Multi-Level Correlation Network For Few-Shot Image Classification

1st Yunkai Dang
College of Intelligence and Computing
Tianjin University
Tianjin, P.R. China
dyk@tju.edu.cn

2nd Meijun Sun
College of Intelligence and Computing
Tianjin University
Tianjin, P.R. China
sunmeijun@tju.edu.cn

3rd Min Zhang
AI Division, School of Engineering
Westlake University)
Hangzhou, P.R. China
zhangmin@westlake.edu.cn

4th Zhengyu Chen
AI Division, School of Engineering
Westlake University)
Hangzhou, P.R. China
chenzhengyu@zju.edu.cn

5th Xinliang Zhang
College of Intelligence and Computing
Tianjin University
Tianjin, P.R. China
zhangxinliang@tju.edu.cn

6th Zheng Wang
College of Intelligence and Computing
Tianjin University
Tianjin, P.R. China
wzheng@tju.edu.cn

7th Donglin Wang †
AI Division, School of Engineering
Westlake University)
Hangzhou, P.R. China
wangdonglin@westlake.edu.cn

Abstract—Few-shot image classification(FSIC) aims to recognize novel classes given few labeled images from base classes. Recent works have achieved promising classification performance, especially for metric-learning methods, where a measure at only image feature level is usually used. In this paper, we argue that measure at such a level may not be effective enough to generalize from base to novel classes when using only a few images. Instead, a multi-level descriptor of an image is taken for consideration in this paper. We propose a multi-level correlation network (MLCN) for FSIC to tackle this problem by effectively capturing local information. Concretely, we present the self-correlation module and cross-correlation module to learn the semantic correspondence relation of local information based on learned representations. Moreover, we propose a pattern-correlation module to capture the pattern of fine-grained images and find relevant structural patterns between base classes and novel classes. Extensive experiments and analysis show the effectiveness of our proposed method on four widely-used FSIC benchmarks.

Index Terms—Few-shot image classification; Metric Learning;

I. INTRODUCTION

Inspired by the human ability to recognize novel concepts from few images, few-shot image classification (FSIC) has attracted the interest of many researchers. FSIC aims to classify unlabeled images from novel classes given only few labeled images based on the trained model from base classes [1–5]. Compared with traditional image classification tasks, the biggest challenge of FSIC is that the label spaces of base and novel classes are inconsistent, *i.e.*, the labels
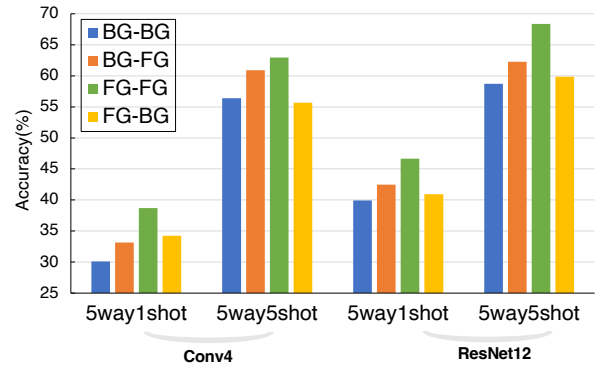
† Corresponding author.

Fig. 1. 5way 1shot and 5shot performance on different background(BG) and foreground(FG) using two backbones of Conv4 and ResNet12 ProNet on *mini*Imagenet datasets.

of novel classes are not seen by base classes [6]. Most works have been proposed to solve the problem and are categorized, mainly divided into fine-tuning, meta-learning and metric-learning methods. **Fine-tuning methods** [7, 8] use non-episode training paradigm (pre-training + test-tuning) to learn novel classes. **Meta-learning methods** [1, 9] use the bi-level optimization with episodic-training paradigm. **Metric-learning methods** [2–4] also use episodic training but use distances to recognize novel classes in the embedding space. In this paper, we consider metric-learning method because it is simple and effective.

Recently, metric-learning methods have achieved excellent performance by comparing the similarities (or distances) between the unlabeled (query set) and labeled (support set)

images in the embedding space. However, the similarity is calculated using only the global information of images. If base and novel classes have the distribution shift, the performance of the trained model degrades. The main reason is that images from the same class tend to share similar backgrounds, which may cause them to be close to each other in the metric space. The transferability of the model trained on base classes becomes worse, when novel classes have different backgrounds from the base classes. We give a example in Fig. 1 to show our motivation that the background is harmful. It can be observed that under any condition, removing the background consistently leads to improved performance. To alleviate the influence of image background, the main challenge of metric-learning methods is how to effectively capture the local information, such as the foreground of the image, on top of the global information.

