Few-Shot Fine-Grained Image Classification via Vision Transformer

Yongqi Liu[1], Tong Xiao[2], Zeao Chen[1], Chen Zhou[1] and Zhi-Jie Wang[1†]

***Abstract*— Few-shot fine-grained image classification (FSFGIC) is to classify images of the same class into fine-grained subclasses, where only a very limited number of labeled samples in each subclass are available (e.g., 5 or even 1 labeled sample). For most of existing methods, the feature representation capabilities are insufficient, which may harm the performance. Vision Transformers (ViTs) have shown strong feature representation capabilities in many research fields. In this paper, we attempt to solve FSFGIC problem via ViT or its variants. Generally, we use an enhanced ViT as the backbone and adopt a three-stage training strategy. More specifically, (i) we utilize an image matting module to enhance the focus of ViT on the main subjects of images; (ii) we introduce a part selection module to better focus on local image details; and (iii) we incorporate an AdaptMLP module into each Transformer Encoder to reduce the number of parameters that require fine-tuning. Extensive experiments based on three benchmark datasets show us that the proposed model is highly competitive, compared against state-of-the-art models.**

I. INTRODUCTION

Image classification has received much attention due to its wide applications [1]. In this community, there are two hot topics: (i) few-shot image classification (FSIC), and (ii) fine-grained image classification (FGIC). Generally speaking, FSIC is required to recognize new image classes based on a limited number of labeled samples. Compared to traditional image classification, the scarcity of samples in FSIC easily incurs inadequate model training, posing a significant challenge when models encounter new classes [2], [3]. As for FGIC, it needs to classify images within the same class into fine-grained subclasses. This requires the model to possess a strong feature representation capability, so as to focus on the subtle feature variations between subclasses [4], [5]. Recently, an increasing number of researchers have begun to explore a more challenging problem: few-shot fine-grained image classification (FSFGIC), which is a combination of the above problems. FSFGIC integrates their challenges, namely, learning subtle feature differences between subclasses when only limited labeled samples are available [6], [7].

In literatures, a popular approach is to employ lightweight models (e.g., Conv-4, ResNet-12) as the backbone for model training. The limited number of parameters in these backbones is helpful to mitigate the challenges posed by the scarcity of training samples in few-shot settings. Nevertheless, the feature representation capabilities of these lightweight models are often inferior to large-parameter models, particularly in capturing fine-grained details [8]. We note that some researchers have begun to employ large-parameter models (e.g., ResNet-50 and ResNet-256) for FSFGIC problem. Nevertheless, compared to competitive models, the above mentioned models are considered to be outdated, as their feature representation capabilities are still insufficient [9].

Recently, Vision Transformers (ViTs) [10], owing to their powerful feature representation capability, have demonstrated excellent performance in many research fields including object detection [11], facial recognition [12], video classification [13], and fine-grained image classification [14]. A natural question is: whether ViT can be also effectively used for FSFGIC problem? To this end, we make the first attempt to solve FSFGIC problem via ViT or its variants. We note that, under the constraints of few-shot setting, the limited training samples challenge the effective training of ViTs, which have a large number of parameters. This impedes ViTs to fully realize their potential. To address it, we develop a framework that not only addresses these training challenges but also further enhances the attention to fine-grained details, thereby achieving a favorable performance. Generally, we employ an enhanced ViT as the backbone and adopt a three-stage training pipeline (pre-training, meta-training, and fine-tuning). More specifically, to enhance the focus of ViT on the main subjects of images, we utilize an image matting module to segment the foreground from each image and eliminate background noise; to focus on local image details, we introduce a part selection module before the last Transformer Encoder in our ViT; in addition, we incorporate an AdaptMLP module into each Transformer Encoder to reduce the number of parameters that require fine-tuning. To summarize, our main contributions can be listed as follows:

- We develop a ViT-based model to address FSFGIC problem. Our model is simple, easy-to-implement, but without loss of effectiveness. To the best of our knowledge, this is the first work to address FSFGIC based on ViTs.
- We conduct extensive experiments on three benchmark datasets including CUB-200-2011, Stanford Dogs, and Stanford Car. The experimental results consistently show us that the proposed method can achieve competitive performance, compared against state-of-the-art methods.

†: Corresponding Author

[1]Yongqi Liu, Zeao Chen, Chen Zhou and Zhi-Jie Wang are with College of Computer Science, Chongqing University, Chongqing 400044, China. 2742651864@qq.com, 845969479@qq.com, 420739269@qq.com, cszjwang@cqu.edu.cn

[2]Tong Xiao is with the Department of Computer Science and Technology, Tsinghua University, Beijing 100084, China. xiaotong@tsinghua.edu.cn

II. Related work

In this section, we review previous work most relevant to ours.

