Pattern Recognition 151 (2024) 110380

Contents lists available at ScienceDirect

Pattern Recognition

journal homepage: www.elsevier.com/locate/pr

Query-centric distance modulator for few-shot classification

Wenxiao Wu [a], Yuanjie Shao [b], Changxin Gao [a], Jing-Hao Xue [c], Nong Sang [a,*]

[a] National Key Laboratory of Science and Technology on Multispectral Information Processing, School of Artificial Intelligence and Automation, Huazhong University of Science and Technology, Wuhan, 430074, China

[b] School of Electronic Information and Communications, Huazhong University of Science and Technology, Wuhan, 430074, China

[c] Department of Statistical Science, University College London, London WC1E 6BT, UK

ARTICLE INFO

Dataset link: https://github.com/Wu-Wenxiao/QCDM

Keywords:
Few-shot classification
Distance metric learning-based
Channel-weighting
Query-centric distance modulator

ABSTRACT

Few-shot classification (FSC) is a highly challenging task, as only a small number of labeled samples are available when identifying new categories. Distance metric learning-based methods have emerged as a prominent approach to FSC, which typically use a distance function to measure the difference between query and support samples for identifying the class membership of the query sample. However, these methods simply treat each channel difference between query and support features equally when computing the class scores. Since different channels in the learned feature seek for different patterns, these distance metrics fail to consider that different channels are of different importance to FSC, and thus cannot accurately measure the similarity between samples. To address this issue, we propose a **Q**uery-**C**entric **D**istance **M**odulator (**QCDM**) to generate query-related weights for each channel difference adaptively. Specifically, since the distribution of difference between a query sample and all support samples in a particular channel can reflect the importance of this channel to the classification of the query sample, we take this difference vector as input and generate a query-specific channel weight through a meta-network. QCDM can guide FSC models to focus on discriminative channel differences and achieve better generalization. QCDM is a plug-and-play module that can be seamlessly integrated with existing distance metric learning-based FSC methods. Extensive experimental results indicate that our method can effectively improve the performance of distance metric learning-based FSC methods. The source code is available in https://github.com/Wu-Wenxiao/QCDM.

1. Introduction

The training of deep neural networks usually requires a large amount of labeled data, which however may not be available in newly emerging or rare fields. This issue significantly limits a wider application of deep learning models. In order to solve this problem, few-shot classification (FSC) [1–3] was born on demand, which intends to identify the unseen categories based on only a few labeled samples.

Due to such a small amount of training data, issues such as over-fitting and thus poor generalization often arise in FSC. To address these issues, a variety of methods have been proposed, among which metric-based FSC methods [4–9] have drawn widespread attention. It primarily emphasizes the exploration of effective feature embeddings and their aligned metric functions.

Due to their simplicity and superior performance, distance metrics, such as Euclidean distance and inner product, are commonly employed as the metric function. Take ProtoNet [4], one of the dominant distance metric learning-based FSC methods, as an example. It takes the mean vector of the embedded labeled samples from the same class as the class-prototype and takes the Euclidean distance as the distance metric. Such a distance-based classifier treats each feature channel equally when calculating the distance. However, the equal treatment across channels is unreasonable in FSC, because it ignores the varying importance of different channels.

Recent works such as [10,11] have illustrated that different channels in the learned feature seek for different patterns, which has also been observed in FSC [12]. For example, as shown in Fig. 1 for a 3-way classification task, we should treat different channels differently for different classes of images. When classifying images from the class *dalmatian*, the three channels can be treated similarly due to their similar patterns. However, in classifying the class *mixing bowl*, the second and third channels fail to provide sufficient discriminative power due to their attention on the surrounding environment. In such a case, it becomes imperative to prioritize the first channel and allocate more attention to its contribution. The same situation also exists in

[*] Corresponding author.
E-mail addresses: wenxiaowu@hust.edu.cn (W. Wu), shaoyuanjie@hust.edu.cn (Y. Shao), cgao@hust.edu.cn (C. Gao), jinghao.xue@ucl.ac.uk (J.-H. Xue), nsang@hust.edu.cn (N. Sang).

https://doi.org/10.1016/j.patcog.2024.110380
Received 29 October 2023; Received in revised form 16 January 2024; Accepted 26 February 2024
Available online 2 March 2024

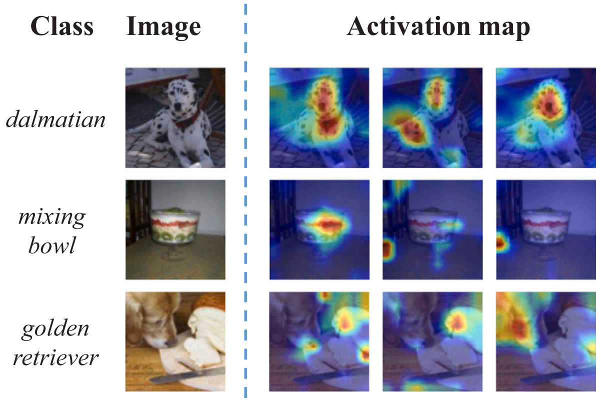


Fig. 1. An example of how different channels are of different importance to classification. In each row, from left to right, the columns present the label of an image from *mini*ImageNet, the image, and its activation maps in three different channels, respectively.

the classification of the class *golden retriever*, in which the third channel should be paid more attention. Recent developments in distance metric learning-based methods tend to focus on enhancing the overall structural aspects of feature extractors [7,13–16] and distance classifiers [17,18]. Although these advancements have yielded substantial improvements, they overlook the crucial issue of varying importance of different feature channels in FSC. It is hence crucial to understand how different channels affect classification in FSC and then tailor specific solutions.

To this end, instead of adjusting the weights of each feature channel directly like most other methods [19–21], we propose a **Q**uery-**C**entric **D**istance **M**odulator (**QCDM**) to dynamically generate the weights assigned to the channel differences between query and support features. In this way, the process of channel weighting can be completely decoupled from the process of feature extraction and, as a result our method can be seamlessly integrated with any given backbone. The inspiration behind our module stems from the VarianceThreshold method in feature selection [22,23]. This method operates in a “zero or one” filter mode, in which a particular attribute will be preserved if its variance is above a preset threshold, with the other attributes discarded. The idea behind this lies in that the more inconsistent the values from different classes on an attribute, the more discriminative this attribute is, which inspires us to generate weights through the inconsistency within each channel of the data from different classes instead of through the correlation between different channels [19,20].

Specifically, for each query sample, we first calculate its query-centric difference matrix by contrasting it with the support samples from different classes. Then, using the values on each matrix column as an input, QCDM generates a specific weight vector over channels for this query sample through a meta-learner. This vector is subsequently applied to modify the distance between the query feature and the support features. By modulating a distance metric with such a weight vector, we can obtain a distance function with desirable adaptability and plasticity, thus effectively improving the generalization to the target novel dataset in FSC.

We summarize our contributions as three-folds:

- In this paper, we turn our attention to a necessary but always neglected factor in distance metric learning-based FSC, which is the weights of channel differences between query and support features. We introduce a lightweight meta-network to generate weights through the inconsistency within the channel of the data from different classes instead of through the interaction between channels.
- Benefiting from its simplicity and flexibility, our approach is a plug-and-play module, which can be seamlessly integrated with any distance classifiers of distance metric learning-based methods for FSC.
- Extensive experiments on several widely-used few-shot learning benchmarks substantiate that our module can significantly enhance the performance of several distance metric learning-based frameworks for FSC.

2. Related work

Many efforts, such as metric-based methods, augmentation-based methods and optimization-based methods, have been devoted to solving the FSC problems. Since our QCDM focuses on distance metrics and channel weighting for FSC, in this section, we only introduce the representative metric-based FSC methods and recent channel-weighting methods in few-shot learning, and their differences from our QCDM.

2.1. Metric-based few-shot classification

Metric-based FSC methods aim to obtain an good embedding space and an aligned distance classifier or module that can measure the category relationships between support and query samples [4,5,13]. For example, Vinyals et al. [13] construct an attention kernel through cosine similarities between query and support images. ProtoNet [4] takes the mean vector of the embedded labeled samples from the same class as the class-prototype and utilizes the Euclidean distance as the metric function. RelationNet [5] employs a CNN-based relation module to measure the similarities between query and support samples. GNN [24] uses graph neural networks for the metric function. DN4 [25] conducts a measure via a k-nearest neighbor search over the local descriptors of feature maps. Liu et al. [16] leverage query samples to diminish bias for prototype rectification. CEN [18] proposes to substitute a feature normalization dissimilarity for Euclidean distance in ProtoNet. DeepEMD [6] uses reconstruction quality, which is measured by transport cost, as a proxy for class membership. FRN [26] reconstructs each query sample from the support set of a class and predicts its class membership through maximum similarity. DeepBDC [7] substitutes Brownian Distance Covariance for the conventional feature of ProtoNet and applies it to both meta-learning and transfer learning frameworks with Euclidean distance as classifier in FSC. LMPNet [8] employs local descriptors to represent each image and generates an embedding space with multiple prototypes. LDP-net [27] establishes a two-branch network to classify the query image and its random local crops respectively, where knowledge distillation is conducted among these two branches to enforce their class affiliation consistency.