To solve the challenge, in this paper, we propose a multi-correlation network (MLCN) to learn enough local information. Specifically, we present the self-correlation module and cross-correlation module to learn the semantic correspondence relations of local information based on the learned representation. Inspired by the way that human beings tend to instinctively focus on the most relevant areas of a pair of labeled and unlabeled images when trying to recognize a sample from an unseen class using only a few images [10], we proposed a pattern-correlation module to capture the regions of fine-grained images and find relevant structural patterns between base classes and novel classes. The proposed MLCN combines the three correlation modules to improve the transferability of learned representations and learn enough local information to generalize to novel classes. Experiments on four standard benchmark datasets demonstrate that the proposed MLCN can effectively improve FSIC accuracy.

Contributions. To summarize, our contributions are:

- We verify that removing background consistently leads to the performance consistently and significantly improved. By taking this into consideration, we propose a multi-level correlation network (MLCN).
- We present the self-correlation module and cross-correlation module to learn the semantic correspondence relation of local information. Moreover, we propose the pattern-correlation module to capture the relevant structure of fine-grained images.
- Experiments and analysis on four standard benchmarks show that our method achieves the state of the art and ablation studies validate the effectiveness of three correlation modules.

II. RELATED WORK

Few-shot image classification. Few-shot image classification (FSIC) aims to recognize unlabeled images from the novel classes given few labeled images by transferring the knowledge from the base classes. To solve this problem, researchers have proposed various methods, such as fine-tuning, meta-learning, and metric-learning methods. **Fine-tuning methods** [7 ? , 8] are also called non-episode meth-

ods. These methods generally follow the standard transfer learning procedure, consisting of two phases, *i.e.*, pre-training with base classes and test-tuning with novel classes. **Meta-learning methods** [1, 9] adopt a *learning-to-learn* paradigm to transfer the knowledge from the base classes to the novel classes. **Metric-learning methods** [2–4] employ a *learning-to-compare* paradigm to learn representations that can be transferred between the base and novel classes. In this paper, our proposed method belongs to the metric-learning methods.

Metric-learning methods. Metric learning [2–4] is the method of learning a distance metric for the input space of base classes from a collection of pairs of similar and dissimilar points. Prototypical Network [3, 9] is widely used in metric-based methods for FSIC. It takes the center point of a support class as its prototype and conducts classification by comparing similarities between query instances and support prototypes. However, such a process does not take into account the similarities between the query and support embedding. The main idea of our work is to improve the transferability of embedding by computing the similarities of the query and support embedding.

Multi-level correlation network. Recent works [12, 13] adopt self-similarity and cross-similarity as an intermediate feature transformation for a deep neural network and show that it helps the network learn an effective representation. CAN [10] proposes a cross attention module to find the semantic relevance between the query and support set. Unlike the previous works [14], our multi-level correlation network (MLCN) directly uses the correlation tensor to refine the representation of the query and support set. Different from [15] using the Expectation-Maximization algorithm to learn features, we use the Bi-level optimization with episodic-training paradigm. Different from [10, 13], we do not use the projection module and the 4D convolution without additional parameters to avoid overfitting for FSIC.

III. METHODOLOGY

A. Preliminary and Overview

In the standard few-shot image classification (FSIC) scenario, a training data $\mathcal{D}_{train}$ from base classes $\mathcal{C}_{base}$ and a testing data $\mathcal{D}_{test}$ from novel classes $\mathcal{C}_{novel}$ are given, where $\mathcal{C}_{base} \bigcap \mathcal{C}_{novel} = \varnothing$. Following [10, 13], we adopt the N-way K-shot episodic paradigm to train and test FSIC models, which has been demonstrated to be effective to enhance the generalization performance. For each episode, we randomly sample N classes and K labeled images per classes as the support set $\mathcal{S} = \{(x_i^s, y_i^s)|i = 1, ..., N \times K\}$ and a fraction of the remaning Q images from the N classes as the query set $\mathcal{Q} = \{(x_i^q, y_i^q)|i = 1, ..., N \times Q\}$. FSIC aims to learn a classifier to classify images into target classes given only a few images for each class. The key issue is to present each support class and query sample and measure the similarity between them for few-shot image classification.