Few-shot image classification. Methods for FSIC can be classified into three categories: (i) meta learning-based [15], (ii) data augmentation-based [3], and (iii) transfer learning-based [2]. Meta learning is to "learn how to learn". Typical meta learning-based methods include ProtoNet [15] and MAML [16]. Data augmentation-based methods expand the training set by generating additional samples [3]. As for transfer learning-based methods, they often leverage models pre-trained on large datasets, and mainly involve pre-training stage and fine-tuning stage, in order to transfer existing knowledge to new tasks [17]. The above mentioned methods work well for FSIC, they however are ineffective for FGIC.

Fine-grained image classification. Methods for FGIC can be roughly divided into two categories: (i) localization-based, and (ii) feature encoding-based. The former often trains a network to identify the most discriminative regions in images and reuses them for classification; typical methods include Deep LAC [4], DF-GMM [5], and PA-CNN [18]. In contrast, the latter often acquires more insightful features through either deriving higher-level data interactions or uncovering the connections within comparative element pairs; typical methods include BCNN [19], DeepKSPD [20] and FDL [21]. The above methods work well for FGIC, whereas they are often unusable for FSIC.

Few-shot fine-grained image classification. As for FS-FGIC, researchers also made a lot of efforts. Existing studies can be generally classified into two categories: (i) light-weight-based methods, where they select light-weight models such as Conv-4 and ResNet-12 as the backbones; and (ii) heavy-weight-based methods, where they employ models with large-scale parameters such as ResNet-50 and ResNet-256 as the backbones. The former ones can be adequately trained on a small number of samples, and representative methods include C2-Net [8], BSFA [6], BSNet [7], Fic-Net [22], MetaIRNet [23], BTG-net [24], and so on. As for the latter, they often have better feature representation capabilities, and representative methods include DEML [9] and RaPSPNet [25]. Nevertheless, the feature representation capabilities of these methods are still insufficient. We note that ViTs have larger parameters and stronger feature representation capabilities, and exhibited excellent performance in various fields such as object detection [11], text recognition [26], and video classification [13]. It is obviously interesting and valuable to explore ViT-based method for FSFGIC problem. This is just the focus of our paper.

III. Methodology

A. Problem Formulation

Following the previous study [27], given a dataset $D = \{(X_i, y_i) \mid y_i \in Y\}$, we divide it into three parts: training set $D_{train} = \{(X_i, y_i) \mid y_i \in Y_{train}\}$, validation set $D_{val} = \{(X_i, y_i) \mid y_i \in Y_{val}\}$, and test set $D_{test} = \{(X_i, y_i) \mid y_i \in Y_{test}\}$, where (X_i, y_i) is the raw feature vector and class label of the i^{th} image, Y is the set of all class labels in D, $Y_{train} \cup Y_{test} \cup Y_{val} = Y$ and $Y_{train} \cap Y_{test} \cap Y_{val} = \emptyset$. FSFGIC aims to perform an "n-way k-shot" image classification task on D_{test} using the knowledge learned from D_{train} and validated on D_{val} with the same task; where n is the number of classes randomly sampled from Y_{test}/Y_{train}/Y_{val}, and k is the number of labeled images randomly sampled from each class. In addition, u unlabeled images are randomly sampled from each class. The set of the $n \times k$ labeled images is called the support set, and the set of the $n \times u$ unlabeled images is the query set. The goal of the "n-way k-shot" image classification task is to classify the images in the query set given the support set.

B. Overview of Our Framework

As shown in Fig. 1, our framework takes ViT as the backbone, on which we integrate three key modules (*i.e.*, ① Image Matting, ② AdaptMLP, and ③ Part Selection) to accomplish the FSFGIC task. The overall workflow of our framework is as follows:

(1) Firstly, the query image for classification goes through the Image Matting module, which removes the background and retains only the foreground of the image, to eliminate the influence of background noise.

(2) Secondly, the image is split into fixed-size patches in a grid-based manner. Specifically, assume the resolution of the image is $(H \times W)$ and that of the patch is $(P \times P)$, then the image is split into $\left(\frac{H}{P}\right) \times \left(\frac{W}{P}\right)$ patches.

(3) Thirdly, these patches are flattened and transformed into a sequence of vectors (called patch embeddings) through a linear projection. In addition, a learnable class embedding is prepended to the sequence of patch embeddings, whose final state serves as the representation of the image and is used for classification.

(4) Fourthly, each patch embedding is added with a position embedding, to retain the position information of each patch.

