinear pooling operates on different inputs, the encoded features may not always be matched. To overcome this obstacle, in our previous work [1], we devise two alignment losses to match the input pairs in the embedding space simultaneously during the training stage, which aims at encouraging the embedding network to generate well-matched features in the testing stage. However, it may be hard to obtain the desired embedding network that fully aligns feature pairs by merely adopting the alignment losses.

Therefore, we design a new feature alignment mechanism inspired by the PointNet [60]. Given a position transform function $\mathbf{T}$ and the encoded feature $\mathcal{X} = [\mathbf{x}_1, \mathbf{x}_2, \cdots, \mathbf{x}_{hw}]$, the transformed feature can be computed as follows:

$$
\begin{aligned}
\mathcal{X}' &= \mathcal{X}\mathbf{T}, \\
s.t. \; \mathbf{T}\mathbf{T}^T &= \mathbf{I},
\end{aligned} \tag{7}
$$

where $\mathbf{T} \in \mathcal{R}^{hw \times hw}$ and $\mathbf{I}$ is an identity matrix. The transformed feature is noted as $\mathcal{X}' = [\mathbf{x}'_1, \mathbf{x}'_2, \cdots, \mathbf{x}'_{hw}]$, in which only the positions of the original feature vectors are rearranged. The transform matrix can be learned with a shallow neural network. Moreover, to ensure the effectiveness of the alignment, we further design two feature alignment losses as follows:

$$
Align_{loss_1}(\mathcal{I}_A, \mathcal{I}_B, \tilde{\mathfrak{E}}) = MSE(\tilde{\mathfrak{E}}(\mathcal{I}_A), \tilde{\mathfrak{E}}(\mathcal{I}_B)\mathbf{T}), \tag{8}
$$

where $\tilde{\mathfrak{E}}$ is the feature encoder. The first $Align_{loss_1}$ loss is a rough approximation of two embedded image descriptors that minimizing the Euclidean distances of two transformed features.

$$
\begin{aligned}
Align_{loss_2}(\mathcal{I}_A, \mathcal{I}_B, \mathfrak{O}) &= MSE(\mathfrak{O}(\mathcal{I}_A), \mathfrak{O}(\mathcal{I}_B)\mathbf{T}), \\
\mathfrak{O}(\mathcal{I}) &= \sum_1^c \tilde{\mathfrak{E}}(\mathcal{I}), \; \tilde{\mathfrak{E}} : \mathcal{I} \longrightarrow \mathcal{X} \in \mathcal{R}^{c \times hw}.
\end{aligned} \tag{9}
$$

The second $Align_{loss_2}$ loss is a more concise feature alignment loss. Inspired by the pooling operation, we sum all the embedded features ($\mathcal{X} \in \mathcal{R}^{c \times hw}$) along with the channel dimension ($\mathcal{R}^c$) first. And then, we measure the MSE of summed features. By training with the proposed alignment losses, we encourage the model to automatically learn the matching features to generate a better pairwise bilinear feature. It is worth noting that the alignment mechanism utilizes feature position rearrangement matrix $\mathbf{T}$ on one image features ($\tilde{\mathfrak{E}}(\mathcal{I}_B)$) to match the target feature ($\tilde{\mathfrak{E}}(\mathcal{I}_A)$). $\mathcal{I}_B$ can be either the support or query image, and in our implementation, we choose the support image as $\mathcal{I}_B$. Under the supervision of alignment losses, the model can generate more compactly matched feature pairs compared to the previous method.

3) Comparator: As Figure 2 indicates, after passing through the above layers, the pairwise comparative bilinear features are sent to a comparator. This module aims to learn the relations between the query images and support classes. In the one-way-K-shot setting, the support classes are represented by a single image, where for $\tilde{C}$-way-K-shot setting, the support classes are computed as the sum value of embedded features of K images in each class, *i.e.*, for each query image, the model generates $\tilde{C}$ comparative bilinear features corresponding to each class. For a pair of query image i and support class j, the comparative bilinear feature can be represented as $\mathcal{Z}_{i,j}$, where $\mathcal{Z} = [\mathbf{z}_1, \mathbf{z}_2, \cdots, \mathbf{z}_{hw}]$, the relation score of i and j is computed as:

$$r_{i,j} = \mathcal{C}(\mathcal{Z}_{i,j}),$$
$$j = 1, 2, \ldots W; \quad i = 1, 2, \ldots, K \times \tilde{C}, \quad (10)$$

where $\mathcal{C}$ is the comparator, and $r_{i,j}$ is the relation score of query i and class j.

4) Model Training: The training loss $\mathcal{L}$ in our bilinear comparator is the MSE loss, which regresses the relation score to the images label similarity. At a certain iteration during the episodic training, there exists m query features and n support class features in total, $\mathcal{L}$ is thus defined as:

$$\mathcal{L} = \sum_{i=1}^{m} \sum_{j=1}^{n} \left(r_{i,j} - \delta\left(y_i = y_j\right)\right)^2, \quad (11)$$

where $\delta\left(y_i = y_j\right)$ is the indicator, it equals to one when $y_i = y_j$ and zeroes otherwise. The LRPABN has two optional alignment losses $Align_{loss_1}$ and $Align_{loss_2}$. We back-propagate the gradients when the alignment losses are computed immediately. That is, during the training stage, the model will be updated twice in one iteration.

C. Network Architecture

The detailed network architecture is shown in Figure 3. It consists of three parts: Embedding Network, Low-rank Bilinear Pooling Layer, and Comparator Network.

Embdeeding Network: For a fair comparison with the state-of-the-art generic few-shot and FSFG approaches, we adopt the same encoder structure in [15]–[19]. It consists of four convolutional blocks, where each block contains a 2D convolutional layer with a 3×3 kernel and 64 filters, a batch normalization layer, and a ReLU layer. Moreover, for the first two convolutional blocks, a 2×2 max-pooling layer is added. For simplicity, we integrate the feature alignment layer into the embedding network as the first-order feature extractor, indicated in Figure 3.(a). Unlike the alignment mechanism used in [42], [60], we devise a simple two layers MLP with the Regulation (7). As our alignment mechanism is inspired by PointNet [60], which originally adopts a deeper network to learn the transformation matrix $\mathbf{T}$. However, in FSFG, we find that a shallow MLP network is more efficient in learning a good transformation $\mathbf{T}$. Besides, two optional alignment losses (8), (9) are applied in the alignment layer to generate the well-matched pairwise features.

Low-rank Bilinear Pooling Layer: For the Low-Rank Pairwise Bilinear Pooling layer in Figure 3.(b), we use a

data set	CUB Birds	DOGS	CARS	NABirds
C_{total}	200	120	196	555
$C_{\mathcal{A}}$	150	90	147	416
$C_{\mathcal{T}}$	50	30	49	139

convolutional layer with 1×1 kernel followed by the batch normalization and a ReLU layer. The Hadamard product and normalization layers are appended to generate the comparative bilinear features.