In this section, we propose the multi-level correlation network (MLCN) to address the challenge of generalization to novel classes from a similarity perspective. As shown in Fig. 2,
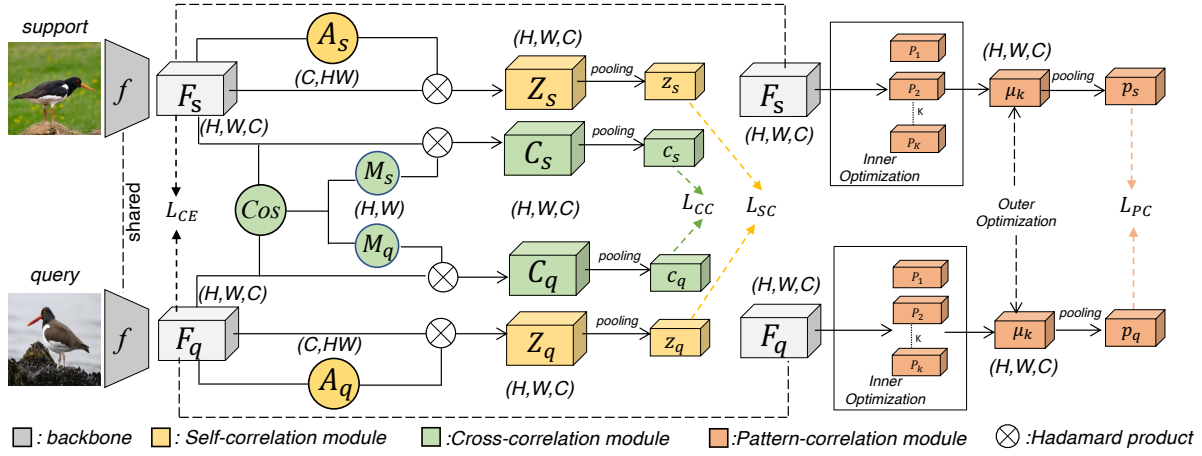
Fig. 2. The overview of our multi-level correlation network (MLCN) for few-shot image classification. The base representations, $\mathbf{F}_q$ and $\mathbf{F}_s$ are the base representations of the backbone feature extractor. MLCN minimizes the self-correlation module loss $\mathcal{L}_{SC}$, the cross-correlation module loss $\mathcal{L}_{CC}$, the pattern-correlation module loss $\mathcal{L}_{PC}$ and the anchor-based classification loss $\mathcal{L}_{CE}$ to improve the transferability of embedding and capture enough local information.

our MLCN has three complementary branches based on the backbone. Then we present technical details of self-correlation module, cross-correlation module and pattern-correlation module in Sec.3.2, Sec.3.3 and Sec.3.4. Last, we introduce the overall loss in Sec.3.5.

B. Self-correlation Module

To improve the generalizability of the representation, as shown in Fig. 2, we propose the architecture of self-correlation module. Here, we use it to transform the base representations, F_q and F_s to self-correlation representations, Z_q and Z_s by the self-correlation attention map A_q and A_s.

Self-correlation Attention Map. To locate the discriminative object regions of query and support set, we firstly produce a self-correlation attention map, A_q and $A_s \in R^{C \times HW}$, which generates the weights of base representations, F_q and $F_s \in R^{H \times W \times C}$.

$$\mathbf{A}_q(x_q) = softmax(F_q(x_q)), \quad (1)$$

where x_q is position at the query feature map and A_s is similarly computed by $\mathbf{E_q}.(1)$. The self-correlation embeddings, $\mathbf{Z}_s$ and $\mathbf{Z}_q \in R^C$, are multiplied by the self-correlation attention map A_q. Then, we compute the final self-correlation embedding $\mathbf{z}_q$ of query feature by using pooling:

$$\mathbf{z_q} = \frac{1}{HW} \sum_{x_q} A_q(x_q) \otimes F_q(x_q), \quad (2)$$

where x denotes a position at the feature map. The final embedding $\mathbf{z_s}$ of support images is computed similarly by $\mathbf{E_q}.(2)$. In the N-way K-shot classification setting, this self-correlation pooling generates a set of NK different views of a query, $\left\{\bar{\mathbf{z}}_\mathbf{q}^{(n)}\right\}_{n=1}^N$, and a set of support embedding in the context of the query, $\left\{\bar{\mathbf{z}}_\mathbf{s}^{(n)}\right\}_{n=1}^N$.