2.2. Channel-weighting for few-shot classification

Several works [12,20,28–31] have proposed channel-weighting or attention-based methods to measure the different importance of channels for FSC. Hou et al. [28] introduce a cross attention module to generates weight maps for each pair of class feature and query feature to highlight the target object regions. Xu et al. [29] propose a meta-filter for channel weighting to learn a dynamic alignment, which effectively highlights both query regions and important channels. Luo et al. [12] propose a simple transformation directly on the values of feature channels to make channel distribution smooth: suppress channels with high magnitude, and largely amplify channels with low magnitude. TDM [32] introduces two novels modules: Support Attention Module (SAM) and Query Attention Module (QAM) to learn task-specific channel weights. SAPENet [30] utilizes multi-head self-attention mechanisms to selectively augment discriminative features in each sample feature map. tSF [20] tries to enhance the attention on the foreground of the image through modeling the correlation between learnable embeddings and the input feature. STANet [21] designs a novel transformer-based SpatialFormer structure to generate more accurate attention regions based on global features instead of local features. However, these works always focus on the process of feature extraction, which intensifies the difficulty of applying these methods in a plug-and-play manner and introduces a tremendous mount of additional parameters. Moreover, these approaches tend to generate channel weights through the interaction between channels, therefore ignoring the vast potential of the data inconsistency within each channel.

In our paper, we propose a query-centric distance modulator for FSC. Instead of modify the feature channels in a sophisticated manner like most previous works, we choose to simply modulate the distance metrics by weighting the channel differences between query and support features. In this way, our module can be decoupled from the feature extraction process and can also be trained separately and efficiently. In addition, we innovatively construct a query-centric difference matrix between each query sample and the entire support set and generate weights for each channel difference based on the data inconsistency within each column of this matrix. By modulating distance metrics, we can make the classifier focus more on discriminative patterns and regions in a simple manner and to nearly-effortlessly yet effectively improve the performance of metric learning-based FSC methods.

3. Proposed method

In this section, we first introduce the definition of the FSC problem and the distance metric-based meta-learning framework for FSC. Then we propose our plug-and-play query-centric distance modulator for distance metrics. Finally, we apply the query-centric distance modulator to various distance metric learning-based FSC methods such as ProtoNet [4], FRN [26] and Meta DeepBDC [7].

3.1. Problem definition

We consider the standard few-shot classification problem, in which a train set D_{train} and a test set D_{test} are given. Note that the label space of D_{train} has no overlap with D_{test}. In FSC, the classification model f_θ is trained on a series tasks randomly sampled from D_{train}, and then tested on a series of tasks randomly sampled from D_{test}. Each task $\mathcal{T}_i$ contains two disjoint sets: the support set $\mathcal{S}_i$ and the query set $\mathcal{Q}_i$. Following the "N-way K-shot" setting, $\mathcal{S}_i = \{x_s^i, y_s^i\}_{s=1}^{N\times K}$ is composed of N classes with K images for each class; and $\mathcal{Q}_i = \{x_q^i, y_q^i\}_{q=1}^{N\times M}$ shares the same label space with $\mathcal{S}_i$, consisting of M instances per class. By leveraging the information in $\mathcal{S}_i$, the task $\mathcal{T}_i$ aims to generalize and make accurate predictions on the instances in $\mathcal{Q}_i$.

3.2. Metric-based meta-learning

The meta-learning framework, also known as "learning to learn", tailors specific solutions to the FSC problems. Among these solutions, the metric-based meta-learning framework has gained dominance for FSC due to its simplicity and effectiveness. These methods aim to learn an embedding space parameterized by θ and an appropriate metric function $\mathcal{D}(\cdot)$ that can generalize to new tasks in an episodic way.

With a series of FSC tasks $\{\mathcal{T}_i\}_{i=1}^T$ sampled from D_{train}, the objective of meta-training stage is formulated as

$$\theta^* = \min_\theta \mathop{\mathbb{E}}_{\mathcal{T}_i\sim p(\mathcal{T})} \mathcal{L}(\mathcal{D}(\mathcal{S}_i, \mathcal{Q}_i);\theta), \tag{1}$$

where θ^* is the learned meta-knowledge that can be transferred to new tasks in the stage of meta-testing; and $\mathcal{L}(\cdot)$ is the loss function that evaluates the performance of a model, such as the cross-entropy loss.

In many metric-based meta-learning methods, a fixed distance metric, such as Euclidean distance or inner product, is employed as the metric function $\mathcal{D}(\cdot)$ to measure the dissimilarities between query and support samples. By assuming the feature z of a sample x is $z = f_\theta(x) = \{z^l\}_{l=1}^d \in \mathbb{R}^d$, where z^l is the lth channel or dimension of feature z and d is the number of its dimensions, the distance between query feature z_q and support feature z_s can be calculated as

$$\mathcal{D}(z_q, z_s) = \sum_{l=1}^d \mathcal{D}(z_q^l, z_s^l). \tag{2}$$

After the meta-training procedure in Eq. (1), the learned meta-knowledge θ^* is transferred to an unseen task $\mathcal{T}_{novel}$ sampled from D_{test} and used to evaluate the performance of the trained model.

3.3. Query-centric distance modulator

3.3.1. Motivation

We can find in Eq. (2) that the distance $\mathcal{D}(z_q^l, z_s^l)$ in each channel has the equal weight while calculating the overall distance $\mathcal{D}(z_q, z_s)$. However, as we discussed in Section 1 that different channels should have different importance to FSC. That is, we should adjust the distance metric by incorporating learnable channel weights w_q^l for $l = 1, \dots, d$.

Let us first rewrite Eq. (2) as

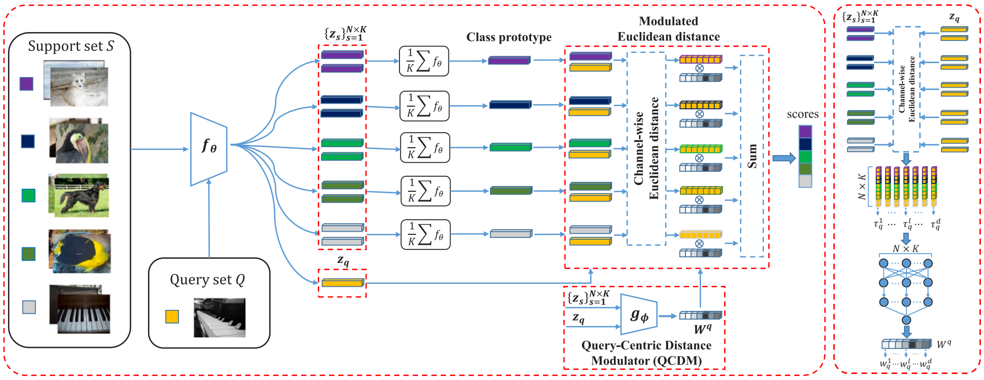
$$\begin{aligned}\mathcal{D}_w(z_q, z_s) &= \sum_{l=1}^d w_q^l \mathcal{D}(z_q^l, z_s^l)\\ &= (W^q)^T[\mathcal{D}(z_q^1, z_s^1), \dots, \mathcal{D}(z_q^l, z_s^l), \dots, \mathcal{D}(z_q^d, z_s^d)]^T,\end{aligned} \tag{3}$$

where the column vector $W^q = (w_q^1, \dots, w_q^l, \dots, w_q^d)^T$, and we note that Eq. (3) is equivalent to Eq. (2) when all $w_q^l = 1$ for $l = 1, \dots, d$. In other words, the conventional distance-based classifier just treats all dimensions of the feature equally. This treatment can potentially result in misclassification, as it fails to adequately capture the different importance of different channels in distinguishing between different classes. Eq. (3) also indicates that, by choosing different values of w_q^l, we can adapt the model's focus to specific channels so that we can make the classification process adaptively. Hence, here our purpose is to adaptively learn a series of w_q^l that can better capture the discriminative channels to improve the performance of existing metric-based FSC methods.

