# Rise by Lifting Others: Interacting Features to Uplift Few-Shot Fine-Grained Classification

Xiaoxu Li, Peiyu Lu, Rui Zhu, Zhanyu Ma, *Senior Member, IEEE*, Jie Cao, Jing-Hao Xue, *Senior Member, IEEE*

***Abstract*—Few-shot fine-grained classification entails notorious subtle inter-class variation. Recent works address this challenge by developing attention mechanisms, such as the task discrepancy maximization (TDM) that can highlight discriminative channels. This paper, however, aims to reveal that, besides designing sophisticated attention modules, a well-designed input scheme, which simply blends two types of features and their interactions capturing different properties of the target object, can also greatly promote the quality of the learnt weights. To illustrate, we design a bi-feature interactive TDM (BiFI-TDM) module to serve as a strong foundation for TDM to discover the most discriminative channels with ease. Specifically, we design a novel mixing strategy to produce four sets of channel weights with different focuses, reflecting the properties of the corresponding input features and their interactions, as well as a proper feature re-weighting scheme. Extensive experiments on four benchmark fine-grained image datasets showcase superior performance of BiFI-TDM in metric-based few-shot methods. Our codes are available at https://github.com/Peiy-Lu/BiFI-TDM.**

***Index Terms*—Few-shot classification, Fine-grained classification, Channel attention, Feature interaction.**

## I. Introduction

DEEP neural networks have shown impressive performances in image classification tasks, but they often require training on a massive number of labelled images. However, image annotation is often expensive. This leads to the challenging task of few-shot image classification [1], aiming to learn well generalisable features, from only a few labelled training images, that can adapt to classify unseen test classes.

In this work, we focus on an even more challenging task than ordinary few-shot learning: fine-grained image classification in the few-shot setting [2]–[5], the aim of which is to classify subcategories with quite similar appearance details but with only a few labelled training samples available. De facto few-shot classifiers, metric-based few-shot methods, rely on a predefined metric [6], [7] or a metric module [8], [9] and usually have limited performance on fine-grained tasks, because they fail to take now subtle inter-class variation into consideration. As remedies to this issue, attention mechanisms [10]–[12] have been involved to boost the discriminative power of the learnt features by assigning higher weights to more discriminative spatial areas or channels.

Corresponding author: Rui Zhu (email: rui.zhu@city.ac.uk)

X. Li, P. Lu, J. Cao are with the School of Computer and Communication, Lanzhou University of Technology, Lanzhou 730050, China.

R. Zhu is with the Faculty of Actuarial Science and Insurance, Bayes Business School, City St George's, University of London EC1Y 8TZ, UK.

Z. Ma is with the Pattern Recognition and Intelligent System Laboratory, School of Artificial Intelligence, Beijing University of Posts and Telecommunications, Beijing 100876, China.

J.-H. Xue is with the Department of Statistical Science, University College London, London WC1E 6BT, U.K.

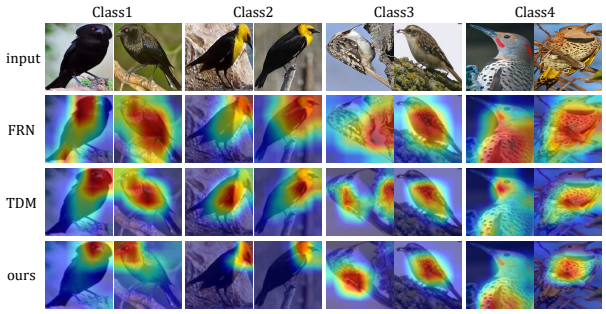



Fig. 1. From top to bottom: input images (two from each class) and their discriminative features identified by FRN, FRN+TDM (shortened as 'TDM') and FRN+BiFI-TDM ('ours') for four bird species in the CUB dataset. Our method can focus on delicate and consistent regions while being less affected by irrelevant objects and background.

In this paper, however, we would like to postulate that, besides developing sophisticated attention modules, a well-designed input scheme to these modules allowing enriched features to interact can also greatly promote the quality of the learnt weights. To demonstrate this idea, here we adopt the task discrepancy maximization (TDM) module [12].

In TDM, task-specific channel weights are linear mixtures of the support weights and the query weights calculated from the support attention module (SAM) and the query attention module (QAM), respectively. In Fig. 1, we visualise the discriminative features of four bird species in the CUB dataset [13] captured by incorporating the TDM module in feature reconstruction network (FRN) [14], a strong metric-based few-shot method that reconstructs the query image via the pooled support features through ridge regression. Clearly, FRN itself tends to identify the whole bird objects as discriminative and sometimes involves the noisy background. Incorporating TDM to FRN (FRN+TDM) can provide apparent improvement to the discriminative regions, which concentrate more on the birds' body parts while being less affected by the background. TDM, nonetheless, adopts simple base features with no emphasis on specific patterns of the target object, making searching for discriminative channels a tough task. It is still noticeable that the TDM features for the same species are not consistent. For example, in class 1, when the birds are facing different directions, the TDM features focus on the head for the first bird while on the body for the second bird. The same pattern can also be observed in class 4. Moreover, in class 3, the wing of the first bird has similar appearance to

the wood, but TDM cannot distinguish them and identify both as important features.

To address the above issues and make exploration of discriminative channels smoother, we propose to design a simple scheme: simply blend two types of features and their interactions capturing different properties of the target object as the input to TDM. We shall showcase that the two types of features and their interactions can serve as a strong foundation with plentiful object-related information for TDM to discover the most discriminative channels with ease.

To this end, we design a novel bi-feature interactive TDM (BiFI-TDM) module to rigorously fuse the two types of features. Specifically, we first feed the two types of features to SAM and QAM separately and obtain two sets of support and query weights, which are then carefully mixed to generate four sets of channel weights with different emphases: two of them are feature-specific while the other two reflect the interaction effects. Next, four sets of re-weighted support and query features are obtained based on the channel weights, and they are input to the metric-based method for final classification. By fusing these two types of features in BiFI-TDM, we shall demonstrate that significantly distinct patterns of objects can be identified, consistency within the same class. In this paper, we adopt two independent backbones for base feature extraction. To ensure that the features extracted by each backbone are distinct, we introduce an orthogonal loss to reduce the similarity between them. However, to prevent situations where one backbone learns only foreground features and the other learns only background features due to the orthogonal constraint, we also incorporate additional anchor losses. This approach ensures that both backbones acquire discriminative knowledge for classification.

By introducing an orthogonal loss, we generate two distinct types of features. As shown in the last row of Fig. 1, by utilising BiFI-TDM, the aforementioned problems in TDM are resolved. In classes 1 and 4, the important regions are consistent within the same class and also more delicate and focused on finer body parts. Additionally, in class 3, the wood is excluded from the highlighted areas to classify the bird. Last but not least, we also would like to note that surely other types of features, which can capture useful characteristics of objects, should also be workable through our proposed scheme by interested researchers.

To sum up, our contributions are four-fold:

- We reveal that, besides designing sophisticated attention mechanisms, a well-designed input scheme allowing enriched features to interact can also greatly promote the quality of the learnt weights.
- We propose the BiFI-TDM module to generate channel weights that can highlight delicate and consistent areas to distinguish fine-grained image categories. The novel mixing mechanism in BiFI-TDM can rigorously fuse two types of features capturing different properties of objects. Four sets of channel weights, output from BiFI-TDM with different focuses, can clearly reflect various properties of the input features and their interactions.
- We propose a feature weighting scheme to properly re-weight the two types of features with the four sets of channel weights obtained from BiFI-TDM.
- We showcase the superior classification performance of utilising BiFI-TDM in metric-based methods on four benchmark fine-grained image datasets. We also demonstrate the effectiveness of the components in BiFI-TDM via extensive ablation studies.

The rest of this paper is organised as follows. In section II, we discuss the closely related work. We then introduce the technical details of BiFI-TDM in section III. Extensive experimental results to demonstrate the effectiveness of BiFI-TDM are reported in section IV. Finally, we draw concluding remarks in section V.