Comparator Network: The comparator is made up of two Fully Connected (FC) layers. A ReLU, as well as a Sigmoid nonlinearity layer, are applied to generate the final relation scores, as Figure 3.(c) shows.

IV. EXPERIMENT

In this section, we evaluate the proposed LRPABN on four widely used fine-grained data sets. First, we give a brief introduction to these data sets. Then we describe the experimental setup in detail. Finally, we analyze the experimental results of the proposed models and compare them with other few-shot learning approaches. For a fair comparison, we conduct two groups of experiments on these data sets, for the first group, we follow the setting, which Wei *et al.* [23] and [1] used, while for the second group, we follow the newest settings in the recent few-shot methods [19], [20].

A. Datasets

There are four data sets used to investigate the proposed models:

- CUB Birds [2] contains 200 categories of birds and a total of 11,788 images.
- DOGS [4] contains 120 categories of dogs and a total of 20,580 images.
- CARS [5] contains 196 categories of cars and a total of 16,185 images.
- NABirds [3] contains 555 categories of north American birds and a total of 48,562 images.

In Section III-A, we randomly divide these data sets into two disjoint sub-sets: the auxiliary data set $\mathcal{A}$, and the target data set $\mathcal{T}$. For the first group of experiments, we use the splits of PCM [23], as shown in Table I. For the second group, we adopt the data set splits of Li's [19], [20], as indicated in Table II. Both of these methods do not use the NABirds data set. Thus, for this data set only, we do our splits.

B. Experimental Setup

In each round of training and testing, for the one-shot image classification setting, the support sample number in each class

equals 1 (in both $\mathcal{B}$ and $\mathcal{S}$, $K = 1$). Therefore, we use the embedded features of these support samples as the class features, *i.e.*, $\tilde{\mathfrak{E}}(\mathcal{I_B})$. For the few-shot setting, we extract the class features by summing all the embedded support features in each category. In our experiments, we compare the below FS as well as FSFG approaches:

The state-of-the-art methods:

- RelationNet [17], a state-of-the-art generic few-shot method proposed in CVPR 2018. It uses a mini-network to learn the similarity between the query image and the support class.
- DN4 [20], the newest generic few-shot method published in CVPR 2019. By using a deep nearest neighbor neural network, DN4 can aggregate the discriminative information of local features and thus improve the final classification performance.
- PCM [23], the first FSFG model published in IEEE TIP 2019. It adopts a self-bilinear model to extracts the fine-grained features of the image and achieves excellent performance on several FSFG tasks.
- CovaMNet [19], a newest FSFG model published in AAAI 2019. It replaces the bilinear pooling with co-variance bilinear pooling and achieves state-of-the-art performance on FSFG classification.

The PABN models [1], our previous work for FSFG tasks that uses pairwise bilinear pooling (4) without feature alignment transform function (7):

- $\text{PABN}_{w/o}$, this model does not use alignment loss on embedded pair features.
- PABN_{niv} and PABN_{cpt} are the models that adopt the alignment loss $Align_{loss_1}$ and $Align_{loss_2}$ for feature alignment, separately. As Section III-B2 discussed, $Align_{loss_1}$ loss is a naive alignment loss where $Align_{loss_2}$ is a more compact loss.

The PABN+ models, these models apply the proposed alignment layer into PABN models, which aims to investigate the effectiveness of the proposed feature alignment transform function (7):

- PABN+_{niv} and PABN+_{cpt} are the models that adopt the alignment loss $Align_{loss_1}$ and $Align_{loss_2}$ in the alignment layer (7).
- PABN+_{cons} adopts Cosine loss on the embedded features in the alignment layer (7).

The LRPABN models, we replace the naive pairwise bilinear pooling (4) with the proposed low-rank bilinear pooling (6), and apply the proposed novel feature alignment layer (7) into the LRPABN models:

- LRPABN_{niv} and LRPABN_{cpt}, which use the alignment loss $Align_{loss_1}$ and the loss $Align_{loss_2}$ in the alignment layer, respectively.

In the first experiment, the LRPABN models are compared with RelationNet, PCM, and our previous proposed PABN models. We follow the data splits (Table I) of PCM and PABN. All of these approaches do not contain the validation data set.

In the second experiment, besides the RelationNet, PABN+ models, and the proposed LRPABN models, we compare the newest state-of-the-art few-shot method DN4 and the newest

TABLE II
THE CLASS SPLIT OF FOUR FINE-GRAINED DATA SETS WHICH IS THE SAME AS [19], [20]. C_{total} IS THE ORIGINAL NUMBER OF CATEGORIES IN THE DATA SETS, $C_{\mathcal{A}.Train}$ IS THE NUMBER OF TRAINING DATA CATEGORIES IN THE AUXILIARY DATA SETS, $C_{\mathcal{A}.Val}$ IS THE NUMBER OF VALIDATION DATA CATEGORIES IN SEPARATED AUXILIARY DATA SETS AND $C_{\mathcal{T}}$ IS THE NUMBER OF CATEGORIES IN TARGET DATA SETS.

data set	CUB Birds	DOGS	CARS	NABirds
C_{total}	200	120	196	555
$C_{\mathcal{A}.Train}$	120	70	130	350
$C_{\mathcal{A}.Val}$	30	20	17	66
$C_{\mathcal{T}}$	50	30	49	139

FSFG approach CovaMNet. To fair compare, we use the same data splits (Table II) and the training strategy of DN4 and CovaMNet.

For all the comparing methods, we conduct both 5-way-1-shot and 5-way-5-shot classification experiments. In the training stage of the first group of experiments, both 5-way-1-shot and 5-way-5-shot experiments have 15 query images, which means there are $15 \times 5 + 1 \times 5 = 80$ images and $15 \times 5 + 5 \times 5 = 100$ images in each mini-batch, respectively. For the testing stage, we follow the RelationNet [17] that have one query for 5-way-1-shot and five queries for 5-way-5-shot in each mini-batch. In both the training and testing stages of the second group of experiments, we randomly select 15 and 10 queries from each category for the 5-way-1-shot and 5-way-5-shot settings, which is the same setting with [19], [20].

For fair comparisons, we select the optimal models using the same validation strategies as [17] for the first group of experiments and [19], [20] for the second group of experiments, separately. In the first group, we randomly sample and construct 100,000 episodes to train the LRPABN and PABN+ models. In each episode, there only contains one learning task, while in the second group, we randomly select 10,000 episodes for training, and in each episode, 100 tasks are randomly batched to train the models. For LRPABN models, we set the dimension of the pairwise bilinear feature as 512, where the feature dimension of PABN and PABN+ is $64 \times 64 = 4096$. In training, the learning rate of parameters is decayed by 0.5 every 10,000 epochs using the StepLR schedule in PyTorch. We resize all the input images from all data sets to 84×84. All experiments use Adam optimize method with an initial learning rate of 0.001, and all models are trained end-to-end from scratch.