(5) Fifthly, the sequence of patch embeddings is fed into $L-1$ layers of Transformer Encoder, each of which has a Multi-Head Self-Attention (MHSA) block and an AdaptMLP module. Inspired by [28], here the AdaptMLP module is used to replace the MLP block in each Transformer Encoder, so as to reduce the number of parameters that need fine-tuning.

(6) Sixthly, the Part Selection module is used to determine the importance of each patch embedding, according to the $L-1$ attention matrices in the aforementioned Transformer Encoders. Only important patch embeddings that contain discriminative feature details are retained and passed to the final layer of the Transfomer Encoder with the class embedding.

(7) Finally, the class embedding after the last layer of Transfomer Encoder is input into the ProtoNet for classification.

Considering the large parameter size of ViT, our framework adopts a three-stage training pipeline to ensure suf-

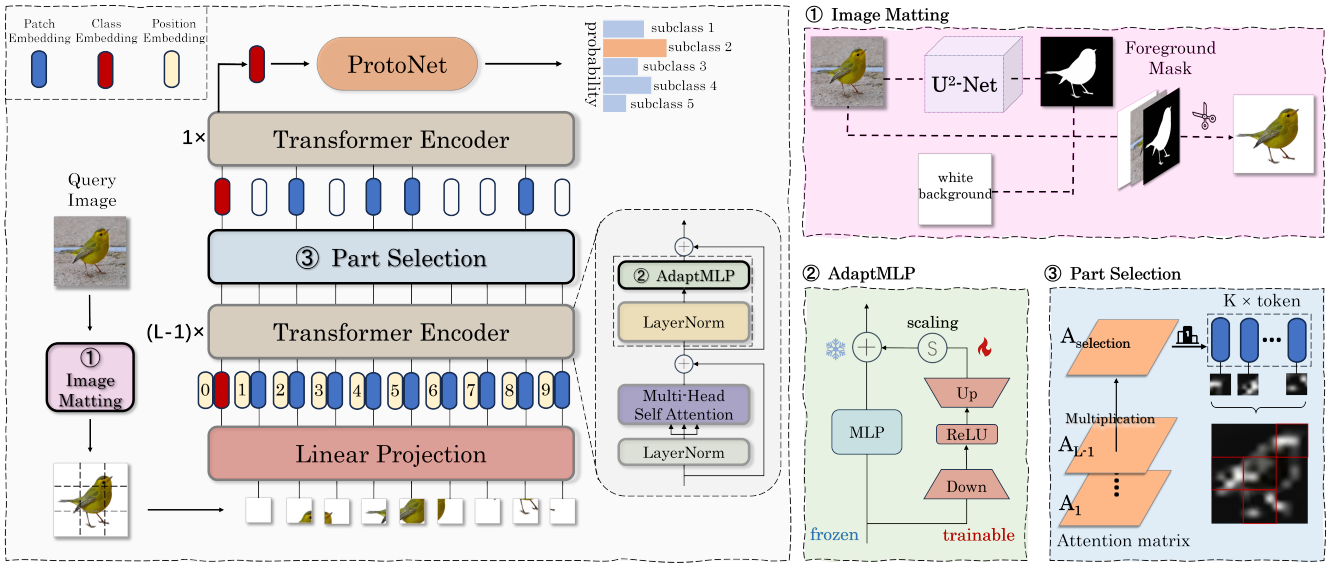

Fig. 1: The overview of our framework and three key modules: ① Image Matting, ② AdaptMLP, and ③ Part Selection.

ficient training, *i.e.*, pre-training → meta-training → fine-tuning.

C. Details of Key Modules

In this subsection, we explain the details of the three key modules in our framework.

1) Image Matting: To make the ViT focus on the salient subject of the query image that contains discriminative information, we introduce the Image Matting module to remove background noise.

As shown in Fig.1, the Image Matting module first employs a U^2-Net [29], a salient object detection model, to generate a foreground mask, as U^2-Net can autonomously identify the most visually appealing parts of an image. Assume the input query image is $I \in \mathbb{R}^{H\times W\times C}$, where (H, W) is the resolution of the image and C is the number of channels, the foreground mask $M \in \mathbb{R}^{H\times W}$ output by the U^2-Net has the following form:

$$M_{i,j} = \begin{cases} 0, & \text{if } I_{i,j} \text{ belongs to the background,} \\ 1, & \text{else.} \end{cases} \tag{1}$$

Then, the foreground of image I (denoted as $F \in \mathbb{R}^{H\times W\times C}$) can be calculated as follows:

$$F_{i,j,c} = I_{i,j,c} \cdot M_{i,j} + w \cdot (1 - M_{i,j}) \tag{2}$$

where i, j, and c represent the row, column, and channel index of a pixel, respectively, w is set to 255 so that the backgroud is filled with white.