## II. Related Work

### A. Metric-based Few-shot Image Classification

Metric-based few-shot methods adopts a metric function to measure the similarity between a query image and the support classes. For example, MatchingNet [6] uses the cosine similarities between images, while ProtoNet [7] is based on the Euclidean distance between the query image and class prototypes, i.e. the averages of the support features from each class. Rather than pre-defined metrics, learnable metric modules have also been introduced, such as the relation metric module [15], graph neural networks [16], [17] and bi-similarity network [4]. To prevent unreliable predictions from a sophisticated metric, BlockMix [18] utilises the interpolation of the images and labels in metric learning. KSTNet [19] enhances few-shot learning by incorporating auxiliary prior knowledge. It uses cosine similarity and contrastive loss optimisation to train visual classifiers. Different from previous work, FRN adopts the image reconstruction error as the metric. It reconstructs the query image as the weighted average of the pool of support features of each class, which can well keep the spatial information within images.

Our BiFI-TDM can be attached to any metric-based few-shot models.

### B. Attention Mechanisms for Fine-grained Few-shot Image Classification

To improve the discriminative abilities of the learnt features for fine-grained classification, attention mechanisms have been involved. For example, Tang et al. [11] propose a multi-level attention pyramid to extract features dominated by the target object with less emphases on backgrounds. Xu et al. [10] develop a dual-branch network with the hard attention capturing deep features related to fine-grained object parts and the soft attention consisting of complementary features from original activations. Kang et al. [15] propose the cross-correlational attention that can weigh the spatial regions to match the target object across support and query images. Besides re-weighting spatial regions in feature maps, Lee et al. [12] design the TDM module to highlight discriminative channels via task-specific channel weights.

Rather than designing new attention mechanisms, we propose a simple yet effective input scheme and weight mixing mechanism to promote the quality of the learnt weights with a demonstration in TDM.

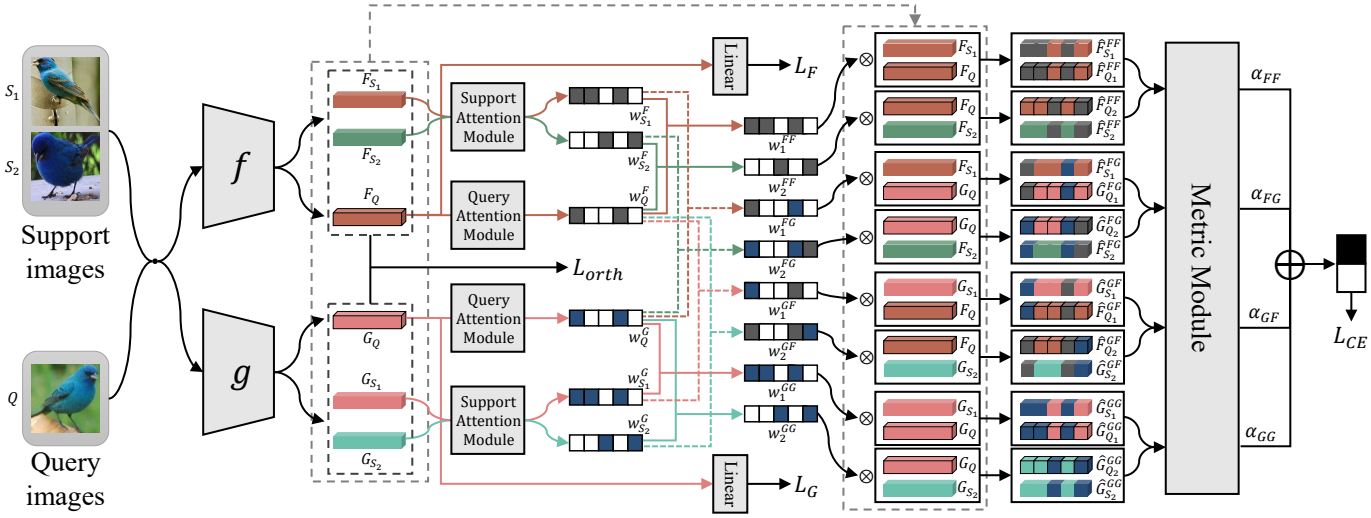



Fig. 2. The architecture overview of using BiFI-TDM in a metric-based few-shot method with an example in 2-way 1-shot setting. Support images and query images are first passed through two feature extractors to obtain two types of features $\mathbf{F}$ and $\mathbf{G}$. The support features are input to two SAMs with shared parameters to obtain $\mathbf{w}_S^F$ and $\mathbf{w}_S^G$, while the query features are input to two QAMs with shared parameters to get $\mathbf{w}_Q^F$ and $\mathbf{w}_Q^G$. Next, the four weights are mixed to generate eight task-specific channel weights, four for each class. Subsequently, they are applied to re-weight features to obtain feature maps focusing on discriminative regions. Finally, a metric module is adopted to calculate four metrics between the query image and the support classes. The weighted average of the metrics is used as the final score for classification. Besides the cross-entropy loss $L_{CE}$, we propose to involve two additional losses $L_F$ and $L_G$ to enhance the discriminative power of the query features. Additionally, we employ an orthogonal loss, $L_{\text{orth}}$, to enforce $\mathbf{F}$ and $\mathbf{G}$ remain distinct.

## III. Method

In this section, we discuss the technical details of BiFI-TDM. We first introduce the preliminaries of few-shot image classification in section III-A and then summarise the workflow of BiFI-TDM in section III-B. The process of calculating the four sets of channel weights is detailed in section III-C and the feature weighting scheme is introduced in section III-D. Finally, the training loss to supervise the model is discussed in section III-E.

### A. Preliminaries

In this paper, we follow the episodic training strategy with the $N$-way $K$-shot setting for few-shot image classification [6]. We randomly divide the data into a training set $\mathcal{D}_{\text{train}}$, a validation set $\mathcal{D}_{\text{val}}$ and a test set $\mathcal{D}_{\text{test}}$, with mutually exclusive sets of classes. We randomly partition each subset into multiple episodes, with each episode consisting of a support set $\mathcal{S} = \{(\mathbf{x}_i, y_i)\}_{i=1}^{N\times K}$ and a query set $\mathcal{Q} = \{(\mathbf{x}_q, y_q)\}_{q=1}^{N\times l}$, where $\mathbf{x}$ denotes the image, $y$ is the corresponding label, $N$ represents the number of classes, $K$ is the number of support samples per class and $l$ is the number of query samples per class. During each training episode, the model is updated to minimise the loss on the query set. The best model is selected by using the validation set. The classification performance of the chosen model is evaluated as the average classification accuracy on the test set.

### B. Architecture Overview

We illustrate the workflow of incorporating BiFI-TDM in a metric-based method in Fig. 2 with an example of 2-way 1-shot setting. The support and query images are fed to two separate feature extractors, $f$ and $g$, respectively, to generate two types of features, $\mathbf{F}$ and $\mathbf{G}$, that can capture different characteristics of objects to serve as a rich source of information. Hereafter, we use subscripts $S$ and $Q$ to denote quantities related to the support and query sets, respectively. In Fig. 2, the numerical subscripts of 1 and 2 are indexes for support classes. We omit the class indexes in this section for a more concise explanation. The support features, $\mathbf{F}_S$ and $\mathbf{G}_S$, are input to two SAMs with shared parameters to obtain support channel weights $\mathbf{w}_S^F$ and $\mathbf{w}_S^G$, while the query features, $\mathbf{F}_Q$ and $\mathbf{G}_Q$, are input to two QAMs with shared parameters to obtain query channel weights $\mathbf{w}_Q^F$ and $\mathbf{w}_Q^G$. Next, the support and query weights are mixed in the following four combinations to generate task-specific channel weights: 1) $\mathbf{w}_S^F$ and $\mathbf{w}_Q^F$, 2) $\mathbf{w}_S^G$ and $\mathbf{w}_Q^G$, 3) $\mathbf{w}_S^F$ and $\mathbf{w}_Q^G$ and 4) $\mathbf{w}_S^G$ and $\mathbf{w}_Q^F$, where the first two capture the feature-specific information while the last two explore the interactions between $\mathbf{F}$ and $\mathbf{G}$. Note that these combinations are class sensitive, since the channel weights are per task. Thus for the $N$-way setting, we could obtain $4N$ such combinations, and in Fig. 2 we have eight combinations for the 2-way example. These task-specific channel weights are then applied to the corresponding features to produce the weighted versions; for example, the weights mixed from $\mathbf{w}_S^F$ and $\mathbf{w}_Q^G$ are applied to weigh $\mathbf{F}_S$ and $\mathbf{G}_Q$. Finally, the weighted features are used in a metric-based method or the metric module in Fig. 2 to calculate four metrics whose weighted average is adopted as the score for classification. To train the network, we adopt the cross-entropy loss $L_{CE}$ to measure the loss of classification on the query set. We also propose to involve two additional losses $L_F$ and $L_G$ to enhance the discriminative power of the query features extracted from $f$ and $g$.