C. Results and Analysis

To the best of our knowledge, there are only a few methods proposed for FSFG image classification [1], [19], [23], [58], [61]. [58] uses larger auxiliary data set than our methods, and [61] is only applied for image retrieval tasks. It is unfair to compare these methods directly. Therefore we compare our LRPABN with PCM [23], PABN [1], and CovaMNet [19]. We also compare our methods with the state-of-the-art generic few-shot learning method RelationNet [17] and DN4 [20]. The original implementation of RelationNet does not report the

TABLE III
FEW-SHOT CLASSIFICATION ACCURACY (%) COMPARISONS ON FOUR FINE-GRAINED DATA SETS. THE SECOND-HIGHEST-ACCURACY METHODS ARE HIGHLIGHTED IN BLUE COLOR. THE HIGHEST-ACCURACY METHODS ARE LABELED WITH THE RED COLOR. '-' DENOTES NOT REPORTED. ALL RESULTS ARE WITH 95% CONFIDENCE INTERVALS WHERE REPORTED.

Methods	CUB Birds		CARS		DOGS		NABirds	
	1-shot	5-shot	1-shot	5-shot	1-shot	5-shot	1-shot	5-shot
PCM [23]	42.10±1.96	62.48±1.21	29.63±2.38	52.28±1.46	28.78±2.33	46.92±2.00	-	-
RelationNet	63.77±1.37	74.92±0.69	56.28±0.45	68.39±0.21	51.95±0.46	64.91±0.24	65.17±0.47	78.35±0.21
$PABN_{w/o}$	65.99±1.35	76.90±0.21	55.65±0.42	67.29±0.23	54.77±0.44	65.92±0.23	67.23±0.42	79.25±0.20
$PABN_{niv}$	65.04±0.44	76.46±0.22	55.89±0.42	68.53±0.23	54.06±0.45	65.93±0.24	66.62±0.44	79.31±0.22
$PABN_{cpt}$	**66.71±0.43**	76.81±0.21	56.80±0.45	68.78±0.22	**55.47±0.46**	66.65±0.23	67.02±0.43	79.02±0.21
$PABN+_{niv}$	66.68±0.42	76.83±0.22	55.35±0.44	67.67±0.22	54.51±0.45	66.60±0.23	66.60±0.44	**81.07±0.20**
$PABN+_{cpt}$	65.44±0.43	77.19±0.22	57.36±0.45	69.30±0.22	54.66±0.45	**66.74±0.22**	67.39±0.43	79.95±0.21
$PABN+_{cos}$	66.45±0.42	**78.34±0.21**	57.44±0.45	68.59±0.22	54.18±0.44	65.70±0.23	66.74±0.44	80.58±0.20
$LRPABN_{niv}$	64.62±0.43	**78.26+0.22**	**59.57±0.46**	**74.66±0.22**	**54.82±0.46**	66.62±0.23	**68.40±0.44**	80.17±0.21
$LRPABN_{cpt}$	**67.97±0.44**	78.04±0.22	**63.11±0.46**	**72.63±0.22**	54.52±0.47	**67.12±0.23**	**68.04±0.44**	**80.85±0.20**

TABLE IV
FEW-SHOT CLASSIFICATION ACCURACY (%) COMPARISONS ON FOUR FINE-GRAINED DATA SETS. THE HIGHEST-ACCURACY AND SECOND-HIGHEST-ACCURACY METHODS ARE HIGHLIGHTED IN RED AND BLUE, SEPARATELY. ALL RESULTS ARE WITH 95% CONFIDENCE INTERVALS WHERE REPORTED.

Methods	CUB Birds		CARS		DOGS		NABirds	
	1-shot	5-shot	1-shot	5-shot	1-shot	5-shot	1-shot	5-shot
RelationNet	59.82±0.77	71.83±0.61	56.02±0.74	66.93±0.63	44.75±0.70	58.36±0.66	64.34±0.81	77.52±0.60
CovaMNet	58.51±0.94	71.15±0.80	56.65±0.86	71.33±0.62	**49.10±0.76**	**63.04±0.65**	60.03±0.98	75.63±0.79
DN4	55.60±0.89	**77.64±0.68**	**59.84±0.80**	**88.65±0.44**	45.41±0.76	**63.51±0.62**	51.81±0.91	**83.38±0.60**
$PABN+_{niv}$	**63.56±0.79**	75.23±0.59	53.39±0.72	66.56±0.64	45.64±0.74	58.97±0.63	**66.96±0.81**	80.73±0.57
$PABN+_{cpt}$	63.36±0.80	74.71±0.60	54.44±0.71	67.36±0.61	45.65±0.71	61.24±0.62	66.94±0.82	79.66±0.62
$PABN+_{cos}$	62.02±0.75	75.35±0.58	53.62±0.73	67.15±0.60	45.18±0.68	59.48±0.65	66.34±0.76	80.49±0.59
$LRPABN_{niv}$	62.70±0.79	75.10±0.61	56.31±0.73	70.23±0.59	**46.17±0.73**	59.11±0.67	66.42±0.83	80.60±0.59
$LRPABN_{cpt}$	**63.63±0.77**	**76.06±0.58**	**60.28±0.76**	**73.29±0.58**	45.72±0.75	60.94±0.66	**67.73±0.81**	**81.62±0.58**

results on four fine-grained data sets. For fair comparisons, we use the open-source code of the RelationNet[1] to conduct the FSFG image classification on these data sets.

In the first group of experiments, we compute both one-shot and five-shot classification accuracies on the four data sets by averaging on 10,000 episodes in testing. We show the experimental results of 10 compared models in Table III. As the table shows, the proposed LRPABN models achieve significant improvements on both 1-shot and 5-shot classification tasks on all data sets compared to the state-of-the-art FSFG methods and generic few-shot methods, which indicates the effectiveness of the proposed framework.

More specifically, the LRPABN, PABN+, and PABN mod-

els [1] both obtain around 10% to 30% higher in classification accuracy than PCM [23], which demonstrates that the comparative pairwise bilinear feature outperforms the self-bilinear feature on FSFG tasks. Besides, the pairwise bilinear feature-based approaches achieve better classification performances than RelationNet [17] that validates the extraction of second-order image descriptors surpasses the naive concatenation of feature pairs [17] for FSFG problems.