$\mathcal{L}_{SC}$: **self-correlation module loss.** We average the K query and support embedding vectors, each of which is attended in the context of kth support from n^{th} class to compute

$\left\{\bar{\mathbf{z}}_\mathbf{q}^{(n)}\right\}_{n=1}^N$ and $\left\{\bar{\mathbf{z}}_\mathbf{s}^{(n)}\right\}_{n=1}^N$. The self-correlation module loss guides the model to map a query embedding close to the prototype embedding of the same class in the metric space, so we have

$$\mathcal{L}_{SC} = -\log \frac{\exp\left(\text{sim}\left(\bar{\mathbf{z}}_\mathbf{s}^{(n)}, \bar{\mathbf{z}}_\mathbf{q}^{(n)}\right)/\tau_1\right)}{\sum_{n'=1}^N \exp\left(\text{sim}\left(\bar{\mathbf{z}}_\mathbf{s}^{(n')}, \bar{\mathbf{z}}_\mathbf{q}^{(n')}\right)/\tau_1\right)}, \quad (3)$$

where τ_1 is a scalar temperature factor and $sim(.,.)$ is a cosine similarity .

C. Cross-correlation Module

The self-correlation module localizes the discriminative object regions of each query and support set may ignore the semantic correspondence relation between them. To learn this relation, as shown in Fig. 2, we use the cross-correlation module to further find the reveal relevant contents between the query and support set.

Cross-correlation Attention Map. To improve the transferability of embedding, we transform base representations F_s and F_q into more compact representations by constructing a 4-dimensional correlation tensor $\mathbf{Cos} \in R^{H \times W \times H \times W}$, which computes the cosine similarity between two features. The cross-correlation attention map $A_q \in R^{H \times W}$ of the query is computed by

$$\mathbf{M}_q = \frac{1}{HW} \sum_{x_s} \frac{\exp\left(Cos(F_q(x_q), F_s(x_s))/\gamma\right)}{\sum_{x_{q'}} \exp\left(Cos(F_q(x_{q'}), F_s(x_s))/\gamma\right)}, \quad (4)$$

where γ is a scalar temperature parameter and $Cos(.,.)$ is a matching score between the positions x_q and x_s of the F_q and F_s. And the M_s is similarly computed by $\mathbf{E_q}.(4)$. Similar to $\mathbf{E_q}.(2)$, we compute the cross-correlation of support feature embedding $c_q \in R^{H \times W \times C}$ by using the pooling:

$$\mathbf{c_q} = \frac{1}{HW} \sum_{x_q} M_q(x_q) \otimes F_q(x_q), \quad (5)$$

where x_q is a position at the query feature map. The final embedding of the query, $\mathbf{c_s}$, is computed similarly by $\mathbf{E_q}$. (5).

$\mathcal{L}_{CC}$ **: cross-correlation module loss.** Similar to $\mathbf{E_q}$. (2), we compute the support feature embedding by multiplying base representations F_s. Similar to self-correlation module loss, we average the K query and support embedding vectors as $\left\{ \bar{\mathbf{c}}_{\mathbf{q}}^{(n)} \right\}_{n=1}^{N}$ and $\left\{ \bar{\mathbf{c}}_{\mathbf{s}}^{(n)} \right\}_{n=1}^{N}$. The cross-correlation module loss guides the model to map a query embedding close to the prototype embedding of the same class:

$$\mathcal{L}_{CC} = -\log \frac{\exp\left(\operatorname{sim}\left(\bar{\mathbf{c}}_{\mathbf{s}}^{(n)}, \bar{\mathbf{c}}_{\mathbf{q}}^{(n)}\right)/\tau_2\right)}{\sum_{n'=1}^{N} \exp\left(\operatorname{sim}\left(\bar{\mathbf{c}}_{\mathbf{s}}^{(n')}, \bar{\mathbf{c}}_{\mathbf{q}}^{(n')}\right)/\tau_2\right)}, \quad (6)$$

where $sim(.,.)$ indicates the cosine similarity and τ_2 is a scalar temperature factor.