### C. BiFI-TDM

BiFI-TDM aims to enrich the information pool for TDM to identify discriminative channels. To achieve this, we input two types of feature representations, $\mathbf{F}_S$ and $\mathbf{F}_Q$ from the feature extractor $f$ and $\mathbf{G}_S$ and $\mathbf{G}_Q$ from the feature extractor $g$. BiFI-TDM contains two SAMs and two QAMs from TDM to calculate the channel weights that can stress the inter-class differences in the support set and the objective-related information in the query set, respectively. Specifically, SAM tries to make the class prototype more compact via obtaining a smaller channel-wise intra-class score $\mathbf{r}_n^{\text{intra}} \in \mathbb{R}^C$, while push the prototypes of different classes further apart via a larger channel-wise inter-class score $\mathbf{r}_n^{\text{inter}} \in \mathbb{R}^C$, where $C$ is the number of channels. The $c$th value of $\mathbf{r}_n^{\text{intra}}$ is calculated as

$$r_{n,c}^{\text{intra}} = \frac{1}{HW}||\mathbf{P}_{n,c} - \bar{\mathbf{P}}_n||_2^2,$$

where $\mathbf{P}_n$ is the prototype of the $n$th class, $\mathbf{P}_{n,c}$ is the $c$th channel of $\mathbf{P}_n$, $\bar{\mathbf{P}}_n$ is the average of all $C$ channels of $\mathbf{P}_n$, $||\cdot||_2$ is the Frobenius norm, and $H$ and $W$ are the height and weight of the feature map, respectively. The $c$th value of $\mathbf{r}_n^{\text{inter}}$ is calculated as

$$r_{n,c}^{\text{inter}} = \frac{1}{HW} \min_{n' \in [1,N], n' \neq n} ||\mathbf{P}_{n,c} - \bar{\mathbf{P}}_{n'}||_2^2,$$

which measures the minimum distance from the prototype of the $n$th class to the mean of prototype of a different class. SAM then outputs the channel weights as the weighted average of $\mathbf{r}_n^{\text{intra}}$ and $\mathbf{r}_n^{\text{inter}}$:

$$\begin{aligned}\mathbf{w}_{S_n} &= uh(\mathbf{r}_n^{\text{intra}}) + (1-u)h(\mathbf{r}_n^{\text{inter}}) \\ &= u\mathbf{w}_{S_n}^{\text{intra}} + (1-u)\mathbf{w}_{S_n}^{\text{inter}},\end{aligned} \tag{1}$$

where $u \in [0,1]$ is a weighting scalar and $h$ is a fully-connected block and we follow the same structure as in TDM. Following the above procedure and feed $\mathbf{F}_S$ and $\mathbf{G}_S$ to two SAMs with shared parameters, we can obtain two channel weights for the $n$th class, $\mathbf{w}_{S_n}^t$ with $t \in \{F, G\}$.

In QAM, since there is no label information of query images during test phase, only $\mathbf{r}_Q^{\text{intra}} \in \mathbb{R}^C$ can be obtained:

$$r_{Q,c}^{\text{intra}} = \frac{1}{HW}||\mathbf{F}_{Q,c} - \bar{\mathbf{F}}_Q||_2^2,$$

where $\mathbf{F}_{Q,c}$ is the $c$th channel of the query feature and $\bar{\mathbf{F}}_Q$ is the mean of all channels of $\mathbf{F}_Q$. Note that here we use $\mathbf{F}_Q$ as an example. The calculation of using $\mathbf{G}_Q$ is the same by replacing $\mathbf{F}_{Q,c}$ with $\mathbf{G}_{Q,c}$ and $\bar{\mathbf{F}}_Q$ with $\bar{\mathbf{G}}_Q$ in the above equation. The channel weights produced by QAM is

$$\mathbf{w}_Q = h(\mathbf{r}_Q^{\text{intra}}). \tag{2}$$

By feeding $\mathbf{F}_Q$ and $\mathbf{G}_Q$ to two QAMs with shared parameters, we also obtain two channel weights for the query image, $\mathbf{w}_Q^t$ with $t \in \{F, G\}$, respectively.

Finally, we calculate the task-specific channel weights $\mathbf{w}_n$ by mixing the weights obtained from the support and query sets:

$$\mathbf{w}_n^k = v\mathbf{w}_{S_n}^t + (1-v)\mathbf{w}_Q^{t'}, \tag{3}$$

where $v \in [0,1]$ is a weight parameter. With $t$ and $t'$ in $\{F, G\}$, we generate four linear mixtures of support and query weights; that is, $k \in \{FF, FG, GF, GG\}$ with the first letter denoting the value of $t$ for the support weight while the second letter denoting the value of $t'$ for the query weight. In these four weights, $\mathbf{w}_n^{FF}$ and $\mathbf{w}_n^{GG}$ exploit the information within each specific feature while $\mathbf{w}_n^{FG}$ and $\mathbf{w}_n^{GF}$ reflect the interactions between the two types of features.

### D. Feature Weighting Scheme

TABLE I
THE FEATURE WEIGHTING SCHEME OF BIFI-TDM. THE NOTATIONS WITH HATS ARE THE RE-WEIGHTED FEATURES CALCULATED BY THE WEIGHTS AND THE FEATURES IN THE CORRESPONDING COLUMNS AND ROWS, RESPECTIVELY.

| | $\mathbf{w}_n^{FF}$ | $\mathbf{w}_n^{FG}$ | $\mathbf{w}_n^{GF}$ | $\mathbf{w}_n^{GG}$ |
|---|---|---|---|---|
| $\mathbf{F}_{S_n}$ | $\hat{\mathbf{F}}_{Sn}^{FF}$ | $\hat{\mathbf{F}}_{Sn}^{FG}$ | - | - |
| $\mathbf{F}_Q$ | $\hat{\mathbf{F}}_{Q_n}^{FF}$ | - | $\hat{\mathbf{F}}_{Q_n}^{GF}$ | - |
| $\mathbf{G}_Q$ | - | $\hat{\mathbf{G}}_{Q_n}^{FG}$ | - | $\hat{\mathbf{G}}_{Q_n}^{GG}$ |
| $\mathbf{G}_{S_n}$ | - | - | $\hat{\mathbf{G}}_{Sn}^{GF}$ | $\hat{\mathbf{G}}_{Sn}^{GG}$ |

The task-specific weights are then applied to the corresponding features to highlight discriminative channels. The weights produced by the $\mathbf{F}$ ($\mathbf{G}$) support features are applied to weigh the $\mathbf{F}$ ($\mathbf{G}$) support features:

$$\hat{\mathbf{F}}_{S_n}^{k_1} = \mathbf{F}_{S_n} \odot (\mathbf{1}_{(H\times W)} \times \mathbf{w}_n^{k_1}), \tag{4}$$

$$\hat{\mathbf{G}}_{S_n}^{k_2} = \mathbf{G}_{S_n} \odot (\mathbf{1}_{(H\times W)} \times \mathbf{w}_n^{k_2}), \tag{5}$$

where $k_1 \in \{FF, FG\}$, $k_2 \in \{GG, GF\}$ and $\mathbf{1}_{H\times W}$ is a vector with $H \times W$ ones. Similarly, the weights produced by the $\mathbf{F}$ ($\mathbf{G}$) query features are applied to weight the $\mathbf{F}$ ($\mathbf{G}$) query features:

$$\hat{\mathbf{F}}_{Q_n}^{k_3} = \mathbf{F}_Q \odot (\mathbf{1}_{(H\times W)} \times \mathbf{w}_n^{k_3}), \tag{6}$$

$$\hat{\mathbf{G}}_{Q_n}^{k_4} = \mathbf{G}_Q \odot (\mathbf{1}_{(H\times W)} \times \mathbf{w}_n^{k_4}), \tag{7}$$

where $k_3 \in \{FF, GF\}$ and $k_4 \in \{GG, FG\}$. This feature weighting scheme is presented in Table I.

