Abstract—The goal of few-shot classification (FSL) is to identify unseen classes with very limited samples has attracted more and more attention. Usually, it is formulated as a metric learning problem. The core issue of few-shot classification is how to learn (1) consistent representations for images in both support and query sets and (2) effective metric learning for images between support and query sets. In this paper, we show that the two challenges can be well modeled simultaneously via a unified Query-Support Transformer (QSFormer) model. To be specific, the proposed QSFormer involves global query-support sample Transformer (sampleFormer) branch and local patch Transformer (patchFormer) learning branch. SampleFormer aims to capture the dependency of support samples and query sets for image representation. It adopts the Encoder, QS-Decoder and Cross-Attention to respectively model the Support, Query (image) representation and Metric learning for few-shot classification task. Also, as a complementary to global learning branch, we adopt a local patch Transformer to extract structural representation for each image sample by capturing the long-range dependence of local image patches. In addition, we introduce a novel Cross-scale Interactive Feature Extractor (CIFE) to extract and fuse different scale CNN features as an effective backbone module for the proposed few-shot learning method. We integrate these into a unified framework and train it in an end-to-end way. A large number of experiments are conducted on four popular datasets to validate the superiority and effectiveness of the proposed QSFormer.

Index Terms—Few-shot learning, transformer, metric learning, deep learning.

I. INTRODUCTION

CURREN T deep neural networks (DNNs) learn from large-scale training samples and achieve good performance on many tasks. However, in many scenarios, data collection and annotation is expensive and it is usually very challenging to collect enough data for training DNNs. Few-Shot Learning (FSL) is widely used in many other applications, such as synthetic aperture radar [1], [2], [3], object detection [4], [5], medical diagnosis [6] and so on. There is increasing interest in few-shot classification that use very limited support/seen samples for recognizing query/unseen classes.

Many deep learning-based methods [7], [8], [9], [10] have emerged to solve FSL problem. These methods can roughly be divided into three aspects, i.e., generation-based methods [7], [11], [12], optimization-based methods [13], [14], [15], [16], [17], [18], [19], [20], [21], [22], [23], [24], [25], [26] and metric-based methods [27], [28], [29], [30], [31]. Metric-based methods usually leverage the learned image representation and the effective metric learning techniques to help distinguish support and query samples. As we know, the core issues for metric-based few-shot classification are two aspects: 1) How to learn consistent representations for images in both support and query sets. 2) How to conduct effective metric learning for images between support and query sets.

According to our observation, existing works [9], [13], [28], [32], [33], [34] usually first employ Convolution Neural Networks (CNNs) to learn image feature representation and then use a metric function to directly compute the similarities (e.g., cosine distance, Euclidean distance) between query and support images for few-shot classification, which can achieve the good performance. However, many recent studies [35], [36] demonstrate that CNN only captures the local relations well due to its limited receptive field. To address this issue, some researchers [37], [38], [39], [40], [41] propose to combine or replace CNN with Transformer networks to model the long-range relationships of local image patches and obtain better image representation results. However, they may still produce the second-best performance for the following two reasons: 1) Existing works generally adopt Transformers (or CNN+Transformer) as the backbone network for engineering each image representation, which obviously ignores the inherent relationships among samples in query and support sets for image representation. 2) Existing works generally adopt the two-stage learning scheme, i.e., ‘representation learning + metric learning’. Although the two stages are usually learned together in an end-to-end manner, this decoupling way may lead to sub-optimal learning results.

To address these challenges, this work develops a unified Query-Support Transformer framework for few-shot learning, termed QSFormer. The core of QSFormer is our new design...
of query-support sample Transformer (named sampleFormer) module, which aims to explore the relationships of samples for coupling sample representations and metric learning of samples together in a unified module for few-shot classification. To be specific, we dexterously employ the Encoder, QS-Decoder and Cross-Attention in our sampleFormer architecture to model the Support, Query (image) representation and Metric learning in few-shot classification task, respectively. For the support branch, we represent all support images as a sequence of image tokens and feed them into the Transformer encoder to enhance the support features. For the query branch, it receives a sequence of query image tokens to learn their representations. Meanwhile, it interacts with the previous support branch via the cross-attention for modeling the similarities/affinities between query and support tokens, therefore, naturally achieving metric learning in the decoding procedure.

Based on our newly proposed sampleFormer, we further extend it by introducing two additional new modules for high-performance few-shot learning, including Cross-scale Interactive Feature Extractor (CIFE) and local patch Transformer (patchFormer) module. Specifically, as shown in Fig. 1, given the query and support images, we first use CIFE as the backbone module to extract the image features. Then, the embedded image tokens are fed into sampleFormer and get the global metrics. Meanwhile, the local/patch correspondence of query-support image pairs is also considered using the patchFormer. The global and local metrics are combined for few-shot classification task. It is noted that the whole network can be optimized in an end-to-end way.

To sum up, the key contributions of this paper are summarized as the following four points, i.e.,

- The unified Query-Support Transformer (termed QSFormer) is developed for few-shot learning, which models the representation learning and metric learning simultaneously.
- A novel Sample Transformer (sampleFormer) module is designed to capture the sample relationships in few-shot problem setting. Also, we propose a patch Transformer (patchFormer) module to help image representation learning and metric learning.
- The Cross-scale Interactive Feature Extractor is introduced for image representation by considering the interaction of different CNN levels.
- A large number of experiments prove the superiority and effectiveness of QSFormer on four popular datasets of few-shot classification task.

II. RELATED WORK

The brief review of CNN-based Few-shot Learning (FSL), Cross-attention based FSL and Transformer-based FSL are conducted as follows.

A. CNN-Based Few-Shot Learning

Currently, many FSL algorithms have appeared, which can be broadly divided into three types: generation-based methods [7], [11], optimization-based methods [8], [14], [17], [23], [25], [26], [32] and metric-based methods [9], [13], [42], [43], [44]. Of these, the metric-based method is more relevant to us, which mainly focuses on the representation learning and metric learning of samples.

Specifically, Vinyals et al. [13] propose a neural framework by combining parametric model (Memory Augment Neural Networks) and non-parametric model (Metric Learning), named as Matching Networks, for one-shot classification tasks. Snell et al. [28] propose Prototypical Networks that each class is represented by averaging its instance features learned in the representation space and validate Euclidean distance is greatly superior to commonly used Cosine similarity during metric learning phase. Sung et al. [30] propose a Relation Network (RN) for FSL problem. The RN method mainly computes the relation scores between support examples and the few examples of new classes in order to classify the examples of new classes. Oreshkin et al. [29] propose metric scaling based on two popular distance metric (i.e., cosine similarity and Euclidean distance) and introduce task representation conditioning to improve the performance of feature extractor for FSL problem. Liu et al. [45] propose a Transductive Propagation Network (TPN), which leverages graph construction to produce parameters based on each example and thus mine the manifold structure of each episode. The TPN uses the whole test set during the inference phase. Zhang et al. [9] introduce Earth Mover’s Distance (EMD), which produces the structural distance of image representations to measure the relevance of support-query images for few-shot classification. Jiang et al. [43] first obtain the multi-scale feature representations by a feature pyramid network and feed them into a multi-scale relation generation network (MRGN) for metric learning of different levels. Xie et al. [44] introduce a deep Brownian Distance Covariance approach to learn image representations and then use distance metric for classification. Previous few-shot learning works usually employ two-stage methods of representation learning and metric learning. These two stages are also conducted in an end-to-end manner. Our QSFormer is also the end-to-end learning model. Different from previous works, our method achieves representation learning and metric learning simultaneously via a single Transformer architecture.