From Table III, compared to PABN models, PABN+ and LRPABN models obtain a definite classification performance boost. For instance, the $PABN+_{niv}$ gains 1.64% and 0.37% improvements over $PABN_{niv}$ in one-shot and five-shot setting on CUB Birds data, while $LRPABN_{cpt}$ achieves 1.26% and 1.23% improvements over $PABN_{cpt}$ in one-shot and five-shot setting on CUB Birds data set. These results demonstrate

---

[1] https://github.com/floodsung/LearningToCompare_FSL

TABLE V
ABLATION STUDY OF LRPABN WITH DIFFERENT COMPONENTS. THE RESULTS ARE REPORTED WITH 95% CONFIDENCE INTERVALS. MODEL SIZE
INDICATES THE NUMBER OF PARAMETERS FOR EACH MODEL, AND THE INFERENCE TIME IS THE TESTING TIME FOR EACH INPUT QUERY IMAGE.

Methods	CUB data set				
	1-shot (%)	5-shot (%)	Model Size	Inference Time (10^{-3} s)	Bilinear Feature Dim
$PABN_{cpt}$ [1]	66.71±0.43	76.81±0.21	375,361	8.65	4096
$PABN+_{cpt}$	65.44±0.43	77.19±0.22	505,682	8.94	4096
$PABN_{new}$	67.39±0.43	78.87±0.21	2,373,819	78.40	512
LRPABN	66.56±0.43	77.60±0.22	213,930	2.23	512
$LRPABN_{only\_cpt}$	66.72±0.44	77.98±0.21	213,930	2.23	512
$LRPABN_{cpt}$	67.97±0.44	78.04±0.22	344,251	2.53	512
DN4 [20]	60.02±0.85	79.64±0.67	112,832	15.20	-

TABLE VI
DISCUSSION ABOUT INPUT IMAGE SIZE FOR FSFG.

Methods	CUB data set		
	1-shot (%)	5-shot (%)	Image Size
PCM:AlexNet [23]	42.10±1.96	62.48±1.21	224 × 224
$LRPABN_{cpt}$:AlexNet	59.34±0.48	69.08±0.24	224 × 224
$LRPABN_{cpt}$:AlexNet	66.19±0.46	75.05±0.23	448 × 448
$LRPABN_{cpt}$:Conv4	67.97±0.44	78.04±0.22	84 × 84
DN4:Conv4 [20]	60.02±0.85	79.64±0.67	84 × 84

that the effectiveness of the proposed feature alignment layer. It can be observed from Table III that LRPABN models achieve the best or second-best classification performance on nearly all data sets compared to other methods under various experimental settings. For CARS data, the $LRPABN_{cpt}$ obtains 5.67%, 6.31%, 6.83% significant improvements over $PABN+_{cos}$, $PABN_{cpt}$ and RelationNet on 1-shot-5-way task, while achieves 5.36%, 5.88%, 6.27% improvements against $PABN+_{cpt}$, $PABN_{cpt}$ and RelationNet on 5-shot-5-way setting, which validates the effectiveness of our low-rank pairwise bilinear pooling. It is worth noting that the dimension of the pairwise bilinear feature in LRPABN is 512, where the corresponding feature dimension of PABN and PABN+ is 4096. LRPABN models adopt the low-rank factorized bilinear pooling operation, which can learn a set projection transform functions fusing the feature pairs, as discussed in Equation (6). Each of the projection function represents a pattern of coalescing the image pairs in feature channels over all the matching positions. Meanwhile, the naive pairwise bilinear pooling in the PABN and PABN+ approaches only applies the matrix outer product on feature pairs once to merge them. Therefore, the LRPABN models can obtain more types of feature extraction than PABN and PABN+ models, which in turn achieves better performance with smaller feature dimensions.

For a further analysis of our models, we conduct an additional experiment on these four data sets comparing the LR-

PABN models with DN4 and CovaMNet. In this experiment, we also compare the PABN+ models. Moreover, we use the same setting to rerun the RelationNet on four data sets as the baseline method. We follow the same data set split with DN4 and CovaMNet, the original papers of these two papers do not report the results on CUB Birds (CUB-2011) [2] and NABirds [3], so we use the open released codes of DN4[2] and CovaMNet[3] to get the results. During the test, 600 episodes are randomly selected from the data.

Table IV presents the average accuracies of different models on the novel classes of the fine-grained data sets. Both the one-shot and five-shot classification results are reported. As the table shows, the proposed LRPABN models get steadily and notably improvements on almost all fine-grained data sets for different experimental settings. More detailed, compared with CovaMNet, our proposed models achieve plainly growth performances on CUB Birds, CARS, and NABirds data sets on both one-shot and five-shot setting. Especially for NABirds data, the $LRPABN_{cpt}$ obtains 7.70% and 5.99% gain over CovaMNet for one-shot and five-shot setting, respectively. These results again firmly prove that the proposed pairwise bilinear pooling is superior compared to the self-bilinear pooling operation. Meanwhile, the feature alignment layer further boosts the final performance.

For the comparisons against the DN4 method, from Table IV, LRPABN models obtain the highest accuracy on one-shot setting on CUB Birds, CARS, NABirds data sets, and get second best results on DOGS data, where DN4 performs poorly in one-shot tasks on almost all data sets. For the five-shot setting, DN4 achieves the highest classification accuracy on all four data sets, while $LRPABN_{cpt}$ achieves the second-highest performance on CUB Birds, CARS, and NABirds. We are surprised to observe that in the one-shot-five-way task on the NABirds data, $LRPABN_{cpt}$ gains 15.92% over DN4. Nevertheless, DN4 gets 15.36% boosts over $LRPABN_{cpt}$ under the five-shot-five-way setting on the CARS data set. That is, the proposed LRPABN method holds a tremendous advantage over DN4 for one-shot classification tasks but slightly inferior to DN4 for five-shot classification.