Given the four sets of re-weighted support and query features, we can obtain four metrics based on a metric-based algorithm, such as FRN or ProtoNet. Then, the dissimilarity between the query image $\mathbf{x}_q$ and the $n$th support class is measured by the weighted average of the four metrics:

$$\begin{aligned}d_n =& \alpha^{FF} d(\hat{\mathbf{F}}_{Q_n}^{FF}, \hat{\mathbf{F}}_{S_n}^{FF}) + \alpha^{FG} d(\hat{\mathbf{G}}_{Q_n}^{FG}, \hat{\mathbf{F}}_{S_n}^{FG}) \\ &+ \alpha^{GF} d(\hat{\mathbf{F}}_{Q_n}^{GF}, \hat{\mathbf{G}}_{S_n}^{GF}) + \alpha^{GG} d(\hat{\mathbf{G}}_{Q_n}^{GG}, \hat{\mathbf{G}}_{S_n}^{GG}),\end{aligned} \tag{8}$$

where $\alpha$'s are learnable parameters optimised during the training process. Clearly, the middle two terms are associated with the feature interactions.

Based on the metric calculated in (8), we assign a test query image to the class with the highest posterior probability

$$P(\hat{y}_q = n|\mathbf{x}_q) = \frac{e^{-\beta d_n}}{\sum_{n' \in [1,N]} e^{-\beta d_{n'}}}, \tag{9}$$

where $\beta$ is a learnable temperature factor for the softmax function.

TABLE II
THE 5-WAY FEW-SHOT CLASSIFICATION ACCURACIES ON FOUR BENCHMARK DATASETS FOR THE RESNET-12 BACKBONE. †REPRESENTS RESULTS REPRODUCED BY OURSELVES. GREEN VALUES INDICATE THE PERFORMANCE IMPROVEMENT OF OUR METHOD RELATIVE TO TDM.

| Method | CUB | | Dogs | | Cars | | Flowers | |
|---|---|---|---|---|---|---|---|---|
| | 1-shot | 5-shot | 1-shot | 5-shot | 1-shot | 5-shot | 1-shot | 5-shot |
| MatchingNet [6]† | 71.87±0.85 | 85.08±0.57 | 66.48±0.88 | 79.57±0.63 | 73.32±0.93 | 87.61±0.55 | 75.70±0.88 | 87.61±0.55 |
| Baseline++ [20] | 64.62±0.98 | 81.15±0.61 | 56.59±0.51 | 77.96±0.70 | 67.92±0.92 | 84.17±0.58 | 69.03±0.92 | 85.72±0.63 |
| DeepEMD [21]† | 71.11±0.44 | 86.30±0.19 | 67.59±0.30 | 81.13±0.20 | 73.30±0.29 | 88.37±0.17 | 70.00±0.35 | 83.63±0.26 |
| RENet [15]† | 79.60±0.44 | 91.51±0.24 | 71.53±0.48 | 85.92±0.30 | 85.54±0.37 | 94.31±0.17 | 78.41±0.48 | 89.42±0.26 |
| MCL [22]† | 83.25±0.25 | 93.01±0.16 | 71.49±0.28 | 85.24±0.23 | 85.04±0.36 | 93.92±0.21 | 76.55±0.26 | 90.31±0.19 |
| FicNet [23] | 80.97±0.57 | 93.17±0.32 | 72.41±0.64 | 85.11±0.37 | 86.81±0.47 | 95.36±0.22 | - | - |
| LCCRN [24]† | 82.99±0.19 | 93.52±0.10 | 75.95±0.20 | 88.55±0.12 | 87.27±0.17 | 96.45±0.06 | 84.12±0.18 | 94.77±0.09 |
| BSFA [25]† | 83.11±0.41 | 93.08±0.23 | 73.54±0.50 | 85.70±0.33 | 88.78±0.38 | 95.31±0.20 | 75.33±0.54 | 86.90±0.36 |
| BiFRN [26]† | 82.90±0.19 | 93.11±0.10 | 74.73±0.21 | 87.76±0.12 | 87.80±0.16 | 96.49±0.06 | 80.30±0.20 | 92.30±0.11 |
| AIS-MLI [27] | **85.60±0.41** | 93.36±0.21 | 76.32±0.47 | 88.25±0.27 | 89.09±0.36 | 97.08±0.14 | 79.85±0.48 | 90.59±0.29 |
| C2-Net [28]† | 83.34±0.42 | 92.20±0.23 | 75.50±0.49 | 87.65±0.28 | 84.81±0.42 | 92.61±0.23 | 80.86±0.46 | 91.54±0.27 |
| ProtoNet [7]† | 78.98±0.21 | 90.61±0.11 | 70.36±0.22 | 86.54±0.13 | 82.29±0.20 | 93.11±0.10 | 75.41±0.22 | 89.46±0.14 |
| +TDM [12]† | 80.36±0.21 | 91.19±0.11 | 70.41±0.22 | 87.27±0.13 | 86.64±0.18 | 95.23±0.08 | 76.65±0.22 | 91.25±0.12 |
| +BiFI-TDM | 81.05±0.20 +0.69 | 91.33±0.11 +0.14 | 74.19±0.22 +3.78 | 87.46±0.12 +0.19 | 87.78±0.17 +1.14 | 95.45±0.08 +0.22 | 78.26±0.21 +1.61 | 91.36±0.12 +0.11 |
| FRN [14]† | 82.95±0.19 | 92.38±0.10 | 76.22±0.21 | 87.95±0.12 | 87.70±0.17 | 95.42±0.08 | 80.65±0.21 | 92.02±0.12 |
| +TDM [12]† | 82.41±0.19 | 92.37±0.10 | 76.11±0.20 | 88.45±0.11 | 87.21±0.17 | 96.11±0.07 | 82.85±0.19 | 93.60±0.10 |
| +BiFI-TDM | 83.80±0.19 +1.39 | **94.26±0.09** +1.89 | **76.72±0.20** +0.61 | **88.96±0.12** +0.51 | **89.15±0.16** +1.94 | **97.31±0.05** +1.20 | **84.42±0.18** +1.57 | **95.57±0.06** +1.97 |

### E. Training Loss

We utilise three losses to train the network. First, the cross-entropy loss is adopted to quantify the classification loss of the query set:

$$\mathcal{L}_{CE} = -\log\left(P(\hat{y}_q = y_q|\mathbf{x}_q)\right). \tag{10}$$

Second, we introduce an orthogonal loss to ensure that the two types of features are distinct. By averaging $\mathbf{F}$ and $\mathbf{G}$ over $H \times W$ dimensions, we obtain $\bar{\mathbf{F}} \in \mathbb{R}^C$ and $\bar{\mathbf{G}} \in \mathbb{R}^C$. The cosine similarity between them is then calculated as the orthogonal loss:

$$\mathcal{L}_{\text{orth}} = \frac{\bar{\mathbf{F}}^T\bar{\mathbf{G}}}{\|\bar{\mathbf{F}}\|\|\bar{\mathbf{G}}\|}. \tag{11}$$

To facilitate better discriminative ability of the two types of query features, an additional linear layer is added after each feature extraction module as an classifier, supervised by two losses, $\mathcal{L}_F$ and $\mathcal{L}_G$:

$$\begin{aligned}\mathcal{L}_F &= -\log\frac{e^{\mathbf{w}_n^T\bar{\mathbf{F}}_q+b_n}}{\sum_{n'=1}^{N}e^{\mathbf{w}_{n'}^T\bar{\mathbf{F}}_q+b_{n'}}},\\ \mathcal{L}_G &= -\log\frac{e^{\mathbf{v}_n^T\bar{\mathbf{G}}_q+m_n}}{\sum_{n'=1}^{N}e^{\mathbf{v}_{n'}^T\bar{\mathbf{G}}_q+m_{n'}}},\end{aligned} \tag{12}$$

where $\{\mathbf{w}_n, b_n\}_{n=1}^N$ and $\{\mathbf{v}_n, m_n\}_{n=1}^N$ represent the weights and biases of the linear layers and $n$ is the correct label for the corresponding features. These two additional losses can also prevent situations where one backbone learns only foreground features and the other learns only background features due to the orthogonal loss.