B. Cross-Attention Based Few-Shot Learning

In recent years, there have been some cross-attention based approaches [42], [46], [47], [48] to solve few-shot learning problem. To be specific, CrossTransformer [47] proposes to produce the enhanced prototype vector of support image by modeling the local/pixel relationships between query-support image pairs and then using it for the final similarity metric. CAN [42] represents a cross-attention module modeling the local/pixel relevance between query and support-class prototype to separately generate the query and support-class prototype feature representation for few-shot classification. PARN [46] concatenates the outputs of self-correlation attention module (SCA) and cross-correlation attention module (CCA) to generate the final relation score between query-support images, where CCA models the local relationships
Fig. 1. An overview of QSFormer framework. The proposed QSFormer mainly consists of Cross-scale Interactive Feature Extractor (CIFE), Sample Transformer Module (sampleFormer), Patch Transformer Module (patchFormer), Metric Learning and Few-shot Classification. Concretely, a shared CIFE extracts the support feature embeddings $\tilde{F}_s$ and query feature embeddings $\tilde{F}_q$, and its illustration can be seen in Fig. 2. The sampleFormer module performs sample representation learning and global metric learning simultaneously. As a complementary to it, we add patchFormer module to learn the context relationship of image patches and conduct local metric learning. The more details of the proposed QSFormer can be seen in Section III.

between query and support images. Xiao et al. [48] introduce a cross-attention module to inject semantic/label information into visual/image patch sequence.

Different from these works, we introduce Sample Transformer Module based on cross-attention mechanism, where cross-attention mechanism is designed to implement the global similarity metric between query-support image pairs and the image-level interaction between query and support image features simultaneously.

C. Transformer-Based Few-Shot Learning

Transformer [49] has universal modeling capability because of its core module self-attention learning mechanism. The modeling of pixel-pixel, object-pixel and object-object relationships is indispensable for computer vision. Thus, in recent years, Transformer has been employed by a large number of researchers for various visual tasks, including object tracking [50], [51], object detection [52], [53], [54], [55], object re-identification [56], [57], [58], multi-label classification [59], [60], [61], Medical Image Segmentation [62], [63], and so on.

For few-shot learning tasks, some works [37], [38], [41], [47], [64], [65], [66] demonstrate that Transformer architecture is also promising. For instance, Ye et al. [64] develop a Few-Shot Embedding Adaptation Transformer (FEAT) to conduct the set-to-set transformation and thus make instance embedding task-specific for FSL problem. Liu et al. [65] introduce a Universal Representation Transformer (URT) by combining feature representations from multiple domains together for multi-domain few-shot classification. He et al. [37] propose a CNN-free framework, called Hierarchically Cascaded Transformers by attribute surrogates learning and spectral tokens pooling. Dong et al. [38] propose a few-shot training framework (i.e., self-promoted supervision) to alleviate the performance degradation of few-shot classification when replacing commonly CNN feature extractor with ViT model. Zhmoginov et al. [41] introduce a transformer-based model, called HyperTransformer (HT), which encodes task-dependent variations in the weights of a small CNN model for FSL problem.

These works mainly employ Transformer architecture for representation learning, i.e., feature enhancement. Differently, in our work, we develop a Query-Support Transformer (QSFormer) for metric learning, which not only considers global metric learning at the sample level but also local metric learning at the patch level.

III. THE PROPOSED METHOD

The purpose of few-shot classification is to recognize the unseen samples when a few samples can be available. Many recent approaches [9], [42], [67], [68] indicate that the episode
mechanism provides an effective way for few-shot classification task. Therefore, we adopt episode mechanism in both training and testing phases. Formally, let $D_{\text{train}}$, $D_{\text{val}}$ and $D_{\text{test}}$ respectively represent meta-training set, meta-validation set and meta-testing set, where $D_{\text{train}} \cap D_{\text{val}} \cap D_{\text{test}} = \emptyset$. Taking $C$-way $K$-shot few-shot classification task as an example, it contains support set $X_s = \{(X_s^l, Y_s^l)\}_{l=1}^{n_s}$ and query set $X_q = \{(X_q^l, Y_q^l)\}_{l=1}^{n_q}$ in each episode. Concretely, we randomly select $C$ classes from meta-training set and $K$ labeled samples of each class to form support set $X_s$, i.e., $n_s = C \times K$. Meanwhile, we randomly sample $q$ samples of each class to form query set $X_q$, i.e., $n_q = C \times q$.

As shown in Fig. 1, we propose a novel Query-Support Transformer (QSFormer) framework for few-shot learning, which contains the following four parts:

- **Cross-Scale Interactive Feature Extractor (CIFE):** we introduce a cross-scale interactive feature extractor as backbone network to obtain the spatial enhanced support/query CNN feature representations.
- **Sample Transformer Module:** we introduce a query-support sample Transformer (sampleFormer) module to couple image sample representation and global metric learning of samples together for few-shot learning.
- **Patch Transformer Module:** we also propose a patch Transformer (patchFormer) module to model the context correlation of patches in each image sample to conduct the local metric learning between query-support sample pairs.
- **Metric Learning and Few-shot Classification:** we acquire the final metric by combining global metric obtained via sampleFormer and local metric obtained via patchFormer together and finally achieve few-shot classification.

Below, we introduce the details of these modules.

A. Cross-Scale Interactive Feature Extractor

Traditional CNN-based backbones generally lack considering the interaction among different scales. Some previous works [69], [70] demonstrate that exploiting the interaction information of different scales of CNN features can further enrich the feature representation. Inspired by this, we introduce a novel Cross-scale Interactive Feature Extractor (CIFE) as our backbone module which aims to employ a cross-attention module to capture the interaction of different scales. CIFE obtains the ego-context CNN feature representations for support and query samples.

As shown in Fig. 2, taking the support image set $X_s = \{X^s_1, X^s_2, \ldots, X^s_{n_s}\}$ as inputs, we first employ the pre-trained ResNet-12 for generating the initial multi-scale image feature representations $F^s_\ell \in \mathbb{R}^{n_s \times c \times h_\ell \times w_\ell}$, $\ell \in [1, 2, 3, 4]$, where $n_s$ represents the number of support samples in each episode and $c_l$, $h_l$ and $w_l$ refer to the number of channel, height and width of support feature map of $\ell$-th level, respectively. Next, we reshape the initial multi-scale support features $F^s_\ell$ to token sequence $T^s_\ell \in \mathbb{R}^{n_s \times h_l w_l c_l}$ and employ a convolutional layer and a ViT [35] block to obtain the enhanced feature representations $\tilde{T}^s_\ell \in \mathbb{R}^{n_s \times h_{l+1} w_{l+1} c_{l+1}}$ of the $(l+1)$-th level. Whereafter, we leverage cross-attention mechanism to achieve the feature interaction of different levels, which can be denoted as $\tilde{T}^s_\ell = LN(Attn_{\ell-1}(Q^s_\ell, K^{l-1}_\ell)\ Y^{l-1}_\ell)$, where $LN(\cdot)$ refers to layer normalization and $Attn_{\ell-1}(Q^s_\ell, K^{l-1}_\ell)$ can be formulated as

$$Attn_{\ell-1}(Q^s_\ell, K^{l-1}_\ell) = \text{Softmax}\left(\frac{Q^s_\ell \cdot \sqrt{c}}{\sqrt{d}}\right)$$

where $Q^s_\ell$, $K^{l-1}_\ell$ and $Y^{l-1}_\ell$ represent the Query, Key and Value that use three different linear projections on $T^s_\ell$ and $\tilde{T}^s_{l-1}$, respectively. $\text{Softmax}(\cdot)$ is a row normalization operation. Finally, we further employ residual connection and Feed-Forward Network (FFN) by following [49] to obtain the spatial enhanced feature representations for support samples as $F^s = \{F^s_1, F^s_2, \ldots, F^s_{n_s}\} \in \mathbb{R}^{n_s \times c \times h \times w}$. Similarly, we obtain the spatial enhanced features for query samples as $F^q = \{F^q_1, F^q_2, \ldots, F^q_{n_q}\} \in \mathbb{R}^{n_q \times c \times h \times w}$. The Feed-Forward Network (FFN) consists of two fully-connected layers with a GeLU activation between two layers (it will not be repeated unless necessary). The parameters of CIFE are shared for support and query branches. In practice, we empirically set $c = 640$ and $h = w = 5$.