3.3.2. Methodology

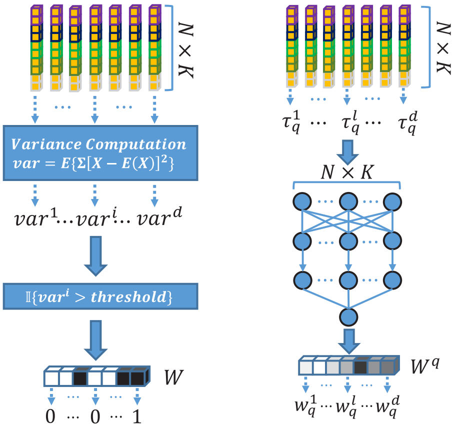
Our channel-weighting strategy was inspired by the VarianceThreshold method in feature selection [22]. The VarianceThreshold method utilizes the variance of each attribute as a statistical measure to filter those attributes whose variance is below a threshold. The underlying principle behind this method is that the more inconsistent the values of an attribute from different classes are, the more discriminative the attribute is. However, as we know, the variance itself is greatly influenced by the scale of values, so using it as weight w_q^l directly can have adverse effects. Specific analysis can be seen in Section 5. Here, as shown in Fig. 3, we compare the Variancethreshold method and our QCDM.

When it comes to the few-shot problem, please note that our objective is that the weight w_q^l should reflect the relative importance of the channels regarding the differences between the query feature and the support features. Hence, we construct a query-centric difference matrix for each query sample, by comparing the query sample with support samples from different classes. The reason that we construct this matrix is that the values τ_q^l within the lth matrix column are the distances between the query feature and the support features from different classes within the lth channel, thus they can reflect the distance inconsistency within this channel. That is, we follow the principle from the VarianceThresh method: the more inconsistent the data from different classes on a dimension are, the more discriminative this dimension is. In addition, we use the values τ_q^l to generate the query-specific weight w_q^l through a meta-learner.


Support set S
Query set Q
f_θ
$\{z_s\}_{s=1}^{N\times K}$
z_q
Class prototype
$\frac{1}{K}\sum f_\theta$
Modulated Euclidean distance
Channel-wise Euclidean distance
Sum
scores
$\{z_s\}_{s=1}^{N\times K}$
z_q
g_ϕ
W^q
Query-Centric Distance Modulator (QCDM)
Channel-wise Euclidean distance
$N\times K$
$\tau_q^1 \cdots \tau_q^l \cdots \tau_q^d$
$N\times K$
W^q
$w_q^1 \cdots w_q^l \cdots w_q^d$

(a) ProtoNet+QCDM (b) QCDM

Fig. 2. (a) The illustration of our query-centric distance modulator integrated with ProtoNet [4] for a 5-way 2-shot problem with one query sample. We split the computation of Euclidean distance into the processes of "Channel-wise Euclidean distance" and "Sum", which correspond to the $\mathcal{D}(z_q^l, z_s^l)$ and $\Sigma_{l=1}^d$ operations in Eq. (3), respectively. We take the query-centric difference matrix between the query feature z_q and the whole support set as input and generate the query-centric weight vector W^q through a meta-learner g_ϕ. The weight vector W^q is then used to modulate the distances between z_q and class prototypes and force the model to pay more attention to discriminative channels. (b) A more detailed diagram of our QCDM.


$N\times K$
Variance Computation
$var = E\{\Sigma[X - E(X)]^2\}$
$var^1 \ldots var^i \ldots var^d$
$\mathbb{I}\{var^i > threshold\}$
W
0 ··· 0 ··· 1
$N\times K$
$\tau_q^1 \cdots \tau_q^l \cdots \tau_q^d$
$N\times K$
W^q
$w_q^1 \cdots w_q^l \cdots w_q^d$

(a) VarianceThreshold (b) QCDM

Fig. 3. The diagrammatic comparison between the VarianceThreshold method and our QCDM.

More specifically, in the "N-way K-shot" setting, the query-centric matrix τ_q for the q-th query sample with d channels is formulated as

$$\tau_q = \begin{bmatrix}\tau_q^1 & \cdots & \tau_q^l & \cdots & \tau_q^d\end{bmatrix} \in \mathbb{R}^{NK\times d}$$
$$= \left[\begin{pmatrix} \mathcal{D}(z_q^1, z_1^1) \\ \vdots \\ \mathcal{D}(z_q^1, z_s^1) \\ \vdots \\ \mathcal{D}(z_q^1, z_{NK}^1) \end{pmatrix} \cdots \begin{pmatrix} \mathcal{D}(z_q^l, z_1^l) \\ \vdots \\ \mathcal{D}(z_q^l, z_s^l) \\ \vdots \\ \mathcal{D}(z_q^l, z_{NK}^l) \end{pmatrix} \cdots \begin{pmatrix} \mathcal{D}(z_q^d, z_1^d) \\ \vdots \\ \mathcal{D}(z_q^d, z_s^d) \\ \vdots \\ \mathcal{D}(z_q^d, z_{NK}^d) \end{pmatrix}\right], \tag{4}$$

where z_q^l and z_s^l represent the lth channel of the q-th query feature z_q and the s-th support feature z_s, respectively. By considering that the values τ_q^l reflect the distribution of differences in the lth channel, we can generate the weight for the lth dimension of z_q through a meta-learner g parameterized by ϕ as follows:

$$w_q^l = g_\phi(\tau_q^l), \ l = 1, \ldots, d. \tag{5}$$

It follows that the query-centric weights for query sample with d channels can be represented as

$$W^q = [g_\phi(\tau_q^1), \ldots, g_\phi(\tau_q^l), \ldots, g_\phi(\tau_q^d)]^T \in \mathbb{R}^d. \tag{6}$$

Subsequently, these weights are utilized to modulate the distance metric and indirectly control the base learner update.

Algorithm 1 ProtoNet+Query-Centric Distance Modulator

```
Require: Task distribution $p(\mathcal{T})$, initial parameters $\theta$ and $\phi$, learning rate $\eta$,
    distance metric $\mathcal{D}(\cdot)$
while training do
    Sample a task $\mathcal{T} = \{\mathcal{S}, \mathcal{Q}\}$ from $p(\mathcal{T})$ where $\mathcal{S} = \{x_s, y_s\}_{s=1}^{N\times K}, \mathcal{Q} = \{x_q, y_q\}_{q=1}^{N\times M}$
    for $(x_s, y_s)$ in $\mathcal{S}$ do
        Compute the feature representation of $x_s$: $z_s = f_\theta(x_s)$
    end for
    Initialize task loss: $\mathcal{L}_{task} = 0$
    for class $n \in [1, N]$ do
        Compute the class prototype: $c_n = \frac{1}{K}\sum_{y_s=n} f_\theta(x_s)$
        for $(x_q, y_q)$ in $\mathcal{Q}$ do
            Compute the feature representation of $x_q$: $z_q = f_\theta(x_q)$
            Compute the query-specific matrix for $x_q$: $\tau_q = [\tau_q^1 \cdots \tau_q^l \cdots \tau_q^d]$
            Compute the query-centric weight vector for $z_q$: $W^q = [g_\phi(\tau_q^1), \cdots, g_\phi(\tau_q^l), \cdots, g_\phi(\tau_q^d)]^T$
            Modulate the distance between $z_q$ and class prototype $c_n$:
            $\mathcal{D}_w(z_q, c_n) = \Sigma_{l=1}^d g_\phi(\tau_q^l)\mathcal{D}(z_q^l, c_n^l)$
            Compute the query loss with modulated distance:
            $\mathcal{L}_{CE} = -\frac{1}{NM}\mathbb{I}\{y_q = n\}[\mathcal{D}_w(z_q, c_n) - \log\sum_{n'} \exp(\mathcal{D}_w(z_q, c_{n'}))]$
            Update task loss: $\mathcal{L}_{task} = \mathcal{L}_{task} + \mathcal{L}_{CE}$
        end for
    end for
    Perform gradient descent to update parameters:
    $(\theta, \phi) \leftarrow (\theta, \phi) - \eta\nabla_{(\theta,\phi)}\mathcal{L}_{task}$
end while
Ensure: The final parameters $\theta$, $\phi$.
```

3.3.3. Diagram and algorithm

The diagram of our proposed query-centric distance modulator integrated with ProtoNet is presented in Fig. 2(a). Please note that our approach operates on the distance metric only and thus does not directly interfere with the feature extraction process. Therefore, our method can be inserted into any distance metric learning-based FSC method with any backbone. As is shown in Fig. 2(b), g_ϕ is a 2-layer Multilayer Perception (MLP) with two *fully-connected* layers denoted as $fc_1(\cdot)$ and $fc_2(\cdot)$. We use the ReLU function as activation between layers. The generation procedure of w_q^l can be presented as

$$w_q^l = fc_1(ReLU(fc_2(\tau_q^l))). \tag{7}$$

The update of base learner f_θ and weight generator g_ϕ using the modulated distance metric $\mathcal{D}_w(\cdot)$ is performed as in

$$(\theta, \phi) \leftarrow (\theta, \phi) - \eta\nabla_{(\theta,\phi)}\mathcal{L}(\mathcal{D}_w(\mathcal{S}_i, \mathcal{Q}_i); \theta; \phi). \tag{8}$$

The overall training procedure in conjunction with ProtoNet is outlined in Algorithm 1.