Finally, the total loss is calculated as

$$\mathcal{L}_T = \mathcal{L}_{CE} + \gamma(\omega\mathcal{L}_F + (1-\omega)\mathcal{L}_G) + \mu\mathcal{L}_{\text{orth}}, \tag{13}$$

where $\gamma$ and $\mu$ are hyper-parameters and $\omega \in (0,1)$ is a learnable parameter to balance $\mathcal{L}_F$ and $\mathcal{L}_G$.

## IV. EXPERIMENTS

In section IV-C, we evaluate BiFI-TDM on four benchmark fine-grained image datasets against the state-of-the-art techniques. The effectiveness of each element in BiFI-TDM is studied in ablation studies in section IV-D. Qualitative visualisations are also presented in section IV-E to illustrate the discriminative regions and channels identified by BiFI-TDM.

### A. Datasets

We benchmark few-shot fine-grained image classifiers on four datasets:

CUB-200-2011 (CUB) [13]: a total of 11,788 bird images, distributed across 200 different bird species, with 100 training, 50 validation and 50 test categories. Following recent works [21], [29], we use the pre-cropped images with human-annotated bounding boxes for labeling.

Flowers [30]: 8,189 flower images of 102 species. It is divided to 51 training, 26 validation, and 25 test categories.

Cars [31]: 11,788 images with 196 car classes. The training set consists of 130 classes, the validation set 17 classes and the test set 49 classes.

Dogs [32]: 20,580 annotated images of 120 breeds of dogs from around the world. It consists of 60 training, 30 validation and 30 test categories.

### B. Implementation Details

We conduct experiments using ResNet-12 as the backbone structure [21], [29], [33]. The input images are resized to $3\times84\times84$, and the output feature maps are of sizes $640\times5\times5$. We train the model by two few-shot settings, 5-way 1-shot and 5-way 5-shot. In meta test, we test each class with 15 query samples in one episode and report the average classification accuracy along with its 95% confidence interval. This is obtained by randomly sampling 10,000 test episodes. For all datasets, we set $u$=0.5 and $v$=0.5 following TDM [12]. For the total loss, we set $\gamma$=0.25 for the CUB and Cars datasets, while $\gamma$=1 for the Dogs and Flowers datasets. We set $\mu$=0.01

TABLE III
THE IMPACT OF THE TWO TYPES OF FEATURES AND THE INTERACTIONS.

| | Interaction | $\mathbf{F}$ | $\mathbf{G}$ | Cars | | Flowers | |
|---|---|---|---|---|---|---|---|
| | | | | 1-shot | 5-shot | 1-shot | 5-shot |
| (a) | ✗ | ✗ | ✗ | 87.21 | 96.11 | 82.85 | 93.60 |
| (b) | ✗ | ✓ | ✓ | 88.45 | 97.03 | 83.11 | 94.70 |
| | ✓ | ✓ | ✓ | **89.15** | **97.31** | **84.42** | **95.57** |

TABLE IV
MORE TESTS ON THE IMPACT OF INTERACTIONS.

| | $\alpha^{GF}$ | $\alpha^{FG}$ | Cars | | Flowers | |
|---|---|---|---|---|---|---|
| | | | 1-shot | 5-shot | 1-shot | 5-shot |
| (a) | ✗ | ✗ | 88.23 | 96.67 | 83.56 | 94.21 |
| (b) | ✓ | ✗ | 88.90 | 97.19 | 84.04 | 95.06 |
| (c) | ✗ | ✓ | 88.93 | 97.17 | 84.09 | 94.79 |
| Ours | ✓ | ✓ | **89.15** | **97.31** | **84.42** | **95.57** |

TABLE V
THE VALUES OF $\alpha$'S IN EQ.(8) FOR FOUR DATASETS.

| | CUB | Dogs | Cars | Flowers |
|---|---|---|---|---|
| $\alpha^{FF}$ | 0.458 | 1.713 | 0.180 | 0.633 |
| $\alpha^{GF}$ | 2.952 | 1.805 | 3.247 | 3.796 |
| $\alpha^{FG}$ | 3.791 | 3.009 | 5.853 | 2.586 |
| $\alpha^{GG}$ | 0.064 | 0.953 | 0.116 | 0.980 |

TABLE VI
THE IMPACT OF THE ELEMENTS IN TOTAL LOSS.

| $\mathcal{L}_{\text{orth}}$ | $\mathcal{L}_F$ | $\mathcal{L}_G$ | Cars | | Flowers | |
|---|---|---|---|---|---|---|
| | | | 1-shot | 5-shot | 1-shot | 5-shot |
| ✓ | ✗ | ✗ | 88.11 | 96.30 | 83.06 | 94.51 |
| ✗ | ✓ | ✗ | 88.21 | 96.23 | 83.10 | 94.62 |
| ✗ | ✗ | ✓ | 88.33 | 96.19 | 83.08 | 94.82 |
| ✗ | ✓ | ✓ | 88.66 | 96.58 | 83.72 | 94.76 |
| ✓ | ✓ | ✗ | 88.83 | 96.81 | 83.88 | 94.96 |
| ✓ | ✗ | ✓ | 88.77 | 97.10 | 83.92 | 94.72 |
| ✓ | ✓ | ✓ | **89.15** | **97.31** | **84.42** | **95.57** |

for Dogs and Flowers, 0.1 for CUB and 2 for Cars. $\alpha$'s in the metric and $\omega$ in the total loss are learnt during the training process.

### C. Comparison with the State-of-the-art Methods

We compare BiFI-TDM with TDM in two metric-based methods, ProtoNet [7] and FRN [14]. We also compare them with the following state-of-the-art methods, MatchingNet [6], ProtoNet [7], Baseline++ [20], DeepEMD [21], RENet [15], MCL [22], FicNet [23], LCCRN [24], BSFA [25], BiFRN [26], AIS-MLI [27] and C2-Net [28]. The classification accuracies together with their 95% confidence intervals are reported in Table II. Obviously, FRN+BiFI-TDM can beat most competitors, expect for AIS-MLI on 1-shot CUB data. Moreover, BiFI-TDM dominates TDM on both ProtoNet and FRN for all datasets and few-shot settings, which demonstrates the effectiveness of the feature fusion strategy in BiFI-TDM. Notably, ProtoNet+BiFI-TDM offers significant improvements over ProtoNet+TDM in the more challenging 1-shot scenarios. Additionally, FRN+BiFI-TDM significantly outperforms FRN+TDM in both 1-shot and 5-shot scenarios. Thus, simply enriching the input features and allowing interactions between them without modifying the attention mechanism is an effective way to boost the classification performance.

### D. Ablation Studies

We conduct extensive ablation experiments to evaluate the effectiveness of BiFI-TDM. In the following experiments, we adopt the 5-way setting and present the results with FRN as the metric-based method.

*1) The impact of the components of BiFI-TDM:* BiFI-TDM consists of two main components: $\mathbf{F}$ and $\mathbf{G}$, along with their interactions. In Table III, we examine how each component affects classification performance. In scenario (a), we use only a single feature extractor, and without $\mathbf{F}$, $\mathbf{G}$, or feature interaction, the model degrades to TMD. In scenario (b), we use two independent backbones with orthogonal loss to generate $\mathbf{F}$ and $\mathbf{G}$, but without feature interaction. The experimental results show a significant improvement, indicating that the two feature extractors, constrained by the orthogonal loss, can outperform TDM even without feature interaction. Finally, using $\mathbf{F}$ and $\mathbf{G}$ along with their interaction achieves the highest classification accuracy.

Previous analysis exhibits that allowing interactions between features is one key to success. In Table IV, we present more tests on the interactions with $\mathbf{F}$ and $\mathbf{G}$ features. By setting $\alpha^{GF}$ or $\alpha^{FG}$ or both as ✗, we exclude the corresponding interaction term(s) in Eq.(8). It is clear that even by involving only one interaction term in scenarios-(b) and (c) can promote accuracies.