B. Sample Transformer Module

To achieve both image sample representation and metric learning of samples in a unified module, we design a novel query-support sample Transformer module, named sampleFormer, as shown in Fig. 1. The proposed sampleFormer mainly consists of Encoder and QS-Decoder and its details are as follows.

1) Encoder: The purpose of the Encoder is to mine the relationships of samples in support set to obtain better support feature representations. To this end, based on the aforementioned support features $F^s \in \mathbb{R}^{n_s \times c \times h \times w}$, we first introduce image tokenize, which utilizes a global average pooling and reshape operation to gain the token sequence $H^s = \{H^s_1, H^s_2, \ldots, H^s_{n_s}\} \in \mathbb{R}^{n_s \times c}$ of support samples, where each token $H^s_\ell$ denotes a support sample. As shown in Fig. 1, we can see that the main component of encoder is attention mechanism, whose inputs are Query $Q^s \in \mathbb{R}^{n_s \times c}$, Key $K^s \in \mathbb{R}^{n_s \times c}$, and Value $Y^s \in \mathbb{R}^{n_s \times c}$ obtained by conducting three linear projections on $H^s$ respectively. Next, it employs dot-product operation to obtain a correlation/affinity
matrix $\text{Attn}_{s \rightarrow s}(Q^s, K^s)$ of different support samples as

$$\text{Attn}_{s \rightarrow s}(Q^s, K^s) = \text{Softmax} \left( \frac{Q^s (K^s)^T}{\sqrt{c}} \right)$$ (2)

where $c$ denotes the dimension of support features. $\text{Softmax}(\cdot)$ is a row normalization operation. It learns the representations for support samples by conducting the message passing operation as

$$\tilde{H}^s = \text{LN}(H^s + \text{Attn}_{s \rightarrow s}(Q^s, K^s) V^s)$$ (3)

where $\text{LN}(\cdot)$ refers to layer normalization. Besides, we add Feed-Forward Network (FFN) [35] and residual operation to obtain the final support sample representations as,

$$\tilde{H}^s = \text{LN}(\tilde{H}^s + \text{FFN}(\tilde{H}^s))$$ (4)

where $\tilde{H}^s = \{H^s_1, H^s_2, \cdots, H^s_n\} \in \mathbb{R}^{n_s \times c}$. $n_s$ denotes the number of support samples and $c$ refers to the dimension of feature. FFN consists of two linear transformations with a GeLU activation.

2) QS-Decoder: The QS-Decoder aims to explore the dependence of samples in query set to learn the representations for query samples and also mines the intrinsic metrics of samples in support and query sets. To be specific, it takes the aforementioned encoded support features $\tilde{H}^s \in \mathbb{R}^{n_s \times c}$ and query feature embeddings $\tilde{F}^q \in \mathbb{R}^{n_q \times c \times h \times w}$ as its inputs. The image tokenize is applied on $\tilde{F}^q$ to obtain the initial query token sequence $H^q = \{H^q_1, H^q_2, \cdots, H^q_{n_q}\} \in \mathbb{R}^{n_q \times c}$, where each token $H^q_j$ represents a query image sample. Similar to the Encoder branch, we first leverage self-attention message passing mechanism to model the relationships among query samples and learn representations for query samples as

$$\text{Attn}_{q \rightarrow q}(Q^q, K^q) = \text{Softmax} \left( \frac{Q^q (K^q)^T}{\sqrt{c}} \right)$$ (5)

$$\tilde{H}^q = \text{LN}(H^q + \text{Attn}_{q \rightarrow q}(Q^q, K^q) V^q)$$ (6)

where $\text{LN}(\cdot)$ denotes layer normalization. $\text{Softmax}(\cdot)$ is a row normalization operation.

Afterward, based on the support features $\tilde{H}^s$ and query features $\tilde{H}^q$, we employ a cross-attention mechanism to explore the relationships between support and query samples for query sample representations. Specifically, it first computes the cross-affinities between support and query samples as follows

$$\text{Attn}_{q \rightarrow s}(Q^q, K^s) = \text{Softmax} \left( \frac{Q^q (K^s)^T}{\sqrt{c}} \right)$$ (7)

Then, it learns query sample representations by aggregating the information from support samples as follows

$$\tilde{H}^q = \tilde{H}^q + \text{LN}(\text{Attn}_{q \rightarrow s}(Q^q, K^s) V^q)$$ (8)

where $\tilde{H}^q \in \mathbb{R}^{n_q \times c}$ and $\text{LN}(\cdot)$ denotes layer normalization. $Q^q \in \mathbb{R}^{n_q \times c}$ is computed by conducting a linear projection on $\tilde{H}^q$. $K^s \in \mathbb{R}^{n_s \times c}$ and $V^q \in \mathbb{R}^{n_q \times c}$ are obtained by conducting two different linear projections on $\tilde{H}^q$, respectively.

3) Remark: The above cross-affinities $\text{Attn}_{q \rightarrow s}(Q^q, K^s)$ naturally reflect the similarities/affinities between support and query samples. In our work, we regard them as global metric $m_g$ for all support and query samples, i.e.,

$$m_g(X^s, X^q) = \text{Attn}_{q \rightarrow s}(Q^q, K^s)$$ (9)

where $m_g(X^s, X^q)$ contains the similarities for all query-support sample pairs in each episode. For convenience, in the following, we also use $m_g(X^s, X^q)$ to denote the metric between image $X^s$ and $X^q$, where $X^s \in X^s$, $X^q \in X^q$. We can utilize $m_g(X^s, X^q)$ for query sample classification, as discussed in the following Section Metric Learning and Few-shot Classification. Therefore, we can note that both query/support sample representation and metric learning in few-shot learning task are conducted simultaneously in our sampleFormer architecture. This is one main aspect of the proposed sampleFormer module.

C. Patch Transformer Module

As a complementary to the above sampleFormer branch, we also develop a query-support Patch Transformer Module (patchFormer) to capture the more visual content of each image sample for local metric. As shown in Fig. 1, patchFormer mainly consists of multi-head self-attention (MSA) and residual connection. Here, we omit Feed-Forward Network used in regular Transformer [35] for simplicity consideration. The parameters of MSA are shared on both support and query branches.

Concretely, we first input a support sample $X^s$ and a query sample $X^q$ into the above Cross-scale Interactive Feature Extractor (CIFE) to generate their feature embedding $\tilde{F}^s \in \mathbb{R}^{c \times h \times w}$ and $\tilde{F}^q \in \mathbb{R}^{c \times h \times w}$, followed by the patch tokenize with Partition Strategy to obtain the initial 2D patch token sequence for each support and query image. The Partition Strategy (PS) denotes reshape operation to generate the initial 2D patch sequence for both support and query images. Based on the support and the query feature maps learned by CIFE, we use the reshape operation to reshape the support/query feature into the 2D support/query patch sequence $P^s = \{p_{s1}, p_{s2}, \cdots, p_{s_{n_{ps}}} \} \in \mathbb{R}^{h_{w} \times c}$ / $P^q = \{p_{q1}, p_{q2}, \cdots, p_{q_{n_{pq}}} \} \in \mathbb{R}^{h_{w} \times c}$. Then, we employ multi-head self-attention (MSA) [49] with shared weights and residual operation to transform the support and query image patch features as

$$\tilde{P}^s = \text{LN}(P^s + \text{MSA}(P^s))$$

$$\tilde{P}^q = \text{LN}(P^q + \text{MSA}(P^q))$$ (10)

where $\text{LN}(\cdot)$ denotes layer normalization.