2) AdaptMLP: We introduce the AdaptMLP module to reduce the number of trainable parameters in fine-tuning. Specifically, for each Transformer Encoder in ViT, we replace its MLP block with an AdaptMLP module.

As shown in Fig.1, AdaptMLP consists of two branches: (1) The left branch has an MLP block that is the same as that in vanilla ViT; (2) The right branch includes a down-projection which compresses the dimensionality of the input, an up-projection which restores the dimensionality, and a ReLU activation function in-between to provide non-linear transformation. Assume the output of MHSA in the l^{th} Transformer Encoder is $\mathbf{x}_l'$, the functions of these two branches can be formulated as follows:

$$\mathbf{x}_l^{left} = \text{MLP}\left(\text{LN}\left(\mathbf{x}_l^{'}\right)\right), \tag{3}$$

$$\mathbf{x}_l^{right} = \text{ReLU}\left(\text{LN}\left(\mathbf{x}_l^{'}\right) \cdot W_{\text{down}}\right) \cdot W_{\text{up}}, \tag{4}$$

where $\mathbf{x}_l^{left}$ (resp. $\mathbf{x}_l^{right}$) is the output of the left (resp. right) branch, MLP$(\cdot)$ represents the function of the MLP block, LN$(\cdot)$ denotes the layer normalization before AdaptMLP, $W_{\text{down}} \in \mathbb{R}^{d\times d_{\text{down}}}$ and $W_{\text{up}} \in \mathbb{R}^{d_{\text{down}}\times d}$ are the weight matrix of the down-projection and up-projection, respectively, $d_{\text{down}} \ll d$. Finally, the outputs of the two branches are added to obtain the output of the AdaptMLP module, as follows:

$$x_l = \mathbf{x}_l^{left} + s \cdot \mathbf{x}_l^{right}, \tag{5}$$

where s is a scaling factor.

During the fine-tuning process, all parameters in the ViT will be frozen, except those in the right branch of AdaptMLP. By this, we aim to reduce the number of trainable parameters in fine-tuning, thus mitigating the risk of overfitting under few-shot settings.

3) Part Selection: Inspired by [14], we introduce the Part Selection module to focus on discriminative details, so as to capture subtle differences between images from fine-grained subclasses.

Specifically, as shown in Fig.1, we first multiply the attention matrices from the first $L-1$ layers of Transformer Encoder one by one, as follows:

$$A_{selection} = \prod_{l=1}^{L-1} A_l, \tag{6}$$

$$A_l = [A_l^1, A_l^2, \ldots, A_l^K], \tag{7}$$

where A_l is the attention matrix from the l^{th} layer of Transformer Encoder, A_l^i $(i \in \{1, 2, \dots K\})$ represents the attention matrix of its i^{th} attention head, and K is the number of attention heads. For ease of explanation, we define $A^i_{selection}$ $(i \in \{1, 2, \dots K\})$ as follows:

$$A^i_{selection} = \prod_{l=1}^{L-1} A_l^i. \tag{8}$$

Then, for $i \in \{1, 2, \dots K\}$, we select a patch embedding with the maximum attention product in $A^i_{selection}$. Finally, the class embedding in the $(L-1)^{th}$ layer of Transformer Encoder and these selected patch embeddings are passed to the last Transformer Encoder layer.

D. Model Training

Motivated by [30], we adopt a three-stage training pipeline consisting of pre-training, meta-training, and fine-tuning, in order to thoroughly train our framework.

1) Pre-training: At this stage, we first load a ViT checkpoint pre-trained on ImageNet21K[1]. We then continue pre-training it on ImageNet1K[2] to initialize the parameters not present in the checkpoint, e.g., those of the AdaptMLP module.

2) Meta-training: At this stage, we train our framework in a meta-learning fashion using the ProtoNet [15] loss. To ensure adequate training, we first extend the training set D_{train} with datasets from the same domain as the dataset D but with non-overlapping classes to avoid data leakage. Then, we construct a support set and a query set from this extended training set as described in Section III-A.

For each class j in the support set, the ProtoNet computes a prototype of it (denoted as c_j) as follows:

$$c_j = \frac{1}{k} \sum_{i:y_i=j} f(x_i), \tag{9}$$

where k is the number of images belonging to class j in the support set, x_i and y_i are the feature vector and class label of the i^{th} image, respectively, and $f(\cdot)$ represents the function of the ViT.