To further show the value of the interaction terms, we present the values of $\alpha$'s in Eq.(8) for four datasets in Table V. Clearly, the interaction terms are assigned the highest weights and play an important role to determine the final classification.

*2) The impact of the elements in total loss:* In Table VI, we investigate the impact of $\mathcal{L}_F$, $\mathcal{L}_G$ and $\mathcal{L}_{\text{orth}}$ on classification accuracy. Note that we retain $\mathcal{L}_{CE}$ in all scenarios to train the model, while individually removing one or two of $\mathcal{L}_F$, $\mathcal{L}_G$ and $\mathcal{L}_{\text{orth}}$. The experimental results show that adding each loss individually leads to performance improvements in most scenarios. This pattern demonstrates the importance of $\mathcal{L}_F$, $\mathcal{L}_G$, and $\mathcal{L}_{\text{orth}}$ in enhancing the discriminative capability of the query features.

*3) The impact of $u$ and $v$:* Next, we conduct a sensitivity analysis of $u$ and $v$ in equations (1) and (3), respectively. In Fig. 3, we test the effect of pairs of $u$ and $v$ ranging in $[0.1, 0.9]$ on the Cars dataset for 5-way 1-shot and 5-shot tasks as examples. We observe that the test accuracy peaks when both $u$ and $v$ are set to 0.5, while significantly deteriorates when both $u$ and $v$ approach extreme values, such as 0.1 and 0.9.

*4) The impact of the learnability of $\gamma$ and $\mu$:* To study the impact of the learnability of $\gamma$ and $\mu$ in Eq.(13), in Table VII,

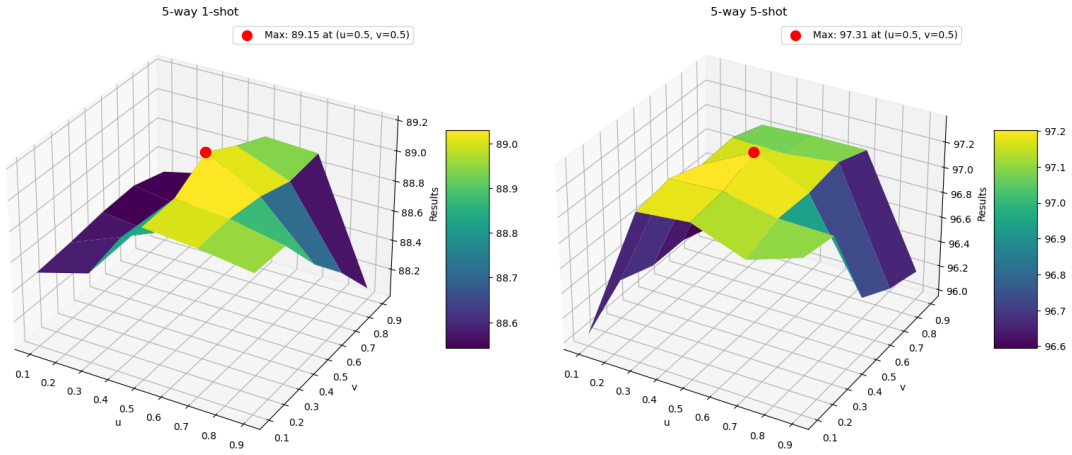



Fig. 3. The impact of channel weight mixing values $u$ and $v$ on the Cars dataset for 5-way 1-shot and 5-shot tasks.

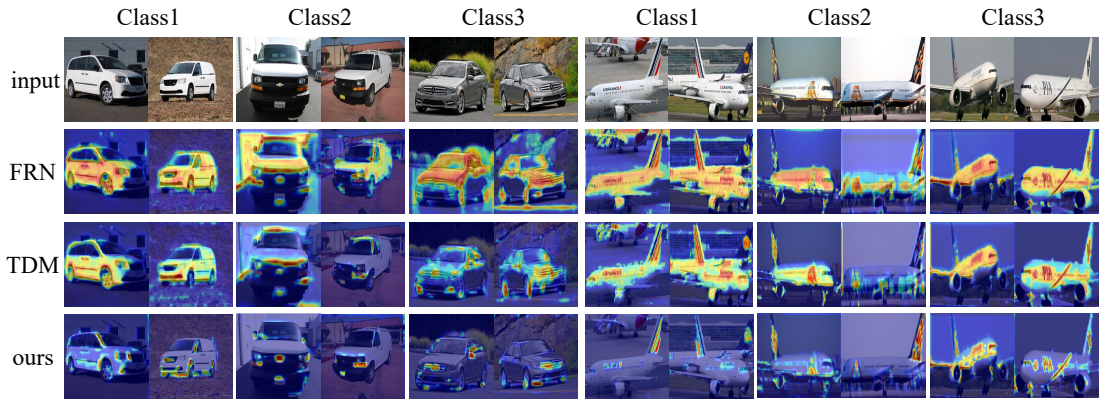



Fig. 4. From top to bottom: input image and their visualisations of discriminative features captured by FRN, FRN+TDM ('TDM') and FRN+BiFI-TDM ('ours') on three classes of the Cars and Aircraft datasets. Our method can highlight the most delicate and consistent areas for classification, e.g. headlights and logos of cars and tails of aircrafts, while being less affected by background and irrelavent objects.

TABLE VII
THE IMPACT OF THE LEARNABILITY OF $\gamma$ AND $\mu$. "✓" REPRESENTS A **LEARNABLE** PARAMETER AND "✗" REPRESENTS A **FIXED** PARAMETER.

| $\omega$ | $\gamma$ | $\mu$ | Cars | | Flowers | |
|---|---|---|---|---|---|---|
| | | | 1-shot | 5-shot | 1-shot | 5-shot |
| ✓ | ✗ | ✗ | 89.15 | **97.31** | **84.42** | 95.57 |
| ✓ | ✓ | ✓ | **89.33** | 97.28 | 83.90 | **95.71** |

TABLE VIII
THE IMPACT OF THE NUMBER OF DISTINCT FEATURES.

| | Cars | | Flowers | |
|---|---|---|---|---|
| | 1-shot | 5-shot | 1-shot | 5-shot |
| 1 | 65.89 | 82.45 | 70.66 | 85.14 |
| 2 | **68.84** | **86.62** | **73.45** | 88.97 |
| 3 | 66.04 | 85.74 | 72.37 | **89.27** |
| 4 | 64.49 | 83.81 | 72.01 | 88.69 |
| 5 | 60.30 | 80.30 | 70.85 | 88.10 |

we compare the classification accuracies of Cars and Flowers datasets when $\gamma$ and $\mu$ are either learnable or fixed. For the fixed value scenario, we use the same values as described in section IV-B. In the learnable scenario, to prevent $\gamma$ and $\mu$ from being learned as extreme negative values, we limit their ranges to $[0.001, 3]$. No significant difference can be observed when comparing the two rows in Table VII, except for a slightly better performance of using fixed values for 1-shot Flowers.

*5) The impact of the number of distinct features:* Finally, we explore the impact of the number of distinct features to be fused in our proposed scheme. For illustration purpose, we use smaller Conv-4 [33], [34] as the backbone network instead of ResNet-12. As shown in Table VIII, a larger number of distinct features tends to provide worse classification accuracies. When using two distinct features, as in BiFI-TDM, we observe large improvements compared with using a single feature, as in TDM. However, as the number of distinct features further increases, the classification accuracy declines in almost all scenarios. A possible explanation for this pattern is that incorporating more distinct features or backbones significantly increases the number of parameters, leading to severe overfitting, especially for the small-sample task considered in this paper.