Moreover, the number of parameters in the modulator is exclusively relevant to the number of support samples regardless of the depth of the backbone. Assume a "N-way K-shot" few-shot classification problem. Since the parameters of g_ϕ are shared across all the tasks, it takes the NK-dimensional vector τ_q^l as input, which is the same as the number of hidden units for intermediate layers. It can be seen that the overall number of learnable parameters added is minimal with $(NK+1)^2$, where N and K are the numbers of class and support samples per class, respectively. We also present the information on overall model parameter changes in Table 5 of Section 5.2. Therefore, our query-centric distance modulator is a lightweight network, which consumes almost no computational resources.

4. Experiments

In this section, we first elaborate on the implementation details and settings of our experiments. Then, under the settings of within-domain and cross-domain, we insert our proposed query-centric distance modulator into three popular distance metric learning-based FSC frameworks: FRN [26], ProtoNet [4] and Meta DeepBDC [7], and compared these modulated versions with some state-of-the-art (SOTA) FSC algorithms on several benchmarks.

4.1. Implementation details

In our experiments, we use two kinds of network architecture, ResNet-12 [33] and ResNet-18 [34], as the base learner and implement our method within PyTorch. For fair comparisons with previous methods, we take images with a resolution of 84 × 84 as input for ResNet-12 and 224 × 224 for ResNet-18. Following the protocol used previously in [1,7], we pre-train a feature extractor as the initialization of base learner with the cross-entropy loss on the corresponding training classes. Please note that the test classes share no joint label space with the training classes. Moreover, we apply our plug-and-play query-centric distance modulator to FRN [26], ProtoNet [4] and Meta DeepBDC [7].

For FRN [26] and ProtoNet [4], all their official implementations are trained under an over 5-way setting at the stage of meta-training, e.g. sampling 20-way tasks for training. For fair comparison, we unify the class number of each few-shot tasks to 5 for both training and testing and present our reproduced results. During the training phase, similar to all the meta-learning frameworks, our query-centric distance modulator acts in an episodic manner. Each episode follows the "N-way K-shot" setting, sampling K labeled support samples and 15 query samples randomly from N classes from the training set. We apply a series of standard data augmentation, including random crop, left–right flip, and color jitter in the meta-training stage. All the experiments use the SGD optimizer with a momentum of 0.9 and a weight decay of 0.0005 with its learning rate η set to 10^{-4} for the ImageNet-based datasets and 10^{-3} for CUB. During the evaluation phase, we also randomly pick K labeled support samples and 15 query samples for each class but from the test set. For each benchmark, we average the results of 2000 independent few-shot classification tasks with 95% confidence interval.

4.2. Evaluation under the within-domain setting

In order to verify the effectiveness of our proposed module, we conduct experiments under the standard within-domain setting of "5-way 1/5-shot" classification. We compare the results from our modulated ProtoNet and Meta DeepBDC frameworks with some SOTA methods. We evaluate the performance on the following three standard benchmarks.

***mini*ImageNet:** The *mini*ImageNet [13] is a subset sampled from ImageNet [35], consisting of 600 images each for all the 100 classes. Following the split provided by [36], we split these classes into 64-16-20 for meta-training, meta-validation, and meta-testing, respectively.

***tiered*ImageNet:** The *tiered*ImageNet [37] is a more significant subset of ImageNet containing 608 classes from 34 super-classes and a total of 779,165 images, which, compared with *mini*ImageNet, also considers the hierarchical structure of ImageNet. For meta-training, 20 super-classes (351 classes) are selected, while 6 super-classes (91 classes) are chosen for meta-validation and 8 super-classes (160 classes) for meta-testing.

CUB: The CUB dataset [38] is composed of 200 bird classes with 11,788 images in total. Following the setting in [1,7], all the experiments for CUB are conducted with the original raw images rather than cropped images by annotated bounding boxes [6,39]. As well as the split in [1], the ratio of the number of classes involved in training, validation and testing set is set to 2:1:1.

The results of 5-way 5-shot and 5-way 1-shot classification on *mini*ImageNet, *tiered*ImageNet and CUB are presented in Tables 1 and 2. Here are several observations. First of all, both ProtoNet and Meta DeepBDC achieve significant performance gains on all the datasets under 5-way-1-shot and 5-way-5-shot settings with our proposed module QCDM inserted, which confirms the validity of modulating distance metrics by weighting channel differences. For example, on *mini*ImageNet, our modulator improves the performance of ProtoNet for the 5-shot classification from 80.77% to 83.28% and achieves a nearly 4% increase on the 1-shot classification tasks. Secondly, our modulated Meta DeepBDC outperforms all the competing methods including some SOTA methods such as H-OT [49] on *tiered*ImageNet and Liu et al. [55] on CUB under 5-way-1-shot and 5-way-5-shot settings, which also illustrates the superiority of our method. Thirdly, the performance improvement on FRN is not as significant as that on ProtoNet and Meta DeepBDC. The explanation behind this may be that the reconstructed "prototype" for classification in FRN is already in a weighted state to be similar with query samples, which may somehow limit further improvement that can be obtained with our idea of generating weights through inconsistencies in data from different classes. Fourthly, compared with current attention-based or transformer-based methods such as tSF [20] and STANet [21] which introduce additional million-level parameters, our QCDM+Meta DeepBDC can beat them with only thousands-level parameters introduced. Last but only least, it can be seen in Table 2 that our QCDM improves the baseline methods a lot with both ResNet-12 and ResNet-18 as the backbone. The results reveal that our method can work with more advanced feature extractors to perform better, demonstrating its flexibility and versatility.

4.3. Evaluation under the cross-domain setting

To further verify the effectiveness of our method, we consider a more challenging and practical cross-domain setting introduced by [1], where the base and novel classes are sampled from significantly different domains. We conduct our experiments under the following two scenarios.

***mini*ImageNet → CUB:** This cross-domain setting was first introduced by [1], further exploring the issue of domain difference in few-shot classification. We perform meta-training on all the *mini*ImageNet classes and test on the test set of CUB following the split in [1,7].

***mini*ImageNet → Cars:** Cars [56] is composed of 196 classes with 16,185 images in total. There are two split rules in this setting: the split from [7,25] and the split from [57–59] (common in prior work). In our paper, we present the results following the split from [7,25].

Table 1
Few-shot classification accuracy (%) with 95% confidence interval on *mini*ImageNet and *tiered*ImageNet under 5-way 1/5-shot scenarios. The best results are in bold [40–48].

Methods	Backbone	Reference	*mini*ImageNet		*tiered*ImageNet	
			1-shot	5-shot	1-shot	5-shot
CovNet [15]	ResNet-12	CVPR'19	64.59 ± 0.45	82.02 ± 0.29	69.75 ± 0.52	84.21 ± 0.26
DN4[b] [25]	ResNet-12	CVPR'19	57.76	77.57	64.41	82.59
CAN[b] [28]	ResNet-12	NIPS'19	66.62	78.96	70.46	84.50
ALFA[a] [40]	ResNet-12	NIPS'20	63.66 ± 0.44	78.73 ± 0.32	64.62 ± 0.49	82.48 ± 0.38
GNN+FT [41]	ResNet-10	ICLR'20	66.32 ± 0.80	81.98 ± 0.55	–	–
Good-Embed [33]	ResNet-12	ECCV'20	64.82 ± 0.60	82.14 ± 0.43	71.52 ± 0.69	82.48 ± 0.38
FEAT [39]	ResNet-12	CVPR'20	66.78 ± 0.20	82.05 ± 0.14	70.80 ± 0.23	86.03 ± 0.58
DeepEMD [6]	ResNet-12	CVPR'20	65.91 ± 0.82	82.41 ± 0.56	71.16 ± 0.87	86.03 ± 0.58
ADM [17]	ResNet-12	IJCAI'20	65.87 ± 0.43	82.05 ± 0.29	70.78 ± 0.52	85.70 ± 0.43
RENet[b] [42]	ResNet-12	ICCV'21	66.83	82.13	70.14	82.70
ALFA+MeTAL [43]	ResNet-12	ICCV'21	66.61 ± 0.28	81.43 ± 0.25	70.29 ± 0.40	86.17 ± 0.35
BLC-MAML [44]	ResNet-12	TCSVT'22	66.26 ± 0.32	82.92 ± 0.30	–	–
Xu et al. [29]	ResNet-12	CVPR'21	67.76 ± 0.46	82.71 ± 0.31	71.89 ± 0.52	85.96 ± 0.35
Sum-min [45]	SF-12	CVPR'22	68.32 ± 0.62	82.71 ± 0.46	73.63 ± 0.88	87.59 ± 0.57
MCL-Katz [46]	ResNet-12	CVPR'22	67.85	84.47	72.13	86.32
PatchProto+tSF [20]	ResNet-12	CVPR'22	69.74 ± 0.47	83.91 ± 0.30	71.98 ± 0.50	85.49 ± 0.35
DMN4 [47]	ResNet-12	AAAI'22	66.58	83.52	72.10	85.72
H-OT [49]	ResNet-12	NIPS'22	65.63 ± 0.32	82.87 ± 0.43	73.71 ± 0.26	87.46 ± 0.35
ProtoNet[a]+Simple [12]	ResNet-12	ICML'22	–	81.31 ± 0.43	–	–
STANet [21]	ResNet-12	AAAI'23	**69.84 ± 0.47**	84.88 ± 0.30	73.08 ± 0.49	86.80 ± 0.34
PDN-PAS [9]	ResNet-50	PR'23	66.38 ± 0.82	84.29 ± 0.56	71.13 ± 0.97	85.75 ± 0.66
SAPENet [30]	ResNet-12	PR'23	66.41 ± 0.20	82.76 ± 0.14	68.63 ± 0.23	84.30 ± 0.16
GLFA [48]	ResNet-12	PR'23	67.25 ± 0.36	82.80 ± 0.30	72.25 ± 0.40	86.37 ± 0.27
ProtoNet [4]	ResNet-12	NIPS'17	62.11 ± 0.44	80.77 ± 0.30	68.31 ± 0.51	83.85 ± 0.36
ProtoNet+Ours	ResNet-12	–	66.08 ± 0.44	83.28 ± 0.38	71.89 ± 0.49	86.99 ± 0.33
FRN[a] [26]	ResNet-12	CVPR'21	66.27 ± 0.19	82.65 ± 0.13	68.72 ± 0.23	85.81 ± 0.15
FRN+Ours	ResNet-12	–	66.97 ± 0.20	83.40 ± 0.13	69.41 ± 0.22	86.29 ± 0.15
Meta DeepBDC [7]	ResNet-12	CVPR'22	67.34 ± 0.43	84.46 ± 0.28	72.34 ± 0.49	87.31 ± 0.31
Meta DeepBDC[a] [7]	ResNet-12	CVPR'22	67.37 ± 0.44	84.06 ± 0.29	73.01 ± 0.49	87.74 ± 0.31
Meta DeepBDC+Ours	ResNet-12	–	67.96 ± 0.44	**85.22 ± 0.27**	**74.74 ± 0.48**	**88.78 ± 0.30**