[2]https://github.com/WenbinLee/DN4
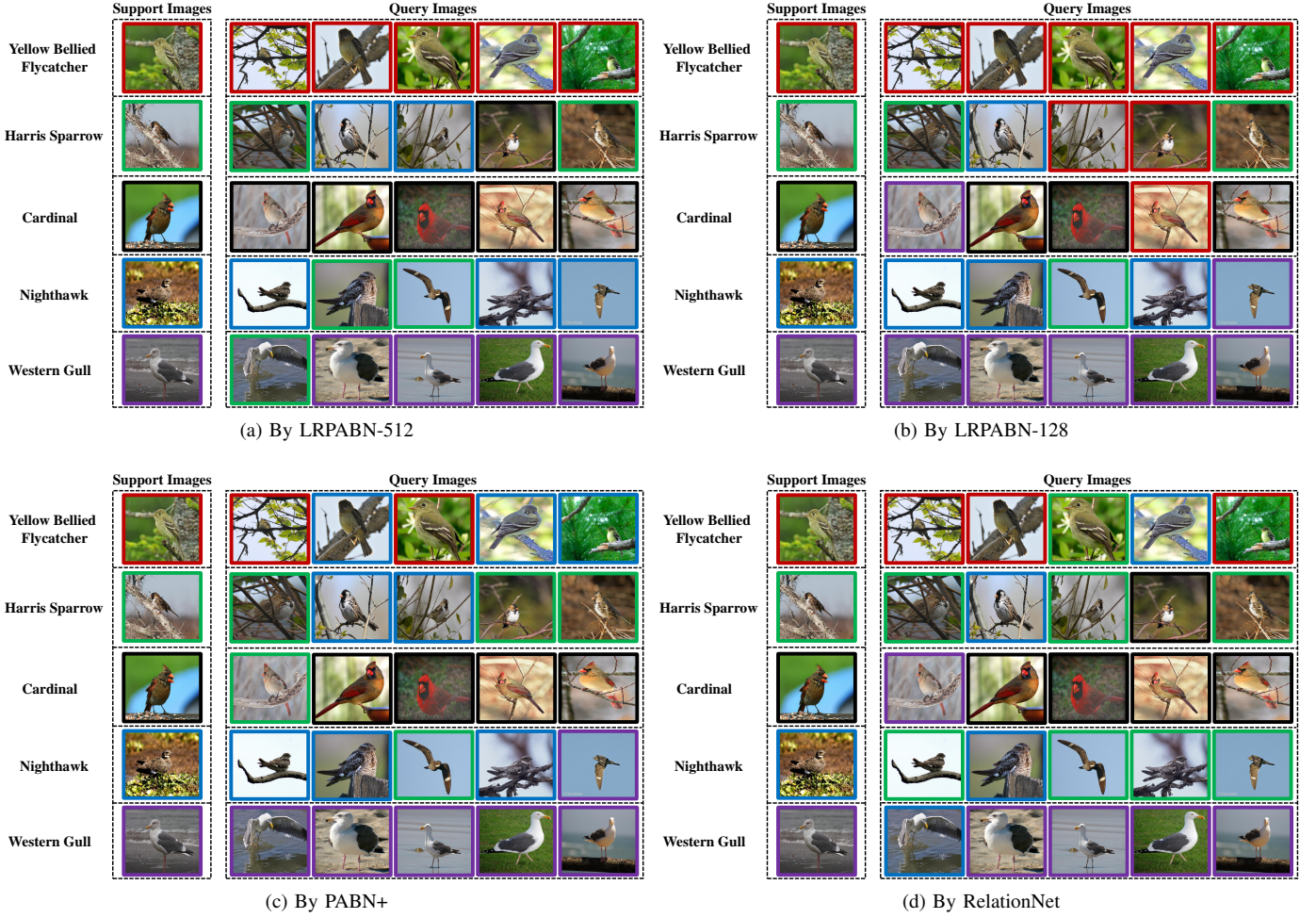[3]https://github.com/WenbinLee/CovaMNet

Fig. 4. Some visual classification results of comparing methods over CUB Birds data set. All the approaches use the same data batch under the five-way-one-shot setting, and for each class, we randomly select five query images as the testing data. We adopt five colors to label the support classes separately. As to the query images, we label the images with the color corresponding to the class label predicted by different models.

The reason for this is that DN4 uses a deep nearest neighbor neural network to search the optimal local features in the support set as the support classes' feature for a given query image. For the target query features (*e.g.,* a set of local features), the algorithm selects the top k nearest local features in the whole support data set according to the cosine similarity between query local features and support local features. That is, the more image in the support classes, the better the class feature will be generated. Thus, for five-shot classification, the DN4 outperforms LRPABN, where under the one-shot setting, DN4 has smaller support features to extract a good representation of the class feature. More importantly, our model is more efficient than DN4. Specifically, under the C-way-K-shot setting, in the inference stage, for each query image, DN4 has $h^2 \times w^2 \times K \times C \times O_{cos}$ computations to predict its label, while LRPABN only needs $h \times w \times C \times O_{comp}$ computations. h and w denote the height and width of the feature map, O_{cos} means the cosine similarity computation used in DN4, and O_{comp} represents the comparator computation in LRPABN. Since $h \times w \times K \times O_{cos} \gg O_{comp}$, DN4 is much slower than LRBPAN during both training and testing, as seen from Table V, DN4 costs 15.20×10^{-3} s

for each query, while LRPABN only needs 2.23×10^{-3} s, which is approximately seven times faster. Moreover, without considering the computation load, our initial low-rank pairwise bilinear model $PABN_{new}$ (Equation (5)) can also achieve the comparable performance against DN4 under both one-shot and five-shot setting, *i.e.,* 78.87% for $PABN_{new}$ compared to 79.64% for DN4 under the five-shot settings. On the other hand, in many practical scenarios, such as endangered species protection, we may only get a one-labeled sample. With higher accuracy under the one-shot setting, our method can achieve more reliable performances compared to DN4 under such circumstances. It indicates the practical value of our models. Considering the proposed LRPABN summing the image features in each category as the class feature, how to generate a good representation of category would further improve the classification performance of our methods.

The classification examples of LRPABN, PABN+, and RelationNet models are shown in Figure 4. We select $LRPABN_{cpt}$ and $PABN+_{cpt}$ as the representative of LRPABN and PABN+ approaches. To investigate the low-rank approximation, we set low-rank comparative feature dimensions as 512 and 128 for LRPABN-512 and LRPABN-128 models separately. By
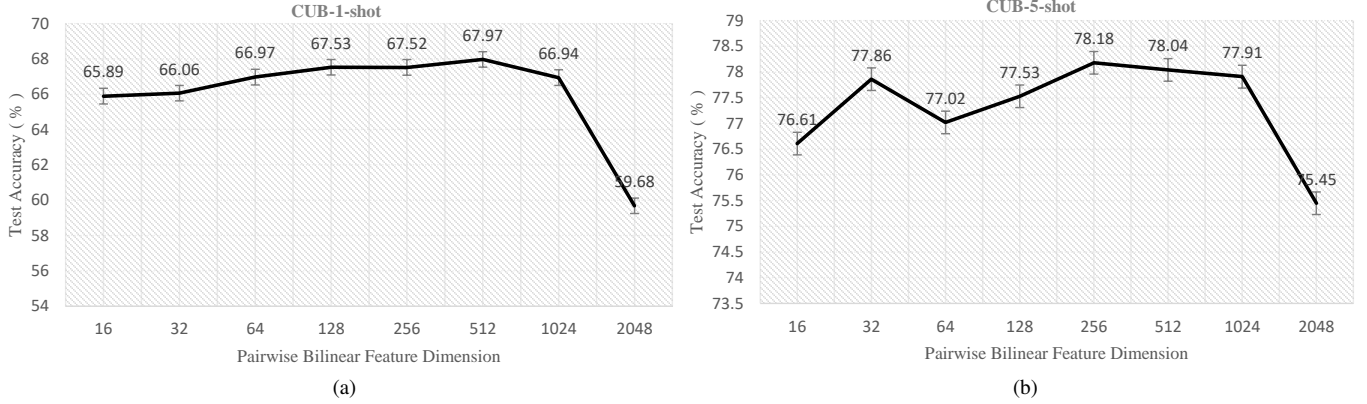
Fig. 5. The pairwise bilinear feature dimension selection experiment. In each sub-figure, the horizontal axis denotes the dimension of the pairwise bilinear feature and the vertical axis represents the test accuracy rate. 5a is the one-shot experiment and 5b is the five-shot experiment on CUB data set.