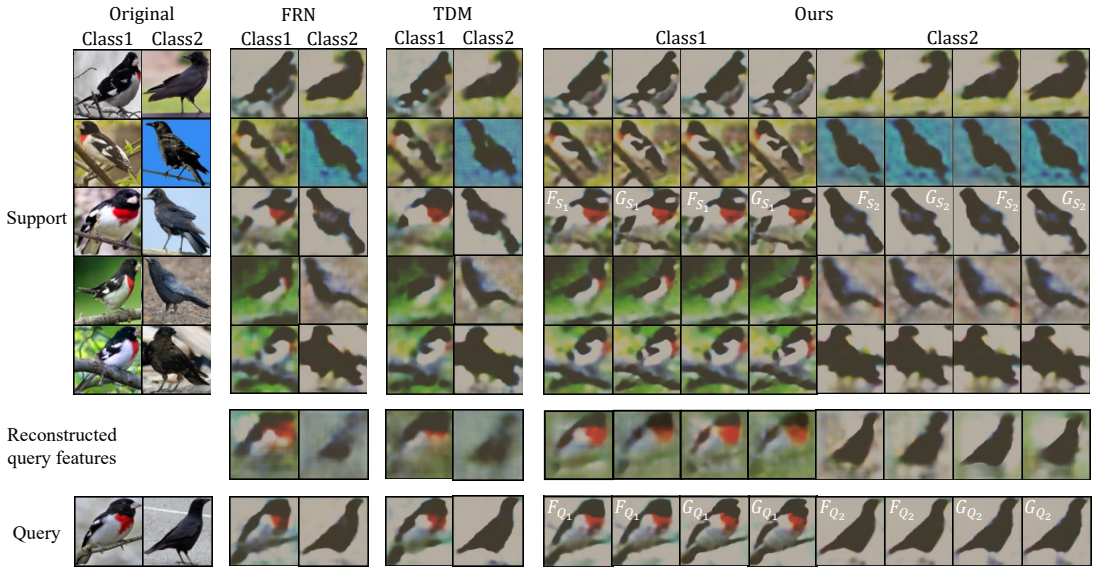



Fig. 5. Top (bottom) panel from left to right: the original support (query) images, the support (query) features extracted by FRN, FRN+TDM ('TDM') and FRN+BiFI-TDM ('Ours'). Middle panel: the reconstructed query features by the three methods. The eight reconstructed query features (four for each class) from our method show complementary information about the two birds and present more details.

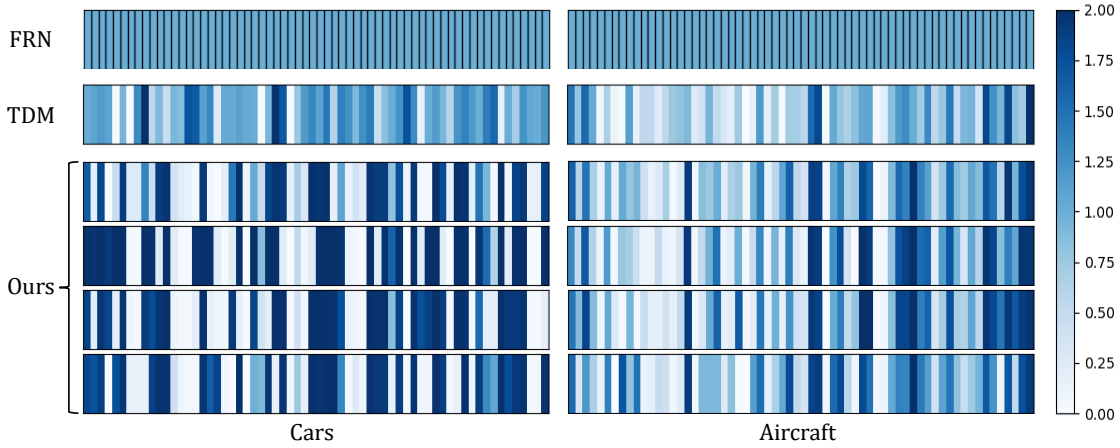



Fig. 6. The visualisations the channel weights of FRN, FRN+TDM ('TDM') and FRN+BiFI-TDM ('Ours') on the Cars and Aircraft datasets. For ours panel, from top to bottom, we visualise $\mathbf{w}^{FF}$, $\mathbf{w}^{FG}$, $\mathbf{w}^{GF}$ and $\mathbf{w}^{GG}$, respectively. The weights obtained by our method from different features and their interactions stress different channels, complementing each other to uplift the classification performance.

### *E. Qualitative Comparisons via Visualisations*

In this section, we first provide visual comparisons of the discriminative regions in Fig. 4 for the Cars and Aircraft datasets with FRN as the metric module. Similar patterns can be observed as in Fig. 1. FRN+BiFI-TDM can identify the most delicate areas to distinguish subcategories, compared with FRN+TDM and FRN. Moreover, consistent areas are highlighted. For example, to classify cars, the headlights and the front logos are important, while to classify aircrafts, the tails painted with company logos are commonly identified. This also matches how humans recognise cars and aircrafts.

Additionally, we depict two examples of the reconstructed query images in the CUB dataset through FRN. In Fig. 5, the upper and bottom panels visualise the features of the support and query images, while the middle panel presents the reconstructed query images. In FRN+BiFI-TDM, four reconstructed images are obtained via the four sets of features. It is clear that the different features can capture more detailed information than the base features; thus, utilising them can provide better classification results. This is consistent with the analysis in Table III. Not surprisingly, the reconstructed query images of our method also present more details. Moreover, the four reconstructed images show complementary details, suggesting that all sets of features shall be involved to determine the image labels.

Finally, to validate that the four sets of channel weights identify different discriminative channels, we visualized the $\mathbf{w}$'s in Eq.(3) in Fig. 6. The darker the colour is, the higher the weight of the channel. Plain FRN gives all channels the same weight of 1 while TDM highlights discriminative channels. The four rows of our method correspond to $\mathbf{w}^{FF}$, $\mathbf{w}^{FG}$, $\mathbf{w}^{GF}$, and $\mathbf{w}^{GG}$, respectively. Clearly, they weigh different channels,

TABLE IX
THE FLOPS AND NUMBER OF PARAMETERS OF TDM AND BIFI-TDM.

| | FLOPs | Params |
|---|---|---|
| TDM | 3.52 G | 17.35 M |
| BiFI-TDM | 7.05 G | 29.84 M |

and this pattern is more obvious in the Cars dataset.

### F. Computational Cost

Since BiFI-TDM consists of two branches of TDM, the number of parameters and FLOPs of BiFI-TDM are higher than TDM, as shown in Table IX. However, with this acceptable sacrifice on computational cost, we can achieve noticeable improvement on classification accuracies for fine-grained data.

## V. CONCLUSION

In this work, we propose the bi-feature interactive TDM (BiFI-TDM) module for few-shot fine-grained image classification. BiFI-TDM takes two types of features as input, capturing different properties of the target object. Through the novel mixing strategy, we encourage interactions between the two types of features, allowing TDM to search for discriminative channels with ease, and generate four sets of channel weights with different emphases. The weights are then applied to reweight the corresponding features via the feature weighting scheme. We conduct extensive experiments and ablation study on four fine-grained benchmark datasets, demonstrating well the effectiveness of BiFI-TDM and its components.

## ACKNOWLEDGEMENT

This work was partly supported by the National Nature Science Foundation of China (Grants 62176110, 62463015, 62225601, U23B2052), the Key Research and Development Program of Gansu Province, China under Grant 22YF7GA130, S&T Program of Hebei, China under Grant SZX2020034, Hong-Liu Distinguished Young Talents Foundation of Lanzhou University of Technology, Beijing Natural Science Foundation Project under Grant L242025, the Youth Innovative Research Team of BUPT under Grant 2023YQTD02, and the Royal Society under International Exchanges Award IEC\NSFC\201071.

## REFERENCES

[1] X. Wang, X. Wang, B. Jiang, and B. Luo, "Few-shot learning meets transformer: Unified query-support transformers for few-shot classification," *IEEE Transactions on Circuits and Systems for Video Technology*, 2023.

[2] X. Shu, J. Tang, G.-J. Qi, Z. Li, Y.-G. Jiang, and S. Yan, "Image classification with tailored fine-grained dictionaries," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 28, pp. 454–467, 2018.

[3] H. Huang, J. Zhang, J. Zhang, J. Xu, and Q. Wu, "Low-rank pairwise alignment bilinear network for few-shot fine-grained image classification," *IEEE Transactions on Multimedia*, vol. 23, pp. 1666–1680, 2019.

[4] X. Li, J. Wu, Z. Sun, Z. Ma, J. Cao, and J.-H. Xue, "BSNet: Bi-similarity network for few-shot fine-grained image classification," *IEEE Transactions on Image Processing*, vol. 30, pp. 1318–1331, 2020.