[a] Reproduced under our settings.

[b] Reproduced by LibFewShot [2].

Table 2
Few-shot classification accuracy (%) with 95% confidence interval on CUB under 5-way 1/5-shot scenarios. The best results are in bold [44,45,48,50–54].

Methods	Backbone	Reference	CUB	
			1-shot	5-shot
DeepEMD [6]	ResNet-12	CVPR'20	75.65 ± 0.82	88.69 ± 0.50
BLC-MAML [44]	ResNet-12	TCSVT'22	82.21 ± 0.42	90.99 ± 0.23
MetaNODE(Induc.) [50]	ResNet-12	AAAI'22	80.82 ± 0.75	91.77 ± 0.49
Sum-min [45]	SF-12	CVPR'22	79.60 ± 0.80	90.48 ± 0.44
CAD [51]	ResNet-12	CVPR'22	82.95 ± 0.67	90.80 ± 0.51
STL DeepBDC [7]	ResNet-18	CVPR'22	84.01 ± 0.42	94.02 ± 0.22
H-OT [49]	ResNet-12	NIPS'22	76.28 ± 0.40	89.87 ± 0.36
Tang et al. [52]	ResNet-12	PR'22	78.73 ± 0.84	89.77 ± 0.47
RankDNN [53]	ResNet-12	AAAI'23	82.93	91.47
SPAENet [30]	Conv4-64	PR'23	70.38 ± 0.23	84.47 ± 0.14
Liu et al. [55]	ResNet-12	PR'23	83.93 ± 0.66	93.95 ± 0.30
GLFA [48]	ResNet-12	PR'23	76.52 ± 0.37	90.27 ± 0.38
QGN [54]	WRN-28-10	PR'23	84.15	91.86
ProtoNet[a] [4]	ResNet-12	NIPS'17	80.07 ± 0.44	89.36 ± 0.23
ProtoNet+Ours	ResNet-12	–	82.45 ± 0.43	91.71 ± 0.21
ProtoNet [4]	ResNet-18	NIPS'17	80.90 ± 0.43	89.81 ± 0.23
ProtoNet+Ours	ResNet-18	–	82.12 ± 0.44	91.63 ± 0.22
FRN[a] [26]	ResNet-12	CVPR'21	78.34 ± 0.22	92.59 ± 0.10
FRN+Ours	ResNet-12	–	79.77 ± 0.20	93.13 ± 0.10
Meta DeepBDC[a] [7]	ResNet-12	CVPR'22	81.94 ± 0.41	92.76 ± 0.19
Meta DeepBDC+Ours	ResNet-12	–	83.66 ± 0.40	93.11 ± 0.18
Meta DeepBDC [7]	ResNet-18	CVPR'22	83.55 ± 0.40	93.82 ± 0.17
Meta DeepBDC[a] [7]	ResNet-18	CVPR'22	83.36 ± 0.40	93.60 ± 0.17
Meta DeepBDC+Ours	ResNet-18	–	**84.28 ± 0.39**	**94.12 ± 0.17**

[a] Reproduced under our settings.

We summarize all the cross-domain results in Table 3. The experimental results in Table 3 exhibit a similar tendency to the within-domain results in Tables 1 and 2 that the methods further modulated by QCDM show a clear advantage over baseline methods. Our QCDM provides an average of 2.3% improvement on ProtoNet and 2.8% on Meta DeepBDC. It indicates that our QCDM can enable the model

Table 3
Cross-domain few-shot classification accuracy (%) on CUB and Cars with 95% confidence interval under 5-way 1/5 shot scenarios. We apply our method to ProtoNet and Meta DeepBDC and compare with other competitive methods. The best results are in bold [41,42,60,61].

Methods	Backbone	Reference	*mini*ImageNet→CUB		*mini*ImageNet→Cars	
			1-shot	5-shot	1-shot	5-shot
Baseline [1]	ResNet-18	ICLR’19	–	65.57 ± 0.70	–	50.29 ± 0.37
Baseline++ [1]	ResNet-18	ICLR’19	–	62.04 ± 0.76	–	46.44 ± 0.37
CovNet [15]	ResNet-12	CVPR’19	–	76.77 ± 0.34	–	52.90 ± 0.37
DN4[b] [25]	ResNet-12	CVPR’19	42.77	61.73	29.08	43.66
CAN[b] [28]	ResNet-12	NIPS’19	43.94	62.37	29.09	39.16
ADM [17]	ResNet-12	IJCAI’20	–	70.55 ± 0.43	–	53.94 ± 0.35
Good-Embed [33]	ResNet-12	ECCV’20	–	67.43 ± 0.44	–	50.18 ± 0.37
GNN+FT [41]	ResNet-10	ICLR’20	47.47 ± 0.75	66.98 ± 0.68	31.61 ± 0.53	44.90 ± 0.64
RENet[b] [42]	ResNet-12	ICCV’21	48.69	65.79	31.09	44.45
TPN+ATA [60]	ResNet-10	IJCAI’21	50.26 ± 0.50	65.31 ± 0.40	34.18 ± 0.40	46.95 ± 0.40
CDCSD [61]	ResNet-12	PR’23	54,71 ± 0.64	76.26 ± 0.55	36.22 ± 0.54	52.91 ± 0.56
Liu et al. [55]	ResNet-12	PR’23	53.14 ± 0.69	73.02 ± 0.59	**37.73 ± 0.53**	51.82 ± 0.58
ProtoNet[a] [4]	ResNet-12	NIPS’17	47.72 ± 0.46	67.19 ± 0.38	30.56 ± 0.31	46.30 ± 0.36
ProtoNet+Ours	ResNet-12	–	49.40 ± 0.45	70.09 ± 0.38	33.21 ± 0.34	48.21 ± 0.36
FRN[a] [26]	ResNet-12	CVPR’21	51.97 ± 0.21	72.52 ± 0.18	34.61 ± 0.14	53.47 ± 0.16
FRN+Ours	ResNet-12	–	52.43 ± 0.21	73.43 ± 0.18	35.10 ± 0.15	53.80 ± 0.15
Meta DeepBDC [7]	ResNet-12	CVPR’22	–	77.87 ± 0.33	–	54.61 ± 0.37
Meta DeepBDC[a] [7]	ResNet-12	CVPR’22	50.10 ± 0.49	77.81 ± 0.33	35.42 ± 0.42	53.88 ± 0.39
Meta DeepBDC+Ours	ResNet-12	–	**57.26 ± 0.45**	**79.02 ± 0.33**	36.71 ± 0.43	**55.48 ± 0.39**

[a] Reproduced under our settings.
[b] Reproduced by LibFewShot [2].

to pay attention to more discriminative channels with the adaptive query-centric weights and the ability can be transferred to new tasks sampled from the target domain. Furthermore, a significant performance enhancement is observed in our approach, specifically in the *mini*ImageNet→CUB scenario using the Meta DeepBDC architecture, where the performance improves from 50.10% to 57.26% under the 1-shot setting. These results show that methods with our QCDM are less affected by domain shift, even when base classes are sampled from a coarse-grained dataset while novel classes from a fine-grained dataset. In fact, the good results are attributed to our unique weight generation mechanism. Our QCDM aims to generate channel weights through the data inconsistency from different classes within each column of the difference matrix rather than the correlation across channels of features. In this way, although the features of samples are severely biased due to the issue of domain shift, all the elements of the query-centric difference matrix τ_q in Eq. (4) will be biased in a similar way. Therefore, the data inconsistency would not be much affected by the feature bias, through which accurate weights can also be generated.