(a) By RelationNet (b) By LRPABN-Dim-128 (c) By PABN+ (d) By LRPABN-Dim-512
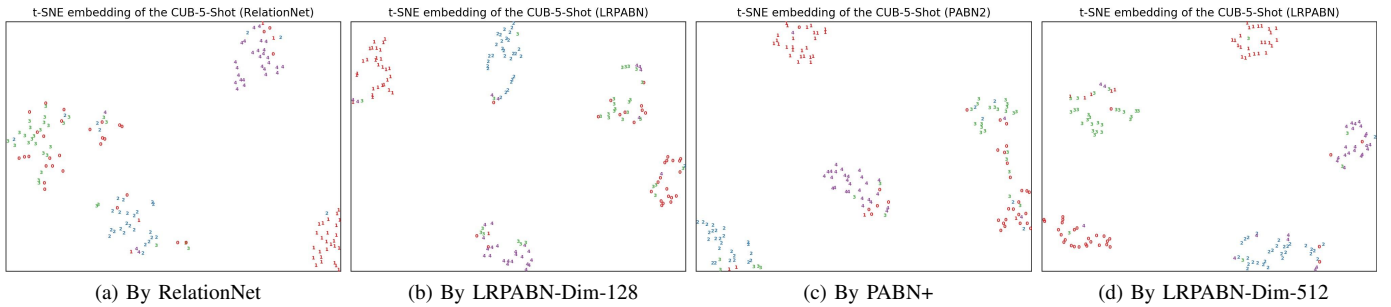
Fig. 6. Visualization of the comparative feature generated by different fusion mechanism in 2D space using t-SNE [62]. Each dot represents a query image that is numeric and marked with different colors according to the real labels. For each class, we randomly select thirty query images to conduct this experiment. The visualization is based on the CUB data set under the 5-way-5-shot setting. (a) shows the result conducted by RelationNet, (b) shows the result conducted by LRPABN$_{cpt}$, and the dimension of the comparative bilinear feature is 128, denoted as LRPABN-Dim-128, (c) shows the result conducted by PABN+$_{cpt}$ model and (d) shows the result conducted by LRPABN$_{cpt}$, and the dimension of the comparative bilinear feature is 512, denoted as LRPABN-Dim-512.

sending a fixed testing batch through the model, which consists of one support sample and five query samples for each of five classes, the prediction of LRPABN-512 only contains six mislabels in the entire 25 queries, while the prediction of LRPABN-128, PABN+ and RelationNet have 7, 8 and 10 wrong labels separately. That validates the effectiveness of the LRPABN models. We also find that in some classes like Nighthawk and Harris Sparrow, the high intra-variance and low inter-variance confuse all the models.

D. Ablation studies

Following the data split used in [1], [23], we conduct several experiments to investigate the different components of the proposed model, the experimental results are shown in Table V. We analysis our methods from various aspects:

Low-Rank Pairwise Bilinear Pooling: First, we replace previous pairwise bilinear pooling (Equation (4)) with Equation (5) as PABN$_{new}$. As seen in Table V, PABN$_{new}$ outperforms PABN$_{cpt}$ on both 1-shot and 5-shot tasks with a lower dimension, which indicates the effectiveness of our proposed initial Low-Rank pairwise pooling (Equation (5)). However, using Equation (5), the model needs to learn a $n \times c \times c$ transformation tensor $\mathcal{W}$ (discussed in Section III-B), which significantly increases the model size and inference time.

Thus, we employ Equation (6) to approximate the transformation tensor as LRPABN. We observe that this approximation achieves superior performance against our previous PABN$_{cpt}$ with a reduced model size as well as a shorter bilinear feature dimension. Specifically, as observed in Table V, the proposed LRPABN costs 2.23 $\times 10^{-3}$ s to identify a query image with a 213K model size, while the previous ICME model PABN$_{cpt}$ requires 8.65 $\times 10^{-3}$ s and 375K parameters. Moreover, the inference time of LRPABN is 2.23×10^{-3} s, while PABN$_{new}$ costs 78.40 $\times 10^{-3}$ s for each query image. That is, our final low-rank pairwise pooling model LRPABN is more advanced than previous PABN models and much more efficient than PABN$_{new}$ model.

Alignment Mechanism: To investigate the effectiveness of the proposed alignment mechanism. We compare PABN$_{cpt}$ and PABN+$_{cpt}$. Besides, we adopt the proposed alignment loss $Align_{loss_2}$ in Equation (9) into LRPABN as LRPABN$_{only\_cpt}$. As seen from Table V, cooperating with the position transform function **T**, PABN+$_{cpt}$ and LRPABN$_{cpt}$ outperform PABN$_{cpt}$ and LRPABN$_{only\_cpt}$, respectively. For instance, under the 5-shot setting, the classification accuracy of PABN+$_{cpt}$ is 77.19% compared to 76.81% of PABN$_{cpt}$.

Input Image Size: It is reported that a higher resolution of the input image can capture a more discriminative feature for Fine-grained classification [8]–[10]. However, few-shot

learning models [15]–[17] usually adopt a low input resolution, e.g., 84 × 84. For a fairness comparison with generic few-shot learning approaches, in Section IV-B, we set the input image size to 84 × 84. To further investigate the affects of input size, we follow [23] to replace the shallow embedding network Conv4 [15]–[17] with AlexNet [59] as LRPABN$_{cpt}$:AlexNet. Moreover, we choose two resolutions for the input images, which are widely used in Fine-grained classification. As Table VI shows, with AlexNet, a higher resolution 448 × 448 brings a significant performance boost compared to lower input size 224 × 224, which validates that a higher input resolution can generate a more subtle comparative feature for FSFG. We also observe that the accuracy of the AlexNet-based methods performs worse than Conv4-based methods. A high input resolution always accompanied by a deep embedding network like AlexNet to extract the informative feature. However, training a deeper embedding network with limited labeled samples is easier to lead the over-fitting problem.