Based on the above patch representations $\tilde{P}^s = \{\tilde{p}_{s1}, \tilde{p}_{s2}, \cdots, \tilde{p}_{s_{n_{ps}}} \}$ and $\tilde{P}^q = \{\tilde{p}_{q1}, \tilde{p}_{q2}, \cdots, \tilde{p}_{q_{n_{pq}}} \}$, we then adopt the Earth Mover’s Distance (EMD) [9], [71] to compute their structural similarity. It first computes the distance between all patch pairs ($\tilde{p}_{s1}$, $\tilde{p}_{q1}$) and then acquires the optimal matching between patches of two images that have the minimum distance cost. Finally, it returns the image-level metric by aggregating the metrics of all matched patch pairs. In this paper, we denote this metric as local metric between support
sample \(X^s\) and query sample \(X^q\), i.e.,

\[
m_1(X^s, X^q) = \sum_{j=1}^{hw} \sum_{i=1}^{hw} EMD(\tilde{p}_j^i, \tilde{p}_j^{i*})
\]

(11)

**D. Metric Learning and Few-Shot Classification**

Given the support samples \((X^s, Y^s) \in X^s\) with known labels and input query sample \(X^q \in X^q\), few-shot classification aims to judge the category label of the query sample. To achieve this task, we first obtain the sample-based global metric \(m_q(X^s, X^q)\) via Equ. (9) and patch-based local metric \(m_1(X^s, X^q)\) via Equ. (11) respectively and combine them together to help us get the final metric/similarity score between \(X^s\) and \(X^q\) as

\[
m(X^s, X^q) = \lambda m_q(X^s, X^q) + (1 - \lambda) m_1(X^s, X^q)
\]

(12)

where \(\lambda \in (0, 1)\) is a tradeoff parameter.

Then, we can conduct few-shot classification by using the nearest neighbor classification strategy, i.e., the label of query \(L\) is determined by the label \(Y^s\) of the support sample \(X^s\) that is most similar with query \(X^q\), as used in previous works [9], [13].

**Loss Function:** In the training phase, we employ two loss functions for the proposed QSFormer. First, for the sample-Former module, we specifically introduce a contrastive loss as suggested in work [72], [73], which encourages the positive query-support sample pairs with same label (i.e., \(Y^s = Y^q\)) to be closing and the negative query-support sample pairs with different labels (i.e., \(Y^s \neq Y^q\)) are far away in each episode. This loss function can be written as follows,

\[
L_{cl} = -\log \frac{\sum_{y^s \neq y^q} e^{m_q(X^s, X^q)}}{\sum_{y^s = y^q} e^{m_q(X^s, X^q)} + \sum_{y^s \neq y^q} e^{m_q(X^s, X^q)}}
\]

(13)

where \(m_q(X^s, X^q)\) is the global metric between query \(X^q\) and support sample \(X^s\). The whole QSFormer network is the end-to-end trained by minimizing the Cross-Entropy (CE) loss function \(L_{ce}\) [9]. Thus, we formulate the total loss function as

\[
L_{total} = \alpha L_{ce}(\hat{Y}^q, Y^q) + (1 - \alpha) L_{cl}^q
\]

(14)

where \(\hat{Y}^q\) is the label prediction obtained by our method and \(Y^q\) denotes the corresponding ground-truth label. \(\alpha \in (0, 1)\) is the balanced hyper-parameter.

In a word, Eq. (13) denotes the regular contrastive loss function which aims to minimize the distance between positive query-support sample pairs while maximizing the distance between negative query-support sample pairs. It is defined based on the distance metric function Eq. (12). The Eq. (14) denotes the whole loss function which involves both contrastive loss and label prediction loss \(L_{ce}\). The contrastive loss is designed for compact feature learning while the cross-entropy loss is employed for the final label prediction. The prediction loss \(L_{ce}\) is defined as the cross-entropy between predicted label \(\hat{Y}^q\) and ground-truth label \(Y^q\).

**IV. EXPERIMENTS**

**A. Datasets and Evaluation Metric**

In order to prove the effectiveness of the proposed QSFormer, we perform the abundant experiments on four publicly popular datasets for few-shot classification task, such as miniImageNet [13], tieredImageNet [80], Fewshot-CIFAR100 [29] and Caltech-UCSD Birds-200-2011 [81]. We also conduct cross-domain experiments for evaluating the domain transfer capability of the proposed QSFormer. The recognition accuracy is adopted as the evaluation metric for our experiments. More details of datasets description are as follow.

**miniImageNet:** This dataset is a sub-dataset of ImageNet [82]. It contains a total of 100 classes with 600 samples in each class. As suggested in work [83], we divide these classes into training set (64 classes), validation set (16 classes) and testing set (20 classes).

**tieredImageNet:** It is a sub-dataset of ImageNet [82] and contains 608 classes from 34 super-classes, with a total of 779,165 samples. Following [80], we split 351 classes (20 super-classes) from 608 classes for meta-training phase, the 97 classes (6 super-classes) from 608 classes for meta-validation phase and the rest (8 super-classes) of 608 classes for meta-testing phase.

**FC100:** Fewshot-CIFAR100 is built upon the CIFAR100 dataset for few-shot classification task, named FC100 for short hereafter. The FC100 dataset contains a total of 60,000 images from 100 classes. To reduce the information overlap, we group the 100 classes into 20 super-classes by following work [29]. Then, we divide these super-classes into training set, validation set and testing set, which contains 12, 4 and 4 super-classes respectively.

**Caltech-UCSD Birds-200-2011:** This dataset is an extended vision of CUB-200 dataset. It’s termed CUB for short hereafter. CUB is originally presented in the fine-grained bird classification task. It contains the total of 11,788 images from 200 classes. As suggested by [64], we divide 100 classes from 200 classes for meta-training, 50 classes from 200 classes for meta-validation and the rest (50 classes) of 200 classes for meta-testing.

**miniImageNet → CUB:** By following [32], we train a model on miniImageNet [13] dataset and evaluate on the CUB [81] dataset to verify the transfer ability of model. Specifically, in this experimental setting, we use all 100 classes of miniImageNet with 600 samples per class during meta-training phase and 50 classes of CUB dataset during meta-testing phase.

**B. Implementation Details**

We adopt the ResNet-12 [9], [32] with fully connected layers removed is adopted as the backbone module for a fair comparison. It is firstly pre-trained from scratch and then use the episodic training based on meta-learning framework by following works [9], [33]. We conduct experiments on 5-way 1-shot or 5-way 5-shot classification task, respectively. For each 5-way 1-shot task, it consisted of 5 × 1 = 5 support images and 5 × 15 = 75 query images, i.e., we randomly select 5 classes with 1 support image per class and 15 query images.
TABLE I
5-WAY RESULT COMPARISON OF OURS AND SOTA METHODS ON MINIIMAGE-NET AND TIEREDIMG-ENET DATASETS. MOST RESULTS
ARE FROM [9] OR THE ORIGINAL PAPERS. THE 1st, 2nd AND 3rd ARE RESPECTIVELY IN RED, BLUE AND GREEN.
* DENOTES THIS METHOD IS REPRODUCED WITH OUR DATA SETTINGS