Then, for an image x_q in the query set, the probability that it belongs to class j is calculated based on a softmax over its distances to all prototypes, as follows:

$$p(y = j \mid x_q) = \frac{\exp(-d(f(x_q), c_j))}{\sum_{j'} \exp(-d(f(x_q), c_{j'}))} \tag{10}$$

where $d(\cdot)$ is a distance function, which is Euclidean distance in this paper.

Finally, we can update all parameters by minimizing the loss defined as follows using gradient descent:

$$\mathcal{L} = -\log p(y = j \mid x_q), \tag{11}$$

where j is the true class of x_q.

[1]https://storage.googleapis.com/vit_models/imagenet21k/ViT-B_16.npz

[2]https://image-net.org

During the meta-training process, a support set and a query set are also constructed from the validation set D_{val} as described in Section III-A to periodically validate the performance of our framework.

3) Fine-tuning: At this stage, we fine-tune only the parameters of the AdaptMLP's right branch while the others are frozen, in the same fashion as meta-training but using different support sets and query sets. Specifically, the support set at this stage is the support set from the test dataset D_{test}, and the query set is obtained by augmenting the support set. Using the ProtoNet loss as in meta-training, the parameters of the AdaptMLP's right branch can be updated by minimizing the loss using gradient descent.

IV. Experiments

A. Experimental Setup

1) Evaluation metric & datasets: Following prior works [8], [25], [31], we use accuracy as the evaluation metric, and we use three benchmark datasets (CUB200-2011[3], Stanford Dogs[4], and Stanford Cars[5]) to evaluate our method. Note that, as for meta-training, expansion data is needed. Four datasets (NABirds[6], the Dogs-in-the-Wild[7], TsingHua Dogs[8] and CompCars[9]) are used to form the expansion data.

As for benchmark datasets, CUB-200-2011 is divided into 100 training, 50 validation and 50 testing classes; Stanford Dogs is split into 70 training, 10 validation and 40 testing classes; and Stanford Cars is divided into 120 training, 26 validation and 50 testing classes. In the meta-training phase, we employ NaBirds in combination with CUB-200-2011 training set to form the expansion data. Similarly, the Dogs-in-the-Wild and Tsinghua Dogs datasets are integrated with the Stanford Dogs training set; and the CompCars is merged with the Stanford Cars training set. It is worthy noting that we eliminate categories that overlap with those in three benchmarks from expansion datasets.

2) Implementation details: We used ViT-B/16 as the backbone (actually, a variant of ViT-B/16 since it contains newly added modules or elements, e.g., part selection module). All input images are resized to 84×84 pixels before being processed. We perform 5-way 1-shot and 5-way 5-shot experiments in which the number of query set samples is set to 10. As for fine-tuning phase, we employ standard data augmentation techniques, including random crop, horizontal flip, and color jitter. The experiments are conducted on an NVIDIA 4090 GPU. We employ the cross-entropy loss function as our optimization criterion. The model parameters

[3]https://www.vision.caltech.edu/datasets/cub_200_2011

[4]http://vision.stanford.edu/aditya86/ImageNetDogs

[5]https://ai.stanford.edu/~jkrause/cars/car_dataset.html

[6]https://dl.allaboutbirds.org/nabirds

[7]https://dl.allaboutbirds.org/nabirds

[8]https://ai.baidu.com/broad/introduction?dataset=canine

[9]https://mmlab.ie.cuhk.edu.hk/datasets/comp_cars/index.html

Method	Publication	Backbone	CUB-200-2011		Stanford Dogs		Stanford Cars	
			5-way 1-shot	5-way 5-shot	5-way 1-shot	5-way 5-shot	5-way 1-shot	5-way 5-shot
OLSA [31]	MM 2021	ResNet-12	77.77±0.44	89.87±0.24	64.15±0.49	78.28±0.32	77.03±0.46	88.85±0.46
RaPSPNet [25]	PR 2024	ResNet-256	76.64±0.86	92.73±0.46	61.46±0.82	78.73±0.62	78.84±0.69	93.79±0.49
C2-Net [8]	AAAI 2024	Conv-4	78.66±0.46	89.43±0.28	66.42±0.50	81.23±0.34	81.29±0.45	91.08±0.26
BSFA [6]	TCSVT 2023	ResNet-12	82.27±0.46	90.76±0.26	69.58±0.50	82.59±0.33	88.93±0.38	95.20±0.20
FicNet [22]	TMM 2024	ResNet-12	80.97±0.57	93.17±0.32	72.41±0.64	85.11±0.37	86.81±0.47	95.36±0.22
SRN [32]	PR 2024	ResNet-12	83.82±0.18	93.45±0.10	76.54±0.21	88.52±0.12	88.02±0.16	96.23±0.07
FRN+TDM [33]	PR 2024	ResNet-12	83.95±0.18	93.17±0.10	**77.01±0.21**	88.60±0.12	89.27±0.16	96.89±0.06
Bi-FRN [27]	AAAI 2023	ResNet-12	**85.44±0.18**	94.73±0.09	76.89±0.21	88.27±0.12	90.44±0.15	97.49±0.05
Ours	-	ViT-B/16	84.88±0.56	**95.26±0.98**	76.10±0.95	**89.38±0.83**	**95.40±0.64**	**98.32±0.88**