D. Pattern-correlation Module

As shown in Fig. 2, we use the pattern-correlation module to find relevant structural pattern between base classes and novel classes. Pattern-correlation module is defined as a probability mixture model which linearly combines probabilities from base distributions as:

$$p(s_i|\theta) = \sum_{k=1}^{K} w_k p_k(s_i|\theta), \quad (7)$$

where w_k denotes the mixing weights satisfying $0 \leq w_k \leq 1$ and $\sum_{k=1}^{K} w_k = 1$. θ denotes the model parameters which are learned by the backbone for extracting embedding. $s_i \in S$ denotes the ith feature samples and $p_k(s_i|\theta)$ denotes the kth base model. Thus, we calculate the $p_k(s_i|\theta)$ as :

$$p_k(s_i|\theta) = \beta(\theta) e^{distance}, \quad (8)$$

where $\beta(\theta)$ is the normalization coefficient, and the $distance$ is defined as the euclidean distance of μ_k and s_i. Here, μ_k is the mean vector of the kth model, and k denotes the concentration parameter. Pattern-correlation module is estimated by using the bi-level optimization, which includes iterative inner optimization and outer optimization. In **inner optimization step**, we use the $p_k(s_i|\theta)$ to extract sample features as

$$P_{ik} = \frac{p_k(s_i|\theta)}{\sum_{k=1}^{K} p_k(s_i|\theta)}, \quad (9)$$

where k denotes the concentration parameter and is set as k =25 in experiments. In **outer optimization**, we use the P_{ik} to update the mean vectors as $u_k \in R^{H \times W \times C}$, which is computed by

$$\mu_k = \frac{\sum_{i=1}^{N} P_{ik} s_i}{\sum_{i=1}^{N} P_{ik}}, \quad (10)$$

where s_i denotes the ith feature samples and N denotes the number of samples.

$\mathcal{L}_{PC}$ **: pattern-correlation module loss.** The final embeddings of and query $\mathbf{p_q}$ and support $\mathbf{p_s}$ are computed by pooling of the μ_k. Similar to $\mathbf{E_q}$. (6) and $\mathbf{E_q}$. (3), we average the K

query and support embedding vectors for each class to obtain a set of prototype embedding $\{\bar{\mathbf{p}}_{\mathbf{q}}^{n}\}_{n=1}^{N}$ and $\{\bar{\mathbf{p}}_{\mathbf{s}}^{n}\}_{n=1}^{N}$. $\mathcal{L}_{PC}$ is computed by cosine similarity between a query and support prototypes:

$$\mathcal{L}_{PC} = -\log \frac{\exp\left(\operatorname{sim}\left(\bar{\mathbf{p}}_{\mathbf{s}}^{(n)}, \bar{\mathbf{p}}_{\mathbf{q}}^{(n)}\right)/\tau_3\right)}{\sum_{n'=1}^{N} \exp\left(\operatorname{sim}\left(\bar{\mathbf{p}}_{\mathbf{s}}^{(n')}, \bar{\mathbf{p}}_{\mathbf{q}}^{(n')}\right)/\tau_3\right)}, \quad (11)$$

where $sim(.,.)$ is cosine similarity and τ_3 is a scalar temperature factor.

E. Overall Loss

the anchor-based classification loss $\mathcal{L}_{CE}$ is computed with an additional fully-connected classification layer on top of average-pooled based feature F_q. It guides the model to classify a query class of class $c \in C_{base}$, so we have

$$\mathcal{L}_{CE} = -\log \frac{\exp\left(\mathbf{w}_c^{\top} \mathbf{F_q} + \mathbf{b}_c\right)}{\sum_{c'=1}^{|C_{base}|} \exp\left(\mathbf{w}_{c'}^{\top} \mathbf{F_q} + \mathbf{b}_{c'}\right)}, \quad (12)$$

where $\left[\mathbf{w}_1^T, ..., \mathbf{w}_{|C_{base}|}^T\right]$ and $\left[\mathbf{b}_1, ..., \mathbf{b}_{|C_{base}|}\right]$ are weights and biases in the fully-connected layer. In summary, the final loss of each episode is defined as :

$$\mathcal{L}_{total} = \mathcal{L}_{CE} + \alpha \mathcal{L}_{SC} + \beta \mathcal{L}_{CC} + \gamma \mathcal{L}_{PC}, \quad (13)$$

where, α, β and γ are weighting factor to balance the loss terms.

IV. EXPERIMENT

A. Datasets and Implementation Details

Datasets. For experiment evaluation, we use four standard benchmarks for few-shot image classification: *mini*ImageNet, *tiered*ImageNet, CUB-200-2011 and CIFAR-FS. Following previous works[10, 13], we adopt the ResNet12 as our backbone, which consists of 4 residual blocks. The backbone network takes an image with spatial size of 84×84 as an input and provides a base representation $F \in R^{5 \times 5 \times 640}$ followed by shifting its channel activation by the channel mean of an episode. For the N-way K-shot evaluation, we test 15 query samples for each class in an episode and report average classification accuracy with 95 % confidence intervals of randomly sampled 2000 test episodes. The model is trained for 100 epochs. For optimization, we use stochastic gradient descent (SGD) with a momentum of 0.9, a weight decay of 5×10^{-4}, a learning rate of 5×10^{-2}. We set the temperature scaling factor τ_1, τ_2 and τ_3 are set to 0.5 for all four datasets. The ratio in the loss is set as α:β:γ = 4:2:1.