| Method          | Backbone      | miniImageNet | 5-shot | miniImageNet | 5-shot |
|-----------------|---------------|--------------|--------|--------------|--------|
|                 |               | 1-shot       |        | 1-shot       |        |
| DHL [74]        | Conv4         | 61.99 ± 0.05 | 57.82 ± 0.05 | 73.62 ± 0.05 |
| cosine classifier [32] | ResNet12 | 55.43 ± 0.05 | 77.18 ± 0.05 | 82.37 ± 0.05 |
| TADAM [29]      | ResNet12 | 58.50 ± 0.05 | 76.70 ± 0.05 | 83.76 ± 0.05 |
| ECM [75]        | ResNet12 | 59.00 ± 0.05 | 77.46 ± 0.05 | 81.97 ± 0.05 |
| TPN [45]        | ResNet12 | 59.46 ± 0.05 | 75.65 ± 0.05 | 83.70 ± 0.05 |
| ProtoNet [28]   | ResNet12 | 60.37 ± 0.05 | 78.02 ± 0.05 | 83.40 ± 0.05 |
| MTL [73]        | ResNet12 | 61.20 ± 0.05 | 75.50 ± 0.05 | 84.10 ± 0.05 |
| DC [77]         | ResNet12 | 62.53 ± 0.05 | 79.77 ± 0.05 | 85.62 ± 0.05 |
| MetaOptNet [78] | ResNet12 | 62.64 ± 0.05 | 78.63 ± 0.05 | 81.56 ± 0.05 |
| MatchNet [13]   | ResNet12 | 63.80 ± 0.05 | 79.90 ± 0.05 | 80.60 ± 0.05 |
| Meta-Baseline [33] | ResNet12 | 63.17 ± 0.05 | 79.26 ± 0.05 | 83.74 ± 0.05 |
| CAN [42]        | ResNet12 | 63.83 ± 0.05 | 79.44 ± 0.05 | 84.23 ± 0.05 |
| PPA [20]        | WRN-28-10 | 59.60 ± 0.05 | 73.74 ± 0.05 | 83.40 ± 0.05 |
| wDAE-GNN [79]   | WRN-28-10 | 61.07 ± 0.05 | 76.75 ± 0.05 | 83.09 ± 0.05 |
| LEO [7]         | WRN-28-10 | 61.76 ± 0.05 | 77.59 ± 0.05 | 81.44 ± 0.05 |
| FEAT [64]*      | ResNet24 | 62.96 ± 0.05 | 78.49 ± 0.05 | 85.43 ± 0.05 |
| HT [41]         | Transformer | 54.10 ± 0.05 | 68.50 ± 0.05 | 73.30 ± 0.05 |
| DeepEMD [9]*    | ResNet12 | 65.43 ± 0.05 | 79.28 ± 0.05 | 84.06 ± 0.05 |
| DeepBDC [44]*   | ResNet12 | 60.76 ± 0.05 | 78.25 ± 0.05 | 81.57 ± 0.05 |
| SUN [38]        | Transformer | 67.80 ± 0.05 | 83.23 ± 0.05 | 86.74 ± 0.05 |
| HCTransformers [37] | Transformer | 74.62 ± 0.05 | 89.19 ± 0.05 | 91.72 ± 0.05 |
| QSFormer (Ours) | ResNet12 | 65.24 ± 0.05 | 79.90 ± 0.05 | 85.43 ± 0.05 |

\* denotes the hybrid network of CNN and Transformer.

per class, respectively. Similarly, for each 5-way 5-shot task, it contains 5 × 5 = 25 support images and 5 × 15 = 75 query images. We empirically conduct the feature interaction of the last two levels in CIFE to obtain the enhanced sample features.

During meta-training, we use Stochastic Gradient Descent (SGD) [84] as the optimizer of the whole network with a momentum of 0.9. Its weight decay is set to 5e-4.

Taking the miniImageNet dataset as an example, we use Sample Transformer Module (sampleFormer) composed of a stack of N = 3 identical layers for global metric. The multi-head self-attention (MSA) of Cross-scale Interactive Feature Extractor (CIFE) and Patch Transformer Module (patchFormer) consists of 10 and 8 heads, respectively. The dropout rate is respectively set to 0.5, 0.5, 0.5, and 0.1 for CIFE, the encoder of sampleFormer, the QS-Decoder of sampleFormer and patchFormer. We set the balanced parameter \( \alpha \) between contrastive loss and Cross-Entropy loss to 0.7. The balanced parameter \( \lambda \) between global metric and local metric is set to 0.1. The learning rate (lr) is dropped by multiplying 0.9 for every 10 epochs. The whole network is trained for 100 epochs on miniImageNet. Following [9] and [44], we use

\[ \text{Authorized licensed use limited to the terms of the applicable license agreement with IEEE. Restrictions apply.} \]
the model with the best accuracy on meta-validation set to evaluate on meta-testing set for obtaining the final performance of the proposed QSFormer. More hyper-parameter settings of the proposed QSFormer on four datasets can be found in Table V. We randomly sample 50/1000/5000 episodes from the training/validation/testing set on four public datasets in all our experiments.

During the inference, the core of global sampleFormer branch is to calculate the attention between one (or several) query and multiple support samples. It is suitable to address both inductive and transductive settings. For transductive setting, our proposed sampleFormer needs to calculate the attention among both a batch of query samples and associated multiple support samples. For an inductive setting, sampleFormer needs to calculate the attention between one query and associated multiple support samples. In this case, the self-attention is not needed and the core cross-attention is reasonable in the decoder module. We compute the mean of accuracy and the corresponding 95% confidence interval to obtain the final performances of four datasets. Our proposed method is implemented by using Python on a server with a single 11G NVIDIA 2080Ti GPU.

C. Comparison With State-of-the-Art Methods

As shown in Table I, Table II and Table III, we show the comparison results of the proposed QSFormer and other state-of-the-art (SOTA) approaches on miniImageNet [13], tieredImageNet [80], FC100 [29] and CUB [81] datasets.

In Table I, we find that, 1) For the Fully-Transformer based SUN [38] and HTC Transformers [37] methods, which employ the self-supervised learning (data augmentation, knowledge distillation, clustering, etc) to enhance the representation of samples and thus generally obtain the better performance than other pure supervised methods. 2) the proposed QSFormer beats many other SOTA models on the miniImageNet dataset. For example, QSFormer respectively exceeds the transformer-based HT [41] method by +11.14% and +11.46% on the 1-shot and 5-shot tasks. For the attention mechanism based Cross Attention Network (CAN) [42], our model also outperforms it on the 1-shot/5-shot task by +1.39/+0.52%. Compared with FETA [64] that is also developed based on

| Hyper-parameters | miniImageNet | tieredImageNet | Datasets | FC100 | CUB | minImageNet → CUB |
|------------------|-------------|---------------|----------|-------|----|------------------|
| Optimizer        | SGD         | SGD           | SGD      | SGD   | SGD| SGD              |
| Initial LR       | 5e-4        | 5e-4          | 1e-4     | 5e-4  | 5e-4| 5e-4             |
| Steps of LR decay| 10          | 10            | 10       | 10    | 10 | 10               |
| Coefficient of LR decay | 0.9 | 0.5 | 0.9 | 0.95 | 0.9 | 0.9             |
| N                | 3           | 3             | 4        | 2     | 3  | 3                |
| Number of Head   | 10.8        | 8.8           | 8.1      | 8.1   | 10.8 | 10.8             |
| dropout rates    | 0.5/0.5/0.5/0.1 | 0.5/0.5/0.5/0.1 | 0.5/0.5/0.5/0.1 | 0.5/0.5/0.5/0.1 | 0.5/0.5/0.5/0.1 |
| α                | 0.7         | 0.5           | 0.5      | 0.5   | 0.5 | 0.5              |
| λ                | 0.1         | 0.1           | 0.4      | 0.3   | 0.1 | 0.1              |
| Epochs           | 100         | 100           | 50       | 150   | 100 | 100              |

TABLE VI

| Variant | Feature Extractor | FC100 | CUB |
|---------|-------------------|-------|-----|
| 1       | CIFE(ResNet12+Transformer) | 37.09 ± 0.26 | 47.11 ± 0.29 |
| 2       | Fully-Transformer | 36.16 ± 0.26 | 32.57 ± 0.24 |

ResNet-24 and Transformer, the proposed QSFormer has better results. Also, we can clearly find the proposed QSFormer achieves the better performance on the tieredImageNet dataset, i.e., 72.47±0.31 and 85.43±0.22 in the 1-shot and 5-shot tasks. It exceeds the Cross Attention Network (CAN) [42] by +2.58 and +1.2 points in 1-shot and 5-shot FSL tasks.