TABLE I: Comparison of Methods Across Datasets (5-way 5-shot and 5-way 1-shot) Across Three Benchmarks. The number in the table represents top-1 accuracy percentages for classification tasks.

are updated using the Adam optimizer. During the meta-training phase, the learning rate is set to 1×10^{-5} for the 5-way 5-shot setting and 1×10^{-4} for the 5-way 1-shot setting. In the fine-tuning phase, the learning rates are adjusted to 1×10^{-6} for the 5-way 5-shot setting and 1×10^{-5} for the 5-way 1-shot setting.

B. Comparing with State-Of-The-Arts

Table I presents the comparison results. We observe that our method achieves the best performance on the 5-way 5-shot tasks on all these three benchmark datasets, demonstrating the superiority of our proposed model. As for the 5-way 1-shot task, our method achieves the best performance on the Stanford Cars dataset, and it ranks top-2 on the CUB-200-2011 dataset, further demonstrating its competitiveness. Note that, although our method on the 5-way 1-shot tasks for Stanford Dogs is inferior to several baselines, it accuracy is near to theirs. Besides, the higher variance (± values) in our data is mainly due to the limited device memory, which restricted the number of query sets to 10. This smaller sample size compared to other studies results in larger fluctuations. In a nutshell, our method is highly competitive.

C. Ablation Study

1) Effectiveness of Image Matting: We evaluate the performance using an identical architecture but without image matting module. We can see from Table II that image matting module has a significant impact on both 1-shot and 5-shot setting across three benchmarks. The improvement is ranging from 3.8% to 6.64%. For example, on CUB-200-2011 dataset, the improvement is about 5% for both tasks. All these evidences essentially demonstrate the effectiveness of image matting module.

TABLE II: The influence of Image Matting module.

Dataset	Matting	5-way 1-shot	5-way 5-shot
CUB-200-2011	✗	79.72	90.08
	✓	**84.88**	**95.26**
Stanford Dogs	✗	72.15	85.80
	✓	**76.10**	**89.38**
Stanford Cars	✗	88.76	94.52
	✓	**95.40**	**98.32**

To get a deeper understanding, we visualize the attention results based on the overall global attention map. We can see from Fig. 2 that, background noise consistently draws unnecessary attention (see the 2nd row). In contrast, when the image matting module is applied, the model can focus more on the object itself (see the 4th row).

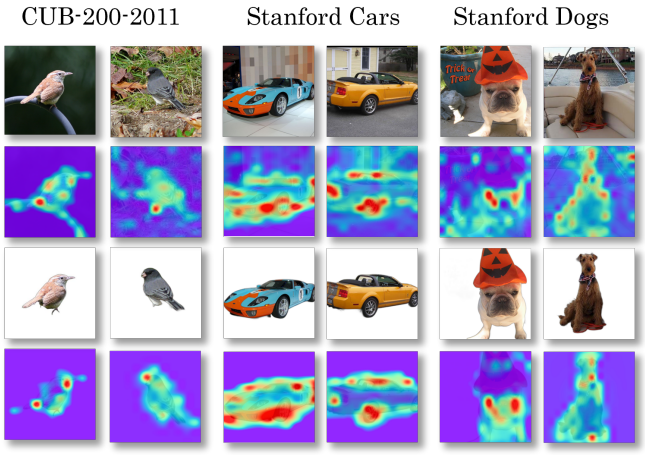

Fig. 2: Visualization results based on global attention map.

2) Effectiveness of Part Selection: To verify the effectiveness of part selection, we keep all settings unchanged while removing the part selection module. From Table III we can see that, for both 5-way 1-shot and 5-way 5-shot settings, the part selection module can help the model achieve better performance, justifying its usefulness.

TABLE III: The influence of Part Selection module.