5. Ablation study and qualitative analysis

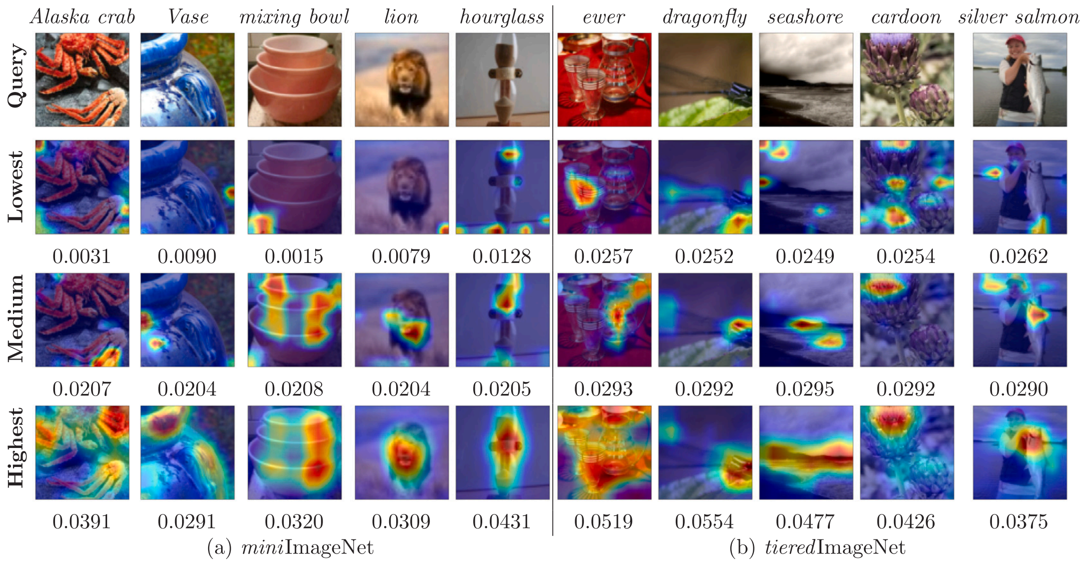
5.1. Whether important channels are assigned with higher weights?

First of all, we select the 10 channels with the highest/medium/lowest weights generated by our QCDM based on ProtoNet, presenting the mean value of weights and corresponding class activation maps of these channels on *mini*ImageNet and *tiered*ImageNet in Fig. 4, respectively. It can be easily observed that channels with lower weights tend to pay attention to the background or less important parts, which are relatively “unimportant” intuitively, while channels with higher weights focus more on the objects to be classified.

Apart from this, we also conduct an experiment where the 10/20/40/80/16 0/320 channels with the highest or lowest weights are selected for classification under the 5-way 5-shot setting on ProtoNet. The results are shown in Table 4. As illustrated in Table 4, ProtoNet with only 20 highest channels selected and weighted for classification can achieve comparable performance to MAML on *mini*ImageNet (74.19%) and *tiered*ImageNet (71.24%) while with 320 channels selected and weighted can perform better than the original ProtoNet (80.77%/83.98%) with all the 640 channels not weighted. Therefore, it is of great belief that different channels are of different importance to the tasks and our method QCDM can actually find the relatively more important channels.

5.2. Discussions on different forms of meta-learner

In addition to using MLP as a form of meta-learner in the main paper, we also explore the usage of Convolutional Layer and Channel Attention Module in CBAM [62] as the meta-learner. The architecture of the meta-learner in the form of Convolutional Layer is a convolutional module with ReLU as the activation function between the two Convolutional Layers $c_1(\cdot)$ and $c_2(\cdot)$ and an AvgPooling layer at the end. We employ 3 × 3 convolutional kernels for each convolutional layer. In this way, the module will not generate weights channel by channel but will take the whole difference matrix as input to generate weights for all channels simultaneously: $W^q = g_\phi(\tau_q) = AvgPool(ReLU(c_2(ReLU(c_1(\tau_q)))))$. In addition, for another architecture of Channel Attention, MaxPooling and AvgPooling are first applied at each spatial position of the query-centric matrix τ_q. Then, the outputs of MaxPooling and AvgPooling are fed into a shared MLP containing two full-connected layers. Finally, elements in both outputs are added and fused, and a Sigmoid activation function is used to obtain the channel attention weights. The formula is expressed as $W^q = g_\phi(\tau_q) = \sigma(MLP(AvgPool(\tau_q)) + MLP(MaxPool(\tau_q)))$. Table 5 also illustrates an improvement on ProtoNet with our QCDM using Convolutional Layer and Channel Attention Module as different forms of meta-learner. Moreover, it can be seen in Table 5 that QCDM with MLP provides more increase on the 5-way 5-shot classification tasks than the QCDM with Conv and Attention. The reason behind this can be that the QCDM with Conv focuses only on a part rather than the whole of differences for each channels due to its limited receptive field, while the QCDM with MLP pays attention to the whole distribution of each dimension so that it can reflect the discriminative power of each channel more accurately. Additionally, even with a large amount of extra parameters added, QCDM with attention mechanism fails to surpass QCDM with MLP. The reason can be that attention mechanism leads to interaction across channels which may conflict with out motivation to generate channel weights through the data inconsistency from different classes within



(a) *mini*ImageNet (b) *tiered*ImageNet

Fig. 4. The Grad-Cam++ class activation maps of the 10 highest/medium/lowest channels on *mini*ImageNet and *tiered*ImageNet. The average weights of the 10 channels generated by QCDM are presented below the corresponding maps. The first five columns come from *mini*ImageNet and the last five columns are from *tiered*ImageNet.

Table 4
ProtoNet with the 10–320 highest or lowest channels selected for classification under the 5-way 5-shot setting.

Highest/lowest	Dataset	10	20	40	80	160	320
Highest	*mini*ImageNet	71.98%	74.82%	77.18%	78.84%	79.82%	80.79%
Lowest	*mini*ImageNet	43.92%	54.12%	62.87%	69.10%	73.22%	78.10%
Highest	*tiered*ImageNet	72.32%	76.93%	80.58%	82.62%	84.72%	85.97%
Lowest	*tiered*ImageNet	49.71%	56.99%	63.71%	69.67%	74.52%	79.64%

Table 5
The comparison results of ProtoNet+QCDM using Convolutional Layer, Channel Attention Module and MLP as different forms of meta-learner on *mini*ImageNet. The best results are in bold.

Methods	Params		*mini*ImageNet		*tiered*ImageNet	
	1-shot	5-shot	1-shot	5-shot	1-shot	5-shot
ProtoNet [4]	12.4M	12.4M	62.11 ± 0.44	80.77 ± 0.30	68.31 ± 0.51	83.85 ± 0.36
+QCDM (Conv)	+0.1K	+0.1K	65.92 ± 0.44	82.46 ± 0.31	71.31 ± 0.50	85.73 ± 0.34
+QCDM (Attention)	+819.2K	+819.2K	64.98 ± 0.44	82.82 ± 0.31	70.70 ± 0.51	86.56 ± 0.33
+QCDM (MLP)	+0.1K	+0.7K	**66.08 ± 0.44**	**83.28 ± 0.38**	**71.89 ± 0.49**	**86.99 ± 0.33**

the same channel instead of interaction across channels. Furthermore, an explanation why the results of ProtoNet+QCDM with MLP or Conv on 5-way 1-shot classification tasks are similar is that the input of each channel is only a 5-dimensional vector so that the limited receptive field can capture discriminative information as well.