We can also get the similar conclusions from the experimental results of Fewshot-CIFAR100 [29] and CUB [81] datasets as shown in Table II and Table III. For example, QSFormer outperforms FEAT [64] method consisting of ResNet-12 and Transformer architecture on the FC100 dataset by +4.23% and +5.21% on the 5-way 1-shot and 5-way 5-shot FSL tasks, respectively. On the CUB dataset, compared to DeepEMD [9] which uses Earth Mover’s Distance as a distance metric, QSFormer improves +4.73 and +0.17 points in 5-way 1-shot and 5-way 5-shot FSL tasks.

In a word, the proposed QSFormer attains SOTA performance on multiple FSL datasets. These fully demonstrate the effectiveness and advantages of our proposed QSFormer model.

D. Ablation Study

To better understand the effectiveness of our QSFormer, this section conducts extensive ablation studies, including feature extractor analysis, component analysis, combination manners analysis of patchFormer and sampleFormer, similarity metric analysis, cross-domain analysis, loss function analysis, complexity, etc.

1) Feature Extractor Analysis: To intuitively compare the feature extraction ability of our proposed Cross-scale Interactive Feature Extractor (CIFE) and Full-Transformer model, we adopt the Fully-Transformer model [5], [35] to replace the proposed CIFE in the QSFormer as Variant #2 for comparative experiments. Due to the limitations of experimental data and hardware resources, we directly conduct meta training for Variant #2. For a fair comparison, we also remove the pre-training model for QSFormer based on CIFE (i.e., Variant #1). The experimental results of the 5way-1shot task on the FC100 [29] and CUB [81] datasets are shown in Table VI. We can observe
TABLE VII

| #  | Baseline | CIPE   | sampleFormer | patchFormer | minImageNet | tieredImageNet | FC100 | CUB   |
|----|----------|--------|--------------|-------------|-------------|----------------|-------|-------|
| 1  | ✓        | ✓      |              |             | 59.64 ± 0.27| 55.87 ± 0.31 | 39.47 ± 0.23| 62.09 ± 0.29|
| 2  | ✓        | ✓      | ✓            |             | 61.15 ± 0.28| 70.73 ± 0.32 | 41.54 ± 0.25| 65.95 ± 0.30|
| 3  | ✓        | ✓      | ✓            | ✓           | 63.97 ± 0.28| 71.64 ± 0.32 | 45.46 ± 0.26| 72.93 ± 0.29|
| 4  | ✓        | ✓      | ✓            | ✓           | 63.05 ± 0.29| 70.55 ± 0.32 | 45.41 ± 0.26| 72.20 ± 0.29|
| 5  | ✓        | ✓      | ✓            | ✓           | 65.24 ± 0.28| 72.47 ± 0.31 | 46.51 ± 0.26| 75.44 ± 0.29|

TABLE VIII

Performance Comparison of the Classical Methods Based on Different Metric Learning. * Denotes the Comparison Methods Is Reproduced With Our Data Setting. The Bold Black Represents The Best Results

| Methods          | Metric | minImageNet | tieredImageNet | FC100 | CUB   |
|------------------|--------|-------------|----------------|-------|-------|
| cosine classifier [32] | Cosine | 55.43 ± 0.81| 61.49 ± 0.91 | 38.47 ± 0.70 | 67.30 ± 0.86 |
| MatchNet [13]    | Cosine | 63.08 ± 0.80| 68.50 ± 0.92 | 43.88 ± 0.75 | 71.87 ± 0.85 |
| ProtoNet [28]    | Euclidean | 60.37 ± 0.83| 65.65 ± 0.92 | 41.54 ± 0.76 | 66.09 ± 0.92 |
| DeepEMD [9]*     | EMD    | 65.45 ± 0.28| 69.84 ± 0.32 | 45.58 ± 0.26| 70.71 ± 0.30|
| QSFormer         | Ours   | 65.24 ± 0.28| 72.47 ± 0.31 | 46.51 ± 0.26| 75.44 ± 0.29|

TABLE IX

Comparison Results of Performance and Additional Cost of Baseline and QSFormer

| Methods          | Dataset       | minImageNet | tieredImageNet | FC100 | CUB   | Parameter | FLOPs (G) | Throughput (fps) |
|------------------|---------------|-------------|----------------|-------|-------|-----------|-----------|------------------|
| Baseline         |               | 59.64 ± 0.27| 55.87 ± 0.31 | 39.47 ± 0.23| 62.09 ± 0.29| 12.4M     | 3.5       | 371.7            |
| QSFormer         |               | 65.24 ± 0.28| 72.47 ± 0.31 | 46.51 ± 0.26| 75.44 ± 0.29| 52.8M     | 4.1       | 333.3            |
| Improvement†     |               | +5.6        | +16.6          | +7.04 | +13.35| -         | -         | -                |

Fig. 3. Comparison of similarity distribution between Baseline and our QSFormer. The similarities of “Q-S pos” become larger while the similarities of “Q-S neg” become smaller, which indicates they are more easily separated.

that Variant #1 obtains the performance of 37.09 ± 0.26 on FC100 dataset and 47.11 ± 0.29 on CUB dataset while Variant #2 achieves 36.16 ± 0.26 and 32.57 ± 0.24, i.e., QSFormer based on CIPE can achieve better performance than QSFormer based on Fully-Transformer model. The possible reasons for this result are as follows: 1) Since the training data of few-shot learning task is small, it is difficult to train a good Full-Transformer model. 2) In terms of model architecture, the pure Transformer architecture usually achieves sub-optimal performance due to the lack of inductive bias [38], [90]. Therefore, we finally adopt CIPE (ResNet12+Transformer) as the feature extractor to extract the initial feature of input images.

2) Component Analysis: Our proposed QSFormer mainly contains three components: Cross-scale Interactive Feature Extractor (CIPE), Sample Transformer Module (sampleFormer) and Patch Transformer Module (patchFormer). The experimental results of ablation study can be seen in Table VII. We reproduce cosine classifier method [32] consisting of CNN network and cosine distance as the Baseline network for comparison. From Table VII, we can observe: (1) By comparing #1 with #2, the performance of Baseline network can be significantly improved with the help of CIPE, which demonstrates the effectiveness of CIPE. (2) By comparing #2 with #3, the proposed sampleFormer significantly improves the performance of model based on #2, which indicates the effectiveness of sampleFormer module. (3) The performance obtained by only using patchFormer (#4) is improved based on #2, which indicates the proposed patchFormer also can learn the discriminative image feature representations. (4) By adding both sampleFormer and patchFormer into #2, we significantly improve the accuracy of whole network. All these experiments fully prove the effectiveness of each component in the proposed QSFormer framework.

3) Analysis on Combination of patchFormer and sampleFormer: We think it is feasible to merge the patchFormer into the cross-scale interactive feature extractor as the last attention module and the patchFormer and sampleFormer are combined in a sequential way. As shown in Table X, we implement this sequential way on 5way-1shot learning task and find that the proposed ‘addition’ manner achieves the performance of 75.44 ± 0.29 on CUB [81] dataset, which performs better than the sequential manner (i.e., 74.08 ± 0.31). We think the possible reason is that in a sequential way, the sampleFormer is dependent on the output representation of patchFormer which may lead to sub-optimal learning results.