Dataset	Part Selection	5-way 1-shot	5-way 5-shot
CUB-200-2011	✗	83.96	95.12
	✓	**84.88**	**95.26**
Stanford Dogs	✗	75.55	89.00
	✓	**76.10**	**89.38**
Stanford Cars	✗	95.04	97.52
	✓	**95.40**	**98.32**

3) Effectiveness of AdaptMLP: We conduct an ablation study on three benchmark datasets, focusing on the impact of AdaptMLP fine-tuning (i.e., freezing the other parameters). The results are shown in Table IV. When we fine-tune the whole network parameters (full tuning), the fine-tuned accuracy is lower than the accuracy when we only fine-tune the AdaptMLP right branch parameters in both 5-way 1-shot and 5-way 5-shot task. The results demonstrate the effectiveness

of AdaptMLP fine-tuning in our method, indicating that lightweight fine-tuning helps the model achieve performance in few-shot tasks.

TABLE IV: The influence of AdaptMLP module.

Dataset	Tuning Option	5-way 1-shot	5-way 5-shot
CUB-200-2011	Full Tuning	83.28	94.40
	AdaptMLP tuning	**84.88**	**95.26**
Stanford Dogs	Full Tuning	75.32	88.38
	AdaptMLP tuning	**76.10**	**89.38**
Stanford Cars	Full Tuning	95.04	97.88
	AdaptMLP tuning	**95.40**	**98.32**

4) Effectiveness of Three-stage Training: To verify the effectiveness of the three-stage training process, we conduct tests on the CUB-200-2011 dataset under the settings where pre-training, meta-training, and fine-tuning is removed, respectively. The results shown in Table V demonstrate that removing any of the training stages significantly degrades the classification performance. These evidences essentially demonstrate that the three-stage training strategy is crucial for adapting ViT to the FSFGIC task.

TABLE V: The influence of three-stage training

setting	5-way 1-shot	5-way 5-shot
without pre-training	73.44	82.50
without meta-training	48.80	70.12
without fine-tuning	80.92	91.76
with three stage	**83.96**	**95.12**

V. Conclusion

In this paper, we have proposed a new framework for few-shot fine-grained image classification. Our method uses an enhanced ViT as the backbone and adopts a three-stage training strategy. To validate the performance and its competitiveness, we have conducted extensive experiments based on three benchmark datasets. The results consistently show that the proposed method is highly competitive, comparing against the strong baselines.

References

[1] G. R. Machado, E. Silva, and R. R. Goldschmidt, "Adversarial machine learning in image classification: A survey toward the defender's perspective," *ACM Comput. Surv.*, vol. 55, no. 2, pp. 8:1–8:38, 2023.

[2] W. Ge and Y. Yu, "Borrowing treasures from the wealthy: Deep transfer learning through selective joint fine-tuning," in *CVPR*, 2017, pp. 10–19.

[3] Y. Song, T. Wang, P. Cai, S. K. Mondal, and J. P. Sahoo, "A comprehensive survey of few-shot learning: Evolution, applications, challenges, and opportunities," *ACM Comput. Surv.*, vol. 55, no. 13s, pp. 271:1–271:40, 2023.

[4] D. Lin, X. Shen, C. Lu, and J. Jia, "Deep LAC: deep localization, alignment and classification for fine-grained recognition," in *CVPR*, 2015, pp. 1666–1674.

[5] Z. Wang, S. Wang, S. Yang, H. Li, J. Li, and Z. Li, "Weakly supervised fine-grained image classification via guassian mixture model oriented discriminative learning," in *CVPR*, 2020, pp. 9746–9755.

[6] Z. Zha, H. Tang, Y. Sun, and J. Tang, "Boosting few-shot fine-grained recognition with background suppression and foreground alignment," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 33, no. 8, pp. 3947–3961, 2023.

[7] 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," *EEE Trans. Image Process.*, vol. 30, pp. 1318–1331, 2021.

[8] Z. Ma, Z. Chen, L. Zhao, Z. Zhang, X. Luo, and X. Xu, "Cross-layer and cross-sample feature optimization network for few-shot fine-grained image classification," in *AAAI*, 2024, pp. 4136–4144.

[9] F. Zhou, B. Wu, and Z. Li, "Deep meta-learning: Learning to learn in the concept space," *CoRR*, vol. abs/1802.03596, 2018.

[10] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, and et al., "An image is worth 16x16 words: Transformers for image recognition at scale," in *ICLR*, 2021.

[11] Y. Li, H. Mao, R. Girshick, and K. He, "Exploring plain vision transformer backbones for object detection," in *ECCV*, 2022, pp. 280–296.

[12] J. Dan, Y. Liu, H. Xie, J. Deng, H. Xie, X. Xie, and B. Sun, "TransFace: Calibrating Transformer Training for Face Recognition from a Data-Centric Perspective," in *ICCV*, 2023, pp. 20 585–20 596.