B. Performance Comparison

We compare the performance of MLCN with several state-of-the-art models including meta-learning and metric learning methods. For fair comparisons, we compare with other methods in the same backbone or smaller backbone in both 5-way-1shot and 5-way-5shot settings on *mini*ImageNet, *tired*ImageNet, CUB-200-2011 and CIFAR-FS datasets. As is shown in Tables 2-5, our method is superior to existing methods and achieves the best performance than both on meta-based methods and metric-based methods.

TABLE I

ABLATION EXPERIMENTS ON *mini*IMAGENET, CIFAR-FS AND CUB200-2011 DATASETS. $\mathcal{L}_{CE}$ IS THE ANCHOR-BASED CLASSIFICATION LOSS. $\mathcal{L}_{SC}$, $\mathcal{L}_{CC}$ AND $\mathcal{L}_{PC}$ ARE THE LOSS OF THE SELF-CORRELATION MODULE, CROSS-CORRELATION MODULE AND PATTERNS-CORRELATION MODULE.

L_{CE}	L_{SC}	L_{CC}	L_{PC}	miniImageNet		CIFAR-FS		CUB200-2011	
				1-shot	5-shot	1-shot	5-shot	1-shot	5-shot
√				57.89 ± 0.44	76.94 ± 0.34	61.25 ± 0.48	79.63 ± 0.34	69.92 ± 0.43	73.64 ± 0.36
√	√			64.31 ± 0.44	81.11 ± 0.31	71.19 ± 0.45	86.37 ± 0.33	73.91 ± 0.44	87.33 ± 0.28
√	√	√		64.41 ± 0.45	81.54 ± 0.31	70.58 ± 0.46	86.55 ± 0.32	74.40 ± 0.67	89.19 ± 0.27
√	√		√	64.89 ± 0.44	81.33 ± 0.30	72.60 ± 0.46	85.78 ± 0.33	76.00 ± 0.45	89.12 ± 0.26
√	√	√	√	**65.54 ± 0.44**	**81.94 ± 0.31**	**74.36 ± 0.47**	**87.24 ± 0.31**	**77.96 ± 0.44**	**91.20 ± 0.24**

TABLE II

RESULTS ON THE *mini*IMAGENET DATASET.

Mthod	Backbone	5-way 1-shot	5-way 5-shot
Versa [16]	*ResNet12*	55.71	70.05
LEO [19]	*ResNet12*	56.62	69.99
BOIL [18]	*ResNet12*	58.87	72.88
R2D2 [17]	*ResNet12*	59.52	74.61
MTL [20]	*ResNet12*	62.67	79.16
TPN [21]	*ResNet12*	59.46	75.65
CC [7]	*ResNet12*	55.43 ± 0.81	77.18 ± 0.61
TapNet [10]	*ResNet12*	61.65 ± 0.15	76.36 ± 0.10
MetaOptNet [22]	*ResNet12*	62.64 ± 0.82	78.63 ± 0.46
MatchNet [2]	*ResNet12*	63.08 ± 0.80	75.99 ± 0.60
ProtoNet [23]	*ResNet12*	62.39 ± 0.21	80.53 ± 0.14
CAN [10]	*ResNet12*	63.85 ± 0.48	79.44 ± 0.34
RFS-simple [24]	*ResNet12*	62.02 ± 0.63	79.64 ± 0.44
MLCN(ours)	*ResNet12*	**65.54 ± 0.43**	**81.63 ± 0.31**

TABLE III

RESULTS ON THE *tiered*IMAGENET DATASET.