Bilinear Feature Dim: For the feature dimension selection, we change the number of dimensions as 16, 32, 64, 128, 256, 512, 1024, and 2048 for both 1-shot and 5-shot classification tasks on CUB Birds data. The model we used for this experiment is LRPABN$_{cpt}$. The results are shown in Figure 5, it can be observed that as the feature dimension gets large, the test accuracy gradually improves to a peak first, then it goes through a drastic drop. For the 1-shot setting, the performance changes smoothly when the dimension is below 1024. For the 5-shot task, the variation of performance is relatively oscillatory, yet it can grow fast and steadily, with the dimension increasing. Moreover, we find that even with a very compact low-rank approximation (*i.e.,* the dimension is 16), the model can still achieve a decent classification performance, which fatherly verifies the stability of the proposed method. When the dimension goes too large, the model performs poorly, and this may be caused by the increased complexity of the framework can not model the data distribution well with few training samples. As [45] discussed, for self-bilinear features, less than 5% of dimensions are informative. For FSFG, the best feature dimensions for LRPABN are 256 and 512 in the experiments, which are around 5% to 10% of the entire self-bilinear feature dimension.

t-SNE visualization: The t-SNE [62] visualization for different comparative features is presented in Figure 6. We randomly select five support images and thirty query images per category from CUB Birds data to conduct the five-way-five-shot tasks. The original comparative feature dimension of RelationNet is 128 × 3 × 3. We use the convolved feature before the first fully-connected layer in classifier as the final comparative feature with dimension size 576. The comparative feature of PABN+ is 64 × 64 = 4096, and we choose LRPABN$_{cpt}$ with comparative dimension 128 and 512 separately (denoted as LRPABN-Dim-128 and LRPABN-Dim-512) for comparison. As the figure shows, the learned LRPABN-Dim-512 feature, which can be grouped into five classes correctly, outperforms others, the discriminative performance of LRPABN-Dim-128 and PABN+ are similar, which outperform RelationNet' feature. The intuitive visualization results among the above methods again validate the superior capacity of the proposed low-rank pairwise bilinear features for FSFG tasks.

V. Conclusion

In this paper, we propose a novel few-shot fine-grained image classification method, which is inspired by the advanced information processing ability of human beings. The main contribution is the low-rank pairwise bilinear pooling operation, which extracts the second-order comparative features for the pair of support images and query images. Moreover, to get a more precise comparative feature, we propose an effective feature alignment mechanism to match the embedded support image features with query ones. Through comprehensive experiments on four fine-grained datasets, we verify the effectiveness of the proposed method. As the future work, we will investigate more sophisticated alignment mechanisms that applies the feature transformation to support and query images jointly.

References

[1] H. Huang, J. Zhang, J. Zhang, Q. Wu, and J. Xu, "Compare more nuanced: Pairwise alignment bilinear network for few-shot fine-grained learning," in *2019 IEEE International Conference on Multimedia and Expo (ICME)*. IEEE, 2019, pp. 91–96.

[2] C. Wah, S. Branson, P. Welinder, P. Perona, and S. Belongie, "The Caltech-UCSD Birds-200-2011 Dataset," California Institute of Technology, Tech. Rep. CNS-TR-2011-001, 2011.

[3] G. Van Horn, S. Branson, R. Farrell, S. Haber, J. Barry, P. Ipeirotis, P. Perona, and S. Belongie, "Building a bird recognition app and large scale dataset with citizen scientists: The fine print in fine-grained dataset collection," in *CVPR*, June 2015.

[4] A. Khosla, N. Jayadevaprakash, B. Yao, and L. Fei-Fei, "Novel dataset for fine-grained image categorization," in *First Workshop on Fine-Grained Visual Categorization, CVPR*, Colorado Springs, CO, June 2011.

[5] J. Krause, M. Stark, J. Deng, and L. Fei-Fei, "3d object representations for fine-grained categorization," in *4th International IEEE Workshop on 3D Representation and Recognition (3dRR-13)*, Sydney, Australia, 2013.

[6] N. Zhang, J. Donahue, R. Girshick, and T. Darrell, "Part-based r-cnns for fine-grained category detection," in *ECCV*. Springer, 2014, pp. 834–849.

[7] J. Fu, H. Zheng, and T. Mei, "Look closer to see better: Recurrent attention convolutional neural network for fine-grained image recognition," in *CVPR*, July 2017.

[8] T.-Y. Lin, A. RoyChowdhury, and S. Maji, "Bilinear cnn models for fine-grained visual recognition," in *ICCV*, December 2015.

[9] Y. Cui, F. Zhou, J. Wang, X. Liu, Y. Lin, and S. Belongie, "Kernel pooling for convolutional neural networks," in *CVPR*, July 2017.

[10] P. Li, J. Xie, Q. Wang, and Z. Gao, "Towards faster training of global covariance pooling networks by iterative matrix square root normalization," in *CVPR*, June 2018.

[11] J. Krause, H. Jin, J. Yang, and L. Fei-Fei, "Fine-grained recognition without part annotations," in *CVPR*, June 2015.

[12] P. Zhuang, Y. Wang, and Y. Qiao, "Wildfish: A large benchmark for fish recognition in the wild," in *MM*. ACM, 2018, pp. 1301–1309.

[13] G. Van Horn, O. Mac Aodha, Y. Song, Y. Cui, C. Sun, A. Shepard, H. Adam, P. Perona, and S. Belongie, "The inaturalist species classification and detection dataset," in *CVPR*, June 2018.

[14] L. Fei-Fei, R. Fergus, and P. Perona, "One-shot learning of object categories," *IEEE TPAMI*, vol. 28, no. 4, pp. 594–611, 2006.

[15] O. Vinyals, C. Blundell, T. Lillicrap, D. Wierstra *et al.*, "Matching networks for one shot learning," in *NIPS*, 2016, pp. 3630–3638.

[16] J. Snell, K. Swersky, and R. Zemel, "Prototypical networks for few-shot learning," in *NIPS*, 2017, pp. 4077–4087.

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

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

[19] W. Li, J. Xu, J. Huo, L. Wang, G. Yang, and J. Luo, "Distribution consistency based covariance metric networks for few-shot learning," in *AAAI*, 2019.

[20] W. Li, L. Wang, J. Xu, J. Huo, G. Yang, and J. Luo, "Revisiting local descriptor based image-to-class measure for few-shot learning," in *CVPR*, 2019.

[21] A. L. Brown, "The development of memory: Knowing, knowing about knowing, and knowing how to know," in *Advances in child development and behavior*. Elsevier, 1975, vol. 10, pp. 103–152.

[22] D. R. John and C. A. Cole, "Age differences in information processing: Understanding deficits in young and elderly consumers," *Journal of consumer research*, vol. 13, no. 3, pp. 297–315, 1986.

[23] X.-S. Wei, P. Wang, L. Liu, C. Shen, and J. Wu, "Piecewise classifier mappings: Learning fine-grained learners for novel categories with few examples," *IEEE TIP*, 2019.

[24] T.-Y. Lin, A. RoyChowdhury, and S. Maji, "Bilinear convolutional neural networks for fine-grained visual recognition," *IEEE TPAMI*, vol. 40, no. 6, pp. 1309–1322, 2018.