4) Similarity Metric Analysis: To analyze the effectiveness of the proposed QSFormer on metric learning, we visualize the similarity distribution of Baseline and QSFormer on the more challenging 5-way 1-shot FSL task, as shown in Fig. 3. For 5-way 1-shot FSL task, each query sample
generates the similarity results of one positive query-support sample pair (i.e., “Q-S pos”) and four negative query-support sample pairs (i.e., “Q-S neg”) during the metric learning process. To facilitate the comparison of the similarity results of “Q-S pos” and “Q-S neg”, we average the similarity values of four “Q-S neg” corresponding to each query sample. For this experiment, we perform 10 episodes, where each episode random selects $15 \times 5 = 75$ query samples for classification, i.e., we can get the $75 \times 10 = 750$ similarity values of “Q-S pos” and “Q-S neg”, respectively. Subsequently, we count the number of “Q-S pos” and “Q-S neg” within a certain range according to the normalized similarity values and thus produce the similarity distribution as shown in Fig. 3. We can observe that: (1) the similarity values of “Q-S pos” obtained by the Baseline method are generally below 0.5, while “Q-S neg” is above 0.25. (2) In our proposed QSFormer, the similarity values of “Q-S pos” are mostly above 0.5, while “Q-S neg” are mostly below 0.25. Therefore, our proposed QSFormer can separate positive and negative query-support sample pairs more accurately.

In addition, we also compare our QSFormer with other metric learning algorithms, including cosine classifier [32], MatchNet [13], ProtoNet [28] and DeepEMD [9]. These compared methods are reproduced with the same settings and training schemes as ours for a more fair comparison. As shown in Table VIII, we observe that the proposed QSFormer obtains the best performance on four publicly popular datasets. These experiments fully demonstrate the effectiveness of the proposed QSFormer on metric learning.

5) Cross-Domain Analysis: To analyze the transferable ability of the proposed QSFormer, we conduct a cross-domain experiment by following [32] and [9]. The training and testing are implemented on miniImagenet dataset and CUB dataset, respectively. As shown in Table IV, our proposed QSFormer achieves the best performance on the 1-shot setting ($55.04 \pm 0.29$) and the second-best results on the 5-shot, i.e., $77.12 \pm 0.24$. These results demonstrate that the proposed QSFormer learns the discriminative information across domains, and adaptively explores the correspondence of query-support samples.

6) Loss Function Analysis: The loss function Eq. (14) is composed of Cross-Entropy loss and Contrastive loss, where Cross-Entropy loss ($L_{ce}$) supervises the prediction label learned by the whole QSFormer network. Moreover, for sampleFormer branch, we add a contrastive loss to further enhance the discriminative ability of global representation, i.e., $L_{cl}$. For patchFormer, we think the contrastive loss $L_{cl}^p$ is not necessary because the patchFormer branch can fully exploit the local information for image representation and thus maintains high discriminative ability. To analyze the usefulness of these loss functions, we conduct ablation studies. As shown in Table XI, we can see that adding the contrastive loss to the patchFormer branch (i.e., #3) obtains $+0.03\%$ and $+0.08\%$ improvements on FC100 and CUB datasets, respectively. As a result, it does not bring obvious improvements but with higher training burden.

7) Complexity: The sampleFormer is the proposed model that can be computed efficiently due to the sample size of few-shot learning problem. For patchFormer, the number of tokens is $5 \times 5$ in our experiments and the self-attention can also be computed efficiently. Overall, the processing time of the model is not significantly large. As shown in Table IX, we can observe that compared with Baseline (371.7 fps) and the proposed QSFormer model needs 333.3 fps. The proposed method gains 5.6%, 16.6%, 7.04% and 13.35% improvements on four datasets.

E. Parameter Analysis

There are two important parameters in our model, including the balanced hyper-parameter $\lambda$ in Eq. (12) for local and global metric, and the number of sampleFormer layers $N$. Here, we respectively conduct the experiments of parameter analysis on the FC100 [29] and CUB [81] datasets on 5-way 1-shot FSL task to check their influence. For $\lambda$, as shown in Fig. 4, we can observe that the performance is relatively stable when we slightly adjust the balanced parameter $\lambda$ in the range of (0.2, 0.6) on the FC100 [29] dataset and finally

---

**TABLE X**

| Methods       | Manners | FC100       | CUB        |
|---------------|---------|-------------|------------|
| QSFormer      | Addition| 75.44 ± 0.29| 74.08 ± 0.31|

**TABLE XI**

| # | $L_{ce}$ | $L_{ce}^p$ | $L_{cl}$ | FC100 | CUB |
|---|---------|------------|----------|-------|-----|
| 1 | ✓       | ✓          | ✓        | 45.70 ± 0.26 | 74.23 ± 0.30 |
| 2 | ✓       | ✓          | ✓        | 46.51 ± 0.26 | 75.44 ± 0.29 |
| 3 | ✓       | ✓          | ✓        | 46.54 ± 0.26 | 75.52 ± 0.29 |

---

![Fig. 4. Ablation study of hyper-parameters $\lambda$ on two datasets. For FC100 [29] dataset, it is relatively insensitive when we slightly adjust $\lambda$ in the range of (0.2, 0.6) and finally we set it to 0.4. For CUB [81] dataset, it gets the best results when $\lambda = 0.4$ and thus we set it to 0.4.](image)

![Fig. 5. Ablation study of the number $N$ of sampleFormer layers. We find that it respectively gets the best results when $N = 4$ and $N = 2$ on FC100 [29] and CUB [81] datasets.](image)
Fig. 6. 2D t-SNE [91] visualization of the learned feature representations on Baseline, patchFormer and sampleFormer. We can observe that the different object/class representations learned by patchFormer and sampleFormer networks are more discriminative than baseline network.

Fig. 7. Visualization of similarity matrix of patchFormer, sampleFormer and QSFormer, where ‘GT’ denotes the ground-truth matrix of query samples.

we set it to 0.4. On the CUB [81] dataset, it gets the best results when $\lambda = 0.4$ and thus we set it to 0.4 on this dataset. For the number $N$ of sampleFormer layers, we can find that our performance is increasing continuously when the $N$ is changing from 2 to 4 on FC100 [29] dataset in Fig. 5 and thus we set $N = 4$ for it. On the CUB [81] dataset, it gets the best results when $N = 2$ on the CUB dataset. Thus, we set it to $N = 2$ on this dataset.

F. Visualization

1) Representation Visualization: In order to verify the feature representation learning ability of patchFormer and sampleFormer module, we use the 2D t-SNE visualization technology. As shown in Fig. 6, it respectively shows the 2D visualizations of the final output representation learned by Baseline, patchFormer and sampleFormer network. The different colors/shapes of Fig. 6 denote different classes. Here, we only use 5 classes. It is clearly found that different object representations learned by the Baseline network are mixed together. However, the object representations obtained by the patchFormer and sampleFormer networks are distributed more clearly, especially those learned by the sampleFormer network. These demonstrate the more discrimination of the proposed patchFormer and sampleFormer modules for representation learning.

2) Similarity Matrix Visualization: The sampleFormer module aims to explore the relationships of samples for coupling sample representations and metric learning of samples together while the patchFormer aims to fully exploit the local cues for image representation. In order to analyze the effectiveness of them respectively, we show the similarity/correlation matrix between query-support sample pairs learned by them respectively. The visualization is shown in Fig. 7. It shows the 5way-1shot setting with 1 query sample per class. Each row represents a query sample and each column represents a support sample. The ‘GT’ refers to the ground-truth matrix of query samples. Here, we can observe that 1) the proposed sampleFormer (b) can learn the relationship between query-support image pairs better than the patchFormer (a). 2) we obtain better similarity measurement through the joint learning of global/sampleFormer and local/patchFormer modules, i.e., (c).

V. CONCLUSION AND FUTURE WORKS

This paper mainly proposes a novel unified Query-Support Transformer (QSFormer) to deeply exploit the sample relationships in query and support sets for few-shot classification task. QSFormer mainly contains sample Transformer (sampleFormer) module and patch Transformer (patchFormer) module. sampleFormer is designed to meet the problem setting of few-shot classification, i.e., it couples the sample representation and metric learning between query and support sets together via a single Transformer architecture. Meanwhile, as a complementary, patchFormer is also adopted to model the local structural metric between query and support samples. A new CNN feature extractor (CIFE) is also proposed to provide an effective CNN backbone for our approach. Extensive experiments demonstrate the effectiveness and superiority of the proposed QSFormer.
The proposed QSFormer contains both sampleFormer and patchFormer branches and integrates them together in the final metric learning stage. Thus, one limitation of the proposed QSFormer is that it lacks full considering the interaction between middle-level representations of two branches. In future work, we will introduce this interaction of the middle-level representations. Also, we will further introduce semantic information to guide the interaction of global and local learning branches.