5.3. Whether QCDM works for different distance metrics?

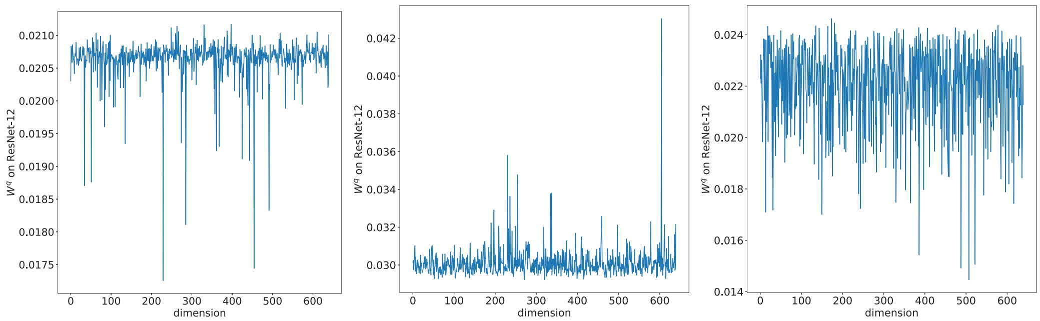
To explore whether our method can capture the more important channels with different metrics, we present a list of comparison results on Meta DeepBDC with inner product, cosine similarity, and Euclidean distance for 5-way 1-shot classification tasks on *mini*ImageNet. From Table 6, we can find that Meta DeepBDC chooses the inner product as distance metric under the 1-shot setting because of its superior performance. Our query-centric distance modulator narrows the performance gap between different distance metrics and notably enhances the performance of cosine similarity to a level comparable to inner product. These results also demonstrate the universality and effectiveness of our method, which can be inserted into different distance metrics. As the latency increase before and after the application of QCDM is really small, our module achieves significant performance improvements without incurring substantial time or computational costs.

Table 6
Ablation analysis of Meta DeepBDC before and after the embedding of our QCDM with different distance metrics. We report accuracy and latency (ms) for 5-way 1-shot tasks on *mini*ImageNet. Latency is measured with a GeForce RTX 3090.

Distance metric	Meta DeepBDC		+QCDM	
	Acc	Latency	Acc	Latency
Inner product	67.37 ± 0.44	69	**67.96 ± 0.44**	73
Cosine similarity	61.74 ± 0.42	69	67.11 ± 0.45	74
Euclidean distance	56.70 ± 0.45	70	59.28 ± 0.45	72

5.4. Visualization of generated weights for different domains

We examine the mean values of weights generated by QCDM on different domains to validate whether it indeed focuses on different channels for different query samples as assumed. We average all the weights generated by QCDM for query samples from 2000 different test tasks on *mini*ImageNet, *tiered*ImageNet and CUB in Fig. 5(a), (b) and (c) respectively. As illustrated in Fig. 5, the generated weights differ for each dimension or channel under different domains (*mini*ImageNet, tieredImageNet and CUB). An observation is that the generated weights and the channels that the modulator focuses on are significantly different for query samples from different domains. It is in belief that our



(a) *mini*ImageNet (b) *tiered*ImageNet (c) CUB

Fig. 5. Visualization of generated weights across all the dimensions for ResNet-12 on ProtoNet.

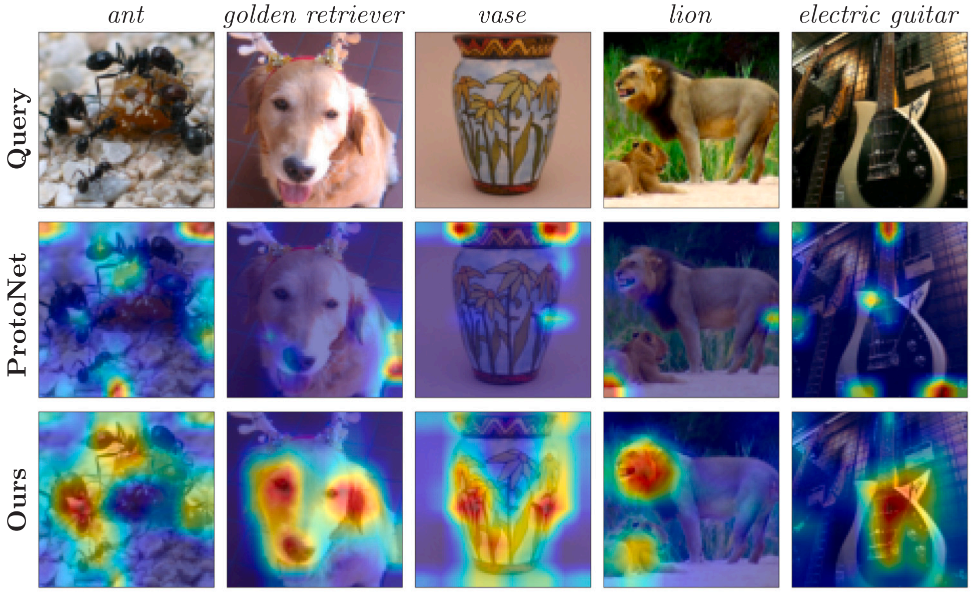


Fig. 6. The Grad-Cam++ [63] class activation maps of query samples without or with our query-centric distance modulator module under the framework of ProtoNet on 5-way 5-shot classification sampled from *mini*ImageNet. The first row illustrates the original images and the second row illustrates the class activation maps of the original ProtoNet, while the third row represents the results of ProtoNet modulated by our module.

Table 7
The comparison results of ProtoNet with Euclidean distance modulated by variance or QCDM on *mini*ImageNet. The best results are in bold.

Methods	*mini*ImageNet	
	1-shot	5-shot
ProtoNet	62.11 ± 0.44	80.77 ± 0.30
ProtoNet+Var	59.30 ± 0.43	79.01 ± 0.32
ProtoNet+QCDM	**66.08** ± **0.44**	**83.28** ± **0.38**

QCDM is aware of the domain differences and has made corresponding adjustments for better generalization.

5.5. The comparison between using generated weights and directly using variances directly as weights

In order to further illustrate the effect of generated weights and the reason why using the variance itself as attribute weights has neglect effects, we present the comparison results of ProtoNet with Euclidean distance modulated by variances or generated weights on *mini*ImageNet in Table 7. It can be seen that using the variance as weight directly can have a negative effect. This outcome can be attributed to the sensitivity of variance to the scale of data and noise, whereby the presence of high variance may arise more from the dimension's large-scale values rather than from significant differences.

5.6. The grad-Cam++ visualization on ProtoNet.

In Fig. 6, we compare some class activation maps using ProtoNet before and after the insertion of our QCDM on *mini*ImageNet. We can observe that modulating distance metrics helps the model adjust the attention to the objects and thus can benefit the classification with only a few labeled images. The equal treatment of different channels on different tasks makes models confused about which part to pay attention to, while our adaptive adjustment of channel weights emphasizes the parts of interest.

6. Conclusion

This paper introduces QCDM, a query-centric distance modulator, to improve the existing distance metric learning-based methods for few-shot classification. We propose to learn from the query-centric difference matrix of each query feature and the whole support set to adaptively generate a query-specific weight that can represent the discriminative power of each channel. Subsequently, the learned weights can modulate the distance between the query feature and the support features. Extensive experiments on several benchmarks show that our method can effectively improve the performance of existing metric-based FSC methods.

Limitations: The FSC methods mentioned in this paper, including our QCDM, all follow the "N-way K-shot" setting. In other words, current FSC methods assume that the number of labeled data in each category is balanced and consistent across all the tasks, increasing the restrictions on data collection and model training. For example, our QCDM may need to be trained from scratch if the number of labeled samples in test tasks is different from that in training tasks. Therefore, a more realistic scenario is to perform few-shot classification on tasks with different shots. The intricate problem of "N-way any-shot" few-shot classification will be investigated in our future works.

CRediT authorship contribution statement

Wenxiao Wu: Conceptualization, Data curation, Formal analysis, Investigation, Methodology, Software, Validation, Visualization, Writing – original draft, Writing – review & editing. **Yuanjie Shao:** Supervision, Validation, Writing – original draft, Writing – review & editing. **Changxin Gao:** Resources, Supervision, Writing – original draft. **Jing-Hao Xue:** Methodology, Supervision, Validation, Writing – original draft, Writing – review & editing. **Nong Sang:** Funding acquisition, Methodology, Project administration, Resources, Supervision, Validation, Writing – original draft, Writing – review & editing.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

Official source code is available in https://github.com/Wu-Wenxiao/QCDM.

Acknowledgments

This work was partly supported by the National Natural Science Foundation of China (No. U22B2053, No. 61901184) and the Knowledge Innovation Program of Wuhan-Shuguang, China (No. 2023010201 020226).

References

[1] W.-Y. Chen, Y.-C. Liu, Z. Kira, Y.-C.F. Wang, J.-B. Huang, A closer look at few-shot classification, in: International Conference on Learning Representations, 2019.

[2] W. Li, Z. Wang, X. Yang, C. Dong, P. Tian, T. Qin, J. Huo, Y. Shi, L. Wang, Y. Gao, et al., Libfewshot: A comprehensive library for few-shot learning, IEEE Trans. Pattern Anal. Mach. Intell. (2023).