[5] Z. Fang, X. Jiang, H. Tang, and Z. Li, "Learning contrastive self-distillation for ultra-fine-grained visual categorization targeting limited samples," *IEEE Transactions on Circuits and Systems for Video Technology*, pp. 1–1, 2024.

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

[7] J. Snell, K. Swersky, and R. S. Zemel, "Prototypical networks for few-shot learning," in *Neural Information Processing Systems*, 2017.

[8] F. Sung, Y. Yang, L. Zhang, T. Xiang, P. H. S. Torr, and T. M. Hospedales, "Learning to compare: Relation network for few-shot learning," *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 1199–1208, 2017.

[9] F. Zhou, L. Zhang, and W. Wei, "Meta-generating deep attentive metric for few-shot classification," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 32, no. 10, pp. 6863–6873, 2022.

[10] S.-L. Xu, F. Zhang, X.-S. Wei, and J. Wang, "Dual attention networks for few-shot fine-grained recognition," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 36, no. 3, 2022, pp. 2911–2919.

[11] H. Tang, C. Yuan, Z. Li, and J. Tang, "Learning attention-guided pyramidal features for few-shot fine-grained recognition," *Pattern Recognition*, vol. 130, p. 108792, 2022.

[12] S. Lee, W. Moon, and J.-P. Heo, "Task discrepancy maximization for fine-grained few-shot classification," *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 5321–5330, 2022.

[13] 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.

[14] D. Wertheimer, L. Tang, and B. Hariharan, "Few-shot classification with feature map reconstruction networks," *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 8008–8017, 2020.

[15] D. Kang, H. Kwon, J. Min, and M. Cho, "Relational embedding for few-shot classification," *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 8802–8813, 2021.

[16] V. G. Satorras and J. Bruna, "Few-shot learning with graph neural networks," *ArXiv*, vol. abs/1711.04043, 2017.

[17] S. Tang, D. Chen, L. Bai, K. Liu, Y. Ge, W. Ouyang, and H. Kong, "Mutual crf-gnn for few-shot learning," *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 2329–2339, 2021.

[18] H. Tang, Z. Li, Z. Peng, and J. Tang, "Blockmix: Meta regularization and self-calibrated inference for metric-based meta-learning," in *Proceedings of the 28th ACM International Conference on Multimedia*, ser. MM '20. New York, NY, USA: Association for Computing Machinery, 2020, p. 610–618. [Online]. Available: https://doi.org/10.1145/3394171.3413884

[19] Z. Li, H. Tang, Z. Peng, G.-J. Qi, and J. Tang, "Knowledge-guided semantic transfer network for few-shot image recognition," *IEEE Transactions on Neural Networks and Learning Systems*, pp. 1–15, 2023.

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

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

[22] Y. Liu, W. Zhang, C. Xiang, T. Zheng, and D. Cai, "Learning to affiliate: Mutual centralized learning for few-shot classification," *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 14 391–14 400, 2021.

[23] H. Zhu, Z. Gao, J. Wang, Y. Zhou, and C. Li, "Few-shot fine-grained image classification via multi-frequency neighborhood and double-cross modulation," *ArXiv*, vol. abs/2207.08547, 2022.

[24] X. Li, Q. Song, J. Wu, R. Zhu, Z. Ma, and J.-H. Xue, "Locally-enriched cross-reconstruction for few-shot fine-grained image classification," *IEEE Transactions on Circuits and Systems for Video Technology*, 2023.

[25] Z. Zha, H. Tang, Y. Sun, and J. Tang, "Boosting few-shot fine-grained recognition with background suppression and foreground alignment," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 33, pp. 3947–3961, 2022.

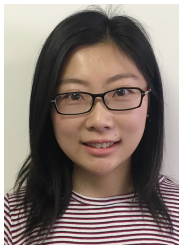
[26] J. Wu, D. Chang, A. Sain, X. Li, Z. Ma, J. Cao, J. Guo, and Y.-Z. Song, "Bi-directional feature reconstruction network for fine-grained few-shot image classification," *ArXiv*, vol. abs/2211.17161, 2022.

[27] L.-J. Zhao, Z.-D. Chen, Z.-X. Ma, X. Luo, and X.-S. Xu, "Angular isotonic loss guided multi-layer integration for few-shot fine-grained

image classification," *IEEE Transactions on Image Processing*, vol. 33, pp. 3778–3792, 2024.
[28] Z.-X. Ma, Z.-D. Chen, L.-J. Zhao, Z.-C. Zhang, X. Luo, and X.-S. Xu, "Cross-layer and cross-sample feature optimization network for few-shot fine-grained image classification," vol. 38, no. 5, pp. 4136–4144, 2024.
[29] H.-J. Ye, H. Hu, D. chuan Zhan, and F. Sha, "Few-shot learning via embedding adaptation with set-to-set functions," *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 8805–8814, 2018.
[30] M.-E. Nilsback and A. Zisserman, "Automated flower classification over a large number of classes," *2008 Sixth Indian Conference on Computer Vision, Graphics & Image Processing*, pp. 722–729, 2008.
[31] J. Krause, M. Stark, J. Deng, and L. Fei-Fei, "3D object representations for fine-grained categorization," *2013 IEEE International Conference on Computer Vision Workshops*, pp. 554–561, 2013.
[32] A. Khosla, N. Jayadevaprakash, B. Yao, and L. Fei-Fei, "Novel dataset for fine-grained image categorization : Stanford dogs," 2012.
[33] K. Lee, S. Maji, A. Ravichandran, and S. Soatto, "Meta-learning with differentiable convex optimization," *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 10 649–10 657, 2019.
[34] M. Tong, S. Wang, B. Xu, Y. Cao, M. Liu, L. Hou, and J.-Z. Li, "Learning from miscellaneous other-class words for few-shot named entity recognition," in *Annual Meeting of the Association for Computational Linguistics*, 2021. [Online]. Available: https://api.semanticscholar.org/CorpusID:235669988

**Xiaoxu Li** received the Ph.D. degree from Beijing University of Posts and Telecommunications in 2012. She is currently a Professor with the School of Computer and Communication, Lanzhou University of Technology. Her research interests include machine learning fundamentals with a focus on applications in image and video understanding. She is also a member of the China Computer Federation.

**Peiyu Lu** He is currently working toward the M.E. degree with Lanzhou University of Technology. His research interests include computer vision and few-shot learning.



**Rui Zhu** received the Ph.D. degree in statistics from University College London in 2017. She is a Senior Lecturer in Statistics in the Faculty of Actuarial Science and Insurance, City St George's, University of London. Her research interests include machine learning, computer vision and interdisciplinary applications in actuarial science. She serves as the Associate Editor for IEEE Transactions on Neural Networks and Learning Systems, IEEE Transactions on Circuits and Systems for Video Technology and Neurocomputing.

**Zhanyu Ma** is currently a Professor at Beijing University of Posts and Telecommunications, Beijing, China, since 2019. He received the Ph.D. degree in electrical engineering from KTH Royal Institute of Technology, Sweden, in 2011. From 2012 to 2013, he was a Postdoctoral Research Fellow with the School of Electrical Engineering, KTH. He has been an Associate Professor with the Beijing University of Posts and Telecommunications, Beijing, China, from 2014 to 2019. His research interests include pattern recognition and machine learning fundamentals with a focus on applications in computer vision, multimedia signal processing. He is a Senior Member of IEEE.

**Jie Cao** received the M.E. degree from Xi'an Jiaotong University, China, in 1994. She is currently a Professor and a Vice President of the Lanzhou University of Technology. Her research interests include machine learning, pattern recognition, speech and speaker recognition, information fusion, and computer vision.

**Jing-Hao Xue** received the Dr.Eng. degree in signal and information processing from Tsinghua University in 1998, and the Ph.D. degree in statistics from the University of Glasgow in 2008. He is a Professor of Statistical Pattern Recognition in the Department of Statistical Science, University College London. His research interests include statistical pattern recognition, machine learning, and computer vision. He received the Best Associate Editor Award of 2021 from the IEEE Transactions on Circuits and Systems for Video Technology, and the Outstanding Associate Editor Award of 2022 from the IEEE Transactions on Neural Networks and Learning Systems.