REFERENCES

[1] L. Zhang et al., “Domain knowledge powered two-stream deep network for few-shot SAR vehicle recognition,” IEEE Trans. Geosci. Remote Sens., vol. 60, 2022, Art. no. 5215315.

[2] H. Ren et al., “Transductive prototypical attention reasoning network for few-shot SAR target recognition,” IEEE Trans. Geosci. Remote Sens., vol. 61, 2023, Art. no. 5206813.

[3] L. Zhang et al., “Azimuth-aware discriminative representation learning for semi-supervised few-shot SAR vehicle recognition,” Remote Sens., vol. 15, no. 2, p. 331, Jan. 2023.

[4] Y. Xiao, V. Lepetit, and R. Marlet, “Few-shot object detection and viewpoint estimation for objects in the Wild,” IEEE Trans. Pattern Anal. Mach. Intell., vol. 45, no. 3, pp. 3090–3106, Mar. 2023.

[5] G. Han, J. Ma, S. Huang, L. Chen, and S. Chang, “Few-shot object detection with fully cross-transformer,” in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR), Jun. 2022, pp. 5311–5320.

[6] X. Chen, Y. Li, L. Yao, E. Adeli, Y. Zhang, and X. Wang, “Generative adversarial U-Net for domain-free few-shot medical diagnosis,” Pattern Recognit. Lett., vol. 157, pp. 112–118, May 2022.

[7] A. A. Rusu et al., “Meta-learning with latent embedding optimization,” in Proc. IEEE/CVF Int. Conf. Learn. Represent., Mar. 2018, pp. 6907–6917.

[8] M. A. Jamal and G. Qi, “Task agnostic meta-learning for few-shot learning,” in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR), Jun. 2019, pp. 11711–11719.

[9] C. Zhang, Y. Cai, G. Lin, and C. Shen, “DeepEMD: Few-shot image classification with differentiable earth mover’s distance and structure classifiers,” in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR), Jun. 2020, pp. 12200–12210.

[10] L. Zhang, S. Wang, X. Chang, J. Liu, Z. Ge, and Q. Zheng, “Auto-FSL: Searching the attribute consistent network for few-shot learning,” IEEE Trans. Circuits Syst. Video Technol., vol. 32, no. 3, pp. 1213–1223, Mar. 2022.

[11] D. Wennehmer, L. Tang, and B. Hariharan, “Few-shot classification with feature map reconstruction networks,” in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR), Jun. 2021, pp. 8008–8017.

[12] Z. Dang, M. Luo, C. Jia, C. Yan, X. Chang, and Q. Zheng, “Counterfactual generation framework for few-shot learning,” IEEE Trans. Circuits Syst. Video Technol., early access, Feb. 1, 2023, doi: 10.1109/TCSVT.2023.3241651.

[13] O. Vinyals et al., “Matching networks for one-shot learning,” in Proc. Adv. Neural Inf. Process. Syst., vol. 29, 2016, pp. 3637–3645.

[14] J. Luketina, M. Berglund, K. Greff, and T. Raiko, “Scalable gradient-based tuning of continuous regularization hyperparameters,” in Proc. IEEE/CVF Int. Conf. Mach. Learn., Mar. 2016, pp. 2952–2960.

[15] A. Santoro, S. Bartunov, M. Botvinick, D. Wierstra, and T. Lillicrap, “Meta-learning with memory-augmented neural networks,” in Proc. Adv. Neural Inf. Process. Syst., vol. 32, no. 11, pp. 7706–7718, Nov. 2019.

[16] A. Zhmoginov, M. Sandler, and M. Vladymyrov, “HyperTransformer: Model generation for supervised and semi-supervised few-shot learning,” 2022, arXiv:2201.04182.

[17] R. Hou, H. Chang, B. Ma, S. Shan, and X. Chen, “Cross attention network for few-shot classification,” in Proc. Adv. Neural Inf. Process. Syst., vol. 32, 2019, pp. 4003–4014.

[18] W. Jiang, W. Zhou, and H. Hu, “Double-stream position learning transformer network for image captioning,” IEEE Trans. Circuits Syst. Video Technol., vol. 31, no. 6, pp. 1355–1365, Nov. 2022.

[19] A. Zhmoginov, M. Sandler, and M. Vladymyrov, “HyperTransformer,” 2022, arXiv:2201.04182.

[20] S. Qiao, C. Liu, W. Shen, and A. Yuille, “Few-shot image recognition by predicting parameters from activations,” in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit., Jun. 2018, pp. 7229–7238.

[21] Y. Liu et al., “Learning to propagate labels: Transductive propagation network for few-shot learning,” in Proc. IEEE/CVF Conf. Int. Conf. Learn. Represent., May 2019.
Xixi Wang is currently pursuing the Ph.D. degree in computer science with Anhui University, Hefei, China. Her main research interests include computer vision and machine learning, especially on person re-identification, few-shot learning, and multi-modal learning.
Xiao Wang (Member, IEEE) received the B.S. degree from West Anhui University, Luannan, China, in 2013, and the Ph.D. degree in computer science from Anhui University, Hefei, China, in 2019. In 2015 and 2016, he was a Visiting Student with the School of Data and Computer Science, Sun Yat-sen University, Guangzhou, China. He also has a visit to the UBTECH Sydney Artificial Intelligence Centre, Faculty of Engineering, The University of Sydney, in 2019. He finished his postdoctoral research with the Peng Cheng Laboratory, Shenzhen, China, from April 2020 to April 2022. He is currently an Associate Professor with the School of Computer Science and Technology, Anhui University. His current research interests are mainly computer vision, event-based vision, machine learning, and pattern recognition. He serves as a Reviewer for a number of journals and conferences, such as IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS FOR VIDEO TECHNOLOGY, IEEE TRANSACTIONS ON NEURAL NETWORKS AND LEARNING SYSTEMS, IEEE TRANSACTIONS ON IMAGE PROCESSING, IJCV, CVIU, PR, CVPR, ICCV, AAAI, ECCV, ACM MM, and WACV.

Bo Jiang received the B.S. degree in mathematics and in applied mathematics and the M.Eng. and Ph.D. degrees in computer science from Anhui University, China, in 2009, 2012, and 2015, respectively. He is currently an Associate Professor in computer science with Anhui University. He has published more than 60 papers, including 20 papers in top conferences and journals, such as CVPR, NeurIPS, IJCV, IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE, IEEE TRANSACTIONS ON IMAGE PROCESSING, and IEEE TRANSACTIONS ON NEURAL NETWORKS AND LEARNING SYSTEMS. His current research interests include image matching, graph data representation, and learning and few-shot learning. He gained some research projects of NSFC and the Natural Science Foundation of Anhui Province. He also received the Rising Star Award of ACM Hefei.

Bin Luo (Senior Member, IEEE) received the B.Eng. degree in electronics and the M.Eng. degree in computer science from Anhui University, China, in 1984 and 1991, respectively, and the Ph.D. degree in computer science from the University of York, U.K., in 2002. He is currently a Professor with Anhui University. He has published more than 200 papers in journals and refereed conferences. His current research interests include random graph based pattern recognition, image and graph matching, and spectral analysis. He is currently the Chair of the IEEE Hefei Subsection. He served as a Peer Reviewer for international academic journals, such as IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE, PATTERN RECOGNITION, and PATTERN RECOGNITION LETTERS.