[13] A. Arnab, M. Dehghani, G. Heigold, C. Sun, M. Lučić, and C. Schmid, "Vivit: A video vision transformer," in *ICCV*, 2021, pp. 6816–6826.

[14] J. He, J. Chen, S. Liu, A. Kortylewski, C. Yang, Y. Bai, and C. Wang, "Transfg: A transformer architecture for fine-grained recognition," in *AAAI*, 2022, pp. 852–860.

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

[16] C. Finn, P. Abbeel, and S. Levine, "Model-agnostic meta-learning for fast adaptation of deep networks," in *ICML*, 2017, pp. 1126–1135.

[17] W. Liu, X. Chang, Y. Yan, Y. Yang, and A. G. Hauptmann, "Few-shot text and image classification via analogical transfer learning," *ACM Trans. Intell. Syst. Technol.*, vol. 9, no. 6, pp. 71:1–71:20, 2018.

[18] H. Zheng, J. Fu, Z. Zha, J. Luo, and T. Mei, "Learning rich part hierarchies with progressive attention networks for fine-grained image recognition," *IEEE Trans. Image Process.*, vol. 29, pp. 476–488, 2020.

[19] T. Lin, A. RoyChowdhury, and S. Maji, "Bilinear CNN models for fine-grained visual recognition," in *ICCV*, 2015, pp. 1449–1457.

[20] M. Engin, L. Wang, L. Zhou, and X. Liu, "Deepkspd: Learning kernel-matrix-based SPD representation for fine-grained image recognition," in *ECCV*, 2018, pp. 629–645.

[21] C. Liu, H. Xie, Z. Zha, L. Ma, L. Yu, and Y. Zhang, "Filtration and distillation: Enhancing region attention for fine-grained visual categorization," in *AAAI*, 2020, pp. 11 555–11 562.

[22] 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," *IEEE Trans. Multim.*, vol. 26, pp. 10 264–10 278, 2024.

[23] S. Tsutsui, Y. Fu, and D. J. Crandall, "Meta-reinforced synthetic data for one-shot fine-grained visual recognition," in *NeurIPS*, 2019, pp. 3057–3066.

[24] Z.-X. Ma, Z.-D. Chen, L.-J. Zhao, Z.-C. Zhang, T. Zheng, X. Luo, and X.-S. Xu, "Bi-directional task-guided network for few-shot fine-grained image classification," in *ACM MM*, 2024, p. 8277–8286.

[25] W. Zhang, Y. Zhao, Y. Gao, and C. Sun, "Re-abstraction and perturbing support pair network for few-shot fine-grained image classification," *Pattern Recognit.*, vol. 148, p. 110158, 2024.

[26] Y. Li, D. Chen, T. Tang, and X. Shen, "Htr-vt: Handwritten text recognition with vision transformer," *Pattern Recognit.*, vol. 158, p. 110967, 2025.

[27] J. Wu, D. Chang, A. Sain, X. Li, Z. Ma, J. Cao, J. Guo, and Y. Song, "Bi-directional feature reconstruction network for fine-grained few-shot image classification," in *AAAI*, 2023, pp. 2821–2829.

[28] S. Chen, C. Ge, Z. Tong, J. Wang, Y. Song, J. Wang, and P. Luo, "Adaptformer: Adapting vision transformers for scalable visual recognition," in *NIPS*, 2022.

[29] X. Qin, Z. Zhang, C. Huang, M. Dehghan, O. R. Zaiane, and M. Jagersand, "U2-net: Going deeper with nested u-structure for salient object detection," *Pattern Recognit.*, vol. 106, p. 107404, 2020.

[30] S. X. Hu, D. Li, J. Stühmer, M. Kim, and T. M. Hospedales, "Pushing the limits of simple pipelines for few-shot learning: External data and fine-tuning make a difference," in *CVPR*, 2022, pp. 9058–9067.

[31] Y. Wu, B. Zhang, G. Yu, W. Zhang, B. Wang, T. Chen, and J. Fan, "Object-aware long-short-range spatial alignment for few-shot fine-grained image classification," in *ACM MM*, 2021, pp. 107–115.

[32] X. Li, Z. Li, J. Xie, X. Yang, J. Xue, and Z. Ma, "Self-reconstruction network for fine-grained few-shot classification," *Pattern Recognit.*, vol. 153, p. 110485, 2024.

[33] X. Li, Z. Guo, R. Zhu, Z. Ma, J. Guo, and J. Xue, "A simple scheme to amplify inter-class discrepancy for improving few-shot fine-grained image classification," *Pattern Recognit.*, vol. 156, p. 110736, 2024.