Mthod	Backbone	5-way 1-shot	5-way 5-shot
Versa [16]	*ResNet12*	57.14	75.48
LEO [19]	*ResNet12*	64.75	81.42
BOIL [18]	*ResNet12*	64.66	80.38
R2D2 [17]	*ResNet12*	65.07	83.04
MTL [20]	*ResNet12*	68.68	84.58
TPN [21]	*ResNet12*	59.91 ± 0.94	73.30 ± 0.75
CC [7]	*ResNet12*	61.49 ± 0.91	82.37 ± 0.67
TapNet [10]	*ResNet12*	63.08 ± 0.15	80.26 ± 0.12
MetaOptNet [22]	*ResNet12*	65.99 ± 0.72	81.56 ± 0.53
MatchNet [2]	*ResNet12*	68.50 ± 0.92	80.60 ± 0.71
ProtoNet [23]	*ResNet12*	68.23 ± 0.23	84.03 ± 0.16
CAN [10]	*ResNet12*	69.89 ± 0.51	84.23 ± 0.37
RFS-simple [24]	*ResNet12*	71.61 ± 0.49	85.29 ± 0.24
MLCN(ours)	*ResNet12*	**71.62 ± 0.49**	**85.58 ± 0.35**

TABLE IV

RESULTS ON THE CUB-200-2011 DATASET.

Mthod	Backbone	5-way 1-shot	5-way 5-shot
RelationNet [4]	*ResNet18*	68.58 ± 0.94	84.05 ± 0.56
CloserLook [7]	*ResNet18*	47.12 ± 0.74	64.16 ± 0.71
Baseline++[7]	*ResNet18*	67.02 ± 0.90	83.58 ± 0.50
MixtFSL [25]	*ResNet18*	73.94 ± 1.10	86.01 ± 0.50
MAML [1]	*ResNet34*[†]	67.28 ± 1.08	83.47 ± 0.59
S2M2 [26]	*ResNet34*[†]	72.92 ± 0.83	86.55 ± 0.51
CC [7]	*ResNet12*	67.30 ± 0.86	84.75 ± 0.60
ProtoNet [23]	*ResNet12*	66.09 ± 0.92	82.50 ± 0.58
MatchNet [2]	*ResNet12*	71.87 ± 0.85	85.08 ± 0.57
FEAT [27]	*ResNet12*	73.27 ± 0.22	85.77 ± 0.14
DeepEMD [28]	*ResNet12*	75.65 ± 0.83	88.69 ± 0.50
MLCN(ours)	*ResNet12*	**77.96 ± 0.44**	**91.20 ± 0.24**

TABLE V

RESULTS ON THE CIFAR-FS DATASET.

Mthod	Backbone	5-way 1-shot	5-way 5-shot
S2M2 [26]	*ResNet34*[†]	62.77 ± 0.23	75.75 ± 0.13
MAML[1]	*ConvNet*	58.90 ± 1.90	71.50 ± 1.00
DeepEMD[28]	*ResNet12*	46.47 ± 0.70	63.22 ± 0.71
R2D2 [17]	*ResNet12*	65.30 ± 0.02	78.30 ± 0.02
RelationNet[4]	*ResNet12*	55.50 ± 1.00	69.30 ± 0.80
CC [7]	*ResNet12*	60.39 ± 0.28	72.85 ± 0.65
RFS-simple [24]	*ResNet12*	71.50 ± 0.80	86.00 ± 0.50
ProtoNet [23]	*ResNet12*	72.20 ± 0.70	83.50 ± 0.50
MetaOptNet [22]	*ResNet12*	72.60 ± 0.70	84.30 ± 0.50
MLCN(ours)	*ResNet12*	**74.36 ± 0.46**	**87.24 ± 0.31**

C. Ablation Study and Visualization

Ablation Study. In this subsection, we study the effectiveness of different components in our method on three datasets. All the results are summarized in Table 1. The results in Table 1 show a signification improvement in the performance of our proposed method compared to the baseline, which only uses the $\mathcal{L}_{CE}$. Specifically, using three modules on the $\mathcal{L}_{CE}$ and $\mathcal{L}_{SC}$ improves the accuracy by an average of 7.2 % (1-shot) and 8.6 % (5-shot) on the three datasets. This allows the representation to generalize better than that only needed for the classification on $\mathcal{L}_{CE}$. Then, we then use the $\mathcal{L}_{CC}$ and

$\mathcal{L}_{PC}$ to capture local information of foreground images. The result jointly based on $\mathcal{L}_{CC}$ and $\mathcal{L}_{PC}$ are further enhanced by an average of 1.4 % (1-shot) and 1.3 % (5-shot) on the three datasets. The results indicate that our method increases the transferability of embedding on novel classes.