[25] M. Riesenhuber and T. Poggio, "Hierarchical models of object recognition in cortex," *Nature neuroscience*, vol. 2, no. 11, p. 1019, 1999.

[26] Z. Gao, Y. Wu, X. Zhang, J. Dai, Y. Jia, and M. Harandi, "Revisiting bilinear pooling: A coding perspective," in *AAAI*, 2020.

[27] L. Xie, Q. Tian, M. Wang, and B. Zhang, "Spatial pooling of heterogeneous features for image classification," *IEEE TIP*, vol. 23, no. 5, pp. 1994–2008, 2014.

[28] S. Gao, I. W.-H. Tsang, and Y. Ma, "Learning category-specific dictionary and shared dictionary for fine-grained image categorization," *IEEE TIP*, vol. 23, no. 2, pp. 623–634, 2014.

[29] X. Zhang, H. Xiong, W. Zhou, and Q. Tian, "Fused one-vs-all features with semantic alignments for fine-grained visual categorization," *IEEE TIP*, vol. 25, no. 2, pp. 878–892, 2016.

[30] J. Xu, V. Jagadeesh, and B. Manjunath, "Multi-label learning with fused multimodal bi-relational graph," *IEEE TMM*, vol. 16, no. 2, pp. 403–412, 2014.

[31] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *IEEE CVPR*, 2016, pp. 770–778.

[32] Y. Gal and Z. Ghahramani, "Dropout as a bayesian approximation: Representing model uncertainty in deep learning," in *ICML*, 2016, pp. 1050–1059.

[33] Y. Yao, F. Shen, J. Zhang, L. Liu, Z. Tang, and L. Shao, "Discovering and distinguishing multiple visual senses for web learning," *IEEE TMM*, 2018.

[34] J. Zhang, Q. Wu, C. Shen, J. Zhang, and J. Lu, "Multilabel image classification with regional latent semantic dependencies," *IEEE TMM*, vol. 20, no. 10, pp. 2801–2813, 2018.

[35] S. Qiao, C. Liu, W. Shen, and A. L. Yuille, "Few-shot image recognition by predicting parameters from activations," in *IEEE CVPR*, 2018, pp. 7229–7238.

[36] J. Zhang, Q. Wu, C. Shen, J. Zhang, and J. Lu, "Mind your neighbours: Image annotation with metadata neighbourhood graph co-attention networks," *IEEE CVPR*, 2018.

[37] C. Huang, H. Li, Y. Xie, Q. Wu, and B. Luo, "Pbc: Polygon-based classifier for fine-grained categorization," *IEEE TMM*, vol. 19, no. 4, pp. 673–684, 2016.

[38] Z. Xu, D. Tao, S. Huang, and Y. Zhang, "Friend or foe: Fine-grained categorization with weak supervision," *IEEE TIP*, vol. 26, no. 1, pp. 135–146, 2017.

[39] Y. Zhang, X.-S. Wei, J. Wu, J. Cai, J. Lu, V.-A. Nguyen, and M. N. Do, "Weakly supervised fine-grained categorization with part-based image representation," *IEEE TIP*, vol. 25, no. 4, pp. 1713–1725, 2016.

[40] B. Zhao, X. Wu, J. Feng, Q. Peng, and S. Yan, "Diversified visual attention networks for fine-grained object classification," *IEEE TMM*, vol. 19, no. 6, pp. 1245–1256, 2017.

[41] H. Yao, S. Zhang, Y. Zhang, J. Li, and Q. Tian, "Coarse-to-fine description for fine-grained visual categorization," *IEEE TIP*, vol. 25, no. 10, pp. 4858–4872, 2016.

[42] Y. Peng, X. He, and J. Zhao, "Object-part attention model for fine-grained image classification," *IEEE TIP*, vol. 27, no. 3, pp. 1487–1500, 2018.

[43] L. Zhang, Y. Yang, M. Wang, R. Hong, L. Nie, and X. Li, "Detecting densely distributed graph patterns for fine-grained image categorization," *IEEE TIP*, vol. 25, no. 2, pp. 553–565, 2016.

[44] A. Iscen, G. Tolias, P.-H. Gosselin, and H. Jégou, "A comparison of dense region detectors for image search and fine-grained classification," *IEEE TIP*, vol. 24, no. 8, pp. 2369–2381, 2015.

[45] Y. Gao, O. Beijbom, N. Zhang, and T. Darrell, "Compact bilinear pooling," in *CVPR*, June 2016.

[46] S. Kong and C. Fowlkes, "Low-rank bilinear pooling for fine-grained classification," in *CVPR*, July 2017.

[47] Y. Suh, J. Wang, S. Tang, T. Mei, and K. Mu Lee, "Part-aligned bilinear representations for person re-identification," in *ECCV*, September 2018.

[48] C. Yu, X. Zhao, Q. Zheng, P. Zhang, and X. You, "Hierarchical bilinear pooling for fine-grained visual recognition," in *ECCV*, September 2018.

[49] R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation," in *CVPR*, June 2014.

[50] N. Pham and R. Pagh, "Fast and scalable polynomial kernels via explicit feature maps," in *ACM SIGKDD*. ACM, 2013, pp. 239–247.

[51] J.-H. Kim, K. W. On, W. Lim, J. Kim, J.-W. Ha, and B.-T. Zhang, "Hadamard Product for Low-rank Bilinear Pooling," in *ICLR*, 2017.

[52] C. Finn, P. Abbeel, and S. Levine, "Model-agnostic meta-learning for fast adaptation of deep networks," in *ICML*, vol. 70. International Convention Centre, Sydney, Australia: PMLR, 06–11 Aug 2017, pp. 1126–1135.

[53] R. Sachin and L. Hugo, "Optimization as a model for few-shot learning," in *ICLR*, 2017.

[54] J. Schmidhuber, "Evolutionary principles in self-referential learning, or on learning how to learn: The meta-meta-... hook," Diplomarbeit, Technische Universitt Mnchen, Mnchen, 1987.

[55] S. Thrun and L. Pratt, Eds., *Learning to Learn*. Norwell, MA, USA: Kluwer Academic Publishers, 1998.

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

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

[58] F. Pahde, P. Jähnichen, T. Klein, and M. Nabi, "Cross-modal hallucination for few-shot fine-grained recognition," *arXiv preprint arXiv:1806.05147*, 2018.

[59] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *NIPS*, 2012.

[60] C. R. Qi, H. Su, K. Mo, and L. J. Guibas, "Pointnet: Deep learning on point sets for 3d classification and segmentation," in *CVPR*, July 2017.

[61] H. Yao, S. Zhang, Y. Zhang, J. Li, and Q. Tian, "One-shot fine-grained instance retrieval," in *MM*. ACM, 2017, pp. 342–350.

[62] L. v. d. Maaten and G. Hinton, "Visualizing data using t-sne," *JMLR*, vol. 9, no. Nov, pp. 2579–2605, 2008.