Visualization. We give a visualization to validate the transferability of embedding produced by our framework on novel classes. Fig. 3 visualizes the gradient-weighted class activation mapping (Grad-CAM) [29] from four loss functions under a ResNet12 feature extractor. It is observed that using $\mathcal{L}_{SC}$ and $\mathcal{L}_{CC}$ pays more attention to the relevant salient object of the local information between query and support images. It further shows that using $\mathcal{L}_{PC}$ captures the relevant pattern of query and support images. Therefore, the proposed multi-level correlation network helps the metric-learning methods to use correct visual features.
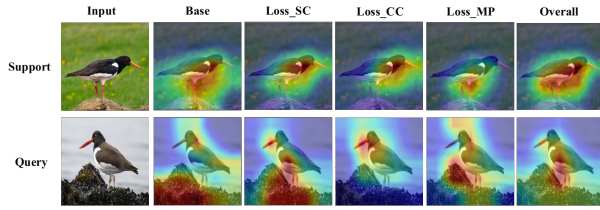
Fig. 3. GradCAM [29] visualization of the self-correlation module loss $\mathcal{L}_{SC}$, the cross-correlation module loss $\mathcal{L}_{CC}$ and the pattern-correlation loss $\mathcal{L}_{PC}$. The overall is the combined loss used in our MLCN.

V. CONCLUSION

In this paper, we propose multi-level correlation network (MLCN) for few-shot image classification (FSIC), which levrages the self-correlation module, cross-correlation module and pattern-correlation module. By combining the three modules, our method effectively captures local information for FSIC. Extensive experiments demonstrate the effectiveness of our method on widely used FSIC benchmarks.

REFERENCES

[1] Finn et al, "Model-agnostic meta-learning for fast adaptation of deep networks," in PMLR, 2017.

[2] Vinyals et al, "Matching networks for one shot learning," in NeurIPS, 2016.

[3] Snell et al, "Prototypical networks for few-shot learning," in NeurIPS, 2017.

[4] Sung et al, "Learning to compare: Relation network for few-shot learning," in CVPR, 2018.

[5] Koch et al, "Siamese neural networks for one-shot image recognition," in ICML, 2015.

[6] Li et al, "Libfewshot: A comprehensive library for few-shot learning," in arXiv preprint arXiv:2109.04898, 2021.

[7] Chen et al, "A closer look at few-shot classification," in ICLR, 2019.

[8] Oreshkin et al, "Tadam: Task dependent adaptive metric for improved few-shot learning," in NeurIPS, 2018.

[9] Chen et al, "Meta-baseline: Exploring simple meta-learning for few-shot learning," in ICCV, 2021.

[10] Hou et al, "Cross attention network for few-shot classification," in NeurIPS, 2019.

[11] Ravichandran et al, "Few-shot learning with embed- ded class models and shot-free meta training," in ICCV, 2019.

[12] Kwon et al, "Learning self-similarity in space and time as generalized motion for video action recognition," in ICCV, 2021.

[13] Kang et al, "Relational embedding for few-shot classification," in ICCV, 2021.

[14] Woo et al, "Cbam: Convolutional block attention module," in ECCV, 2018.

[15] Yang et al, "Prototype mixture models for few-shot semantic segmentation," in ECCV, 2020.

[16] Gordon et al, "Versa: Versatile and efficient few-shot learning," in NeurIPS, 2018.

[17] Rusu et al, "Meta-learning with latent embedding optimization," in ICLR, 2018.

[18] Oh et al, "Boil: Towards representation change for few-shot learning," in ICLR, 2021.

[19] Bertinetto et al, "Meta-learning with differentiable closed-form solvers," in ICLR, 2019.

[20] Sun et al, "Meta-transfer learning for few-shot learning," in ICCV, 2019.

[21] Liu et al, "Learning to propagate labels: Transductive propagation network for few-shot learning," in ICLR, 2018.

[22] Lee et al, "Meta-learning with differentiable convex optimization," in ICCV, 2019.

[23] Snell et al, "Prototypical networks for few-shot learning," in NeurIPS, 2017.

[24] Tian et al, "Rethinking few-shot image classification: a good embedding is all you need?," in ECCV, 2020.

[25] Afrasiyabi et al, "Mixture-based feature space learning for few-shot image classification," in ICCV, 2021.

[26] Mangla et al, "Charting the right manifold: Manifold mixup for few-shot learning," in WACV, 2020.

[27] Ye et al, "Few-shot learning via embedding adaptation with set-to-set functions," in ICCV, 2020.

[28] Zhang et al, "Deepemd: Few-shot image classification with differentiable earth mover's distance and structured classifiers," in ICCV, 2020.

[29] Selvaraju et al, "Grad-cam: Visual explanations from deep networks via gradient-based localization," in ICCV, 2017.