[3] X. Li, X. Yang, Z. Ma, J.-H. Xue, Deep metric learning for few-shot image classification: A review of recent developments, Pattern Recognit. (2023) 109381.

[4] J. Snell, K. Swersky, R. Zemel, Prototypical networks for few-shot learning, in: Neural Information Processing Systems, Vol. 30, 2017.

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

[6] C. Zhang, Y. Cai, G. Lin, C. Shen, Deepemd: Few-shot image classification with differentiable earth mover's distance and structured classifiers, in: IEEE Conference on Computer Vision and Pattern Recognition, 2020, pp. 12203–12213.

[7] J. Xie, F. Long, J. Lv, Q. Wang, P. Li, Joint distribution matters: Deep brownian distance covariance for few-shot classification, in: IEEE Conference on Computer Vision and Pattern Recognition, 2022, pp. 7972–7981.

[8] H. Huang, Z. Wu, W. Li, J. Huo, Y. Gao, Local descriptor-based multi-prototype network for few-shot learning, Pattern Recognit. 116 (2021) 107935.

[9] W. Chen, Z. Zhang, W. Wang, L. Wang, Z. Wang, T. Tan, Few-shot learning with unsupervised part discovery and part-aligned similarity, Pattern Recognit. 133 (2023) 108986.

[10] B. Zhou, A. Khosla, A. Lapedriza, A. Oliva, A. Torralba, Object detectors emerge in deep scene cnns, in: International Conference on Learning Representations, 2015.

[11] D. Bau, B. Zhou, A. Khosla, A. Oliva, A. Torralba, Network dissection: Quantifying interpretability of deep visual representations, in: IEEE Conference on Computer Vision and Pattern Recognition, 2017, pp. 6541–6549.

[12] X. Luo, J. Xu, Z. Xu, Channel importance matters in few-shot image classification, in: International Conference on Machine Learning, PMLR, 2022, pp. 14542–14559.

[13] O. Vinyals, C. Blundell, T. Lillicrap, D. Wierstra, et al., Matching networks for one shot learning, in: Neural Information Processing Systems, Vol. 29, 2016.

[14] M.N. Rizve, S. Khan, F.S. Khan, M. Shah, Exploring complementary strengths of invariant and equivariant representations for few-shot learning, in: IEEE Conference on Computer Vision and Pattern Recognition, 2021, pp. 10836–10846.

[15] D. Wertheimer, B. Hariharan, Few-shot learning with localization in realistic settings, in: IEEE Conference on Computer Vision and Pattern Recognition, 2019, pp. 6558–6567.

[16] J. Liu, L. Song, Y. Qin, Prototype rectification for few-shot learning, in: Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part I 16, Springer, 2020, pp. 741–756.

[17] W. Li, L. Wang, J. Huo, Y. Shi, Y. Gao, J. Luo, Asymmetric distribution measure for few-shot learning, in: International Joint Conference on Artificial Intelligence, 2021, pp. 2957–2963.

[18] V.N. Nguyen, S. Løkse, K. Wickstrøm, M. Kampffmeyer, D. Roverso, R. Jenssen, Sen: A novel feature normalization dissimilarity measure for prototypical few-shot learning networks, in: Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XXIII 16, Springer, 2020, pp. 118–134.

[19] J. Hu, L. Shen, G. Sun, Squeeze-and-excitation networks, in: IEEE Conference on Computer Vision and Pattern Recognition, 2018, pp. 7132–7141.

[20] J. Lai, S. Yang, W. Liu, Y. Zeng, Z. Huang, W. Wu, J. Liu, B.-B. Gao, C. Wang, tSF: Transformer-based semantic filter for few-shot learning, in: European Conference on Computer Vision, Springer, 2022, pp. 1–19.

[21] J. Lai, S. Yang, W. Wu, T. Wu, G. Jiang, X. Wang, J. Liu, B.-B. Gao, W. Zhang, Y. Xie, et al., SpatialFormer: Semantic and target aware attentions for few-shot learning, in: AAAI Conference on Artificial Intelligence, 2023.

[22] G. Chandrashekar, F. Sahin, A survey on feature selection methods, Comput. Electr. Eng. 40 (1) (2014) 16–28.

[23] B. Venkatesh, J. Anuradha, A review of feature selection and its methods, Cybern. Inf. Technol. 19 (1) (2019) 3–26.

[24] V.G. Satorras, J.B. Estrach, Few-shot learning with graph neural networks, in: International Conference on Learning Representations, 2018.

[25] W. Li, L. Wang, J. Xu, J. Huo, Y. Gao, J. Luo, Revisiting local descriptor based image-to-class measure for few-shot learning, in: IEEE Conference on Computer Vision and Pattern Recognition, 2019, pp. 7260–7268.

[26] D. Wertheimer, L. Tang, B. Hariharan, Few-shot classification with feature map reconstruction networks, in: IEEE Conference on Computer Vision and Pattern Recognition, 2021, pp. 8012–8021.

[27] F. Zhou, P. Wang, L. Zhang, W. Wei, Y. Zhang, Revisiting prototypical network for cross domain few-shot learning, in: IEEE Conference on Computer Vision and Pattern Recognition, 2023, pp. 20061–20070.

[28] R. Hou, H. Chang, B. Ma, S. Shan, X. Chen, Cross attention network for few-shot classification, in: Neural Information Processing Systems, Vol. 32, 2019.

[29] C. Xu, Y. Fu, C. Liu, C. Wang, J. Li, F. Huang, L. Zhang, X. Xue, Learning dynamic alignment via meta-filter for few-shot learning, in: IEEE Conference on Computer Vision and Pattern Recognition, 2021, pp. 5182–5191.

[30] X. Huang, S.H. Choi, Sapenet: self-attention based prototype enhancement network for few-shot learning, Pattern Recognit. 135 (2023) 109170.

[31] Z. Li, Z. Hu, W. Luo, X. Hu, SaberNet: Self-attention based effective relation network for few-shot learning, Pattern Recognit. 133 (2023) 109024.

[32] S. Lee, W. Moon, J.-P. Heo, Task discrepancy maximization for fine-grained few-shot classification, in: IEEE Conference on Computer Vision and Pattern Recognition, 2022, pp. 5331–5340.

[33] Y. Tian, Y. Wang, D. Krishnan, J.B. Tenenbaum, P. Isola, Rethinking few-shot image classification: a good embedding is all you need? in: Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XIV 16, Springer, 2020, pp. 266–282.

[34] B. Liu, Y. Cao, Y. Lin, Q. Li, Z. Zhang, M. Long, H. Hu, Negative margin matters: Understanding margin in few-shot classification, in: Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part IV 16, Springer, 2020, pp. 438–455.

[35] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, L. Fei-Fei, Imagenet: A large-scale hierarchical image database, in: IEEE Conference on Computer Vision and Pattern Recognition, 2009, pp. 248–255.

[36] S. Ravi, H. Larochelle, Optimization as a model for few-shot learning, in: International Conference on Learning Representations, 2017.

[37] M. Ren, E. Triantafillou, S. Ravi, J. Snell, K. Swersky, J.B. Tenenbaum, H. Larochelle, R.S. Zemel, Meta-learning for semi-supervised few-shot classification, in: International Conference on Learning Representations, 2018.

[38] C. Wah, S. Branson, P. Welinder, P. Perona, S. Belongie, The caltech-ucsd birds-200–2011 dataset, Calif. Inst. Technol. (2011).
[39] H.-J. Ye, H. Hu, D.-C. Zhan, F. Sha, Few-shot learning via embedding adaptation with set-to-set functions, in: IEEE Conference on Computer Vision and Pattern Recognition, 2020, pp. 8808–8817.
[40] S. Baik, M. Choi, J. Choi, H. Kim, K.M. Lee, Meta-learning with adaptive hyperparameters, in: Neural Information Processing Systems, Vol. 33, 2020, pp. 20755–20765.
[41] H.-Y. Tseng, H.-Y. Lee, J.-B. Huang, M.-H. Yang, Cross-domain few-shot classification via learned feature-wise transformation, in: International Conference on Learning Representations, 2020.
[42] D. Kang, H. Kwon, J. Min, M. Cho, Relational embedding for few-shot classification, in: IEEE International Conference on Computer Vision, 2021, pp. 8822–8833.
[43] S. Baik, J. Choi, H. Kim, D. Cho, J. Min, K.M. Lee, Meta-learning with task-adaptive loss function for few-shot learning, in: IEEE Conference on Computer Vision and Pattern Recognition, 2021, pp. 9465–9474.
[44] Y. Shao, W. Wu, X. You, C. Gao, N. Sang, Improving the generalization of MAML in few-shot classification via bi-level constraint, IEEE Trans. Circuits Syst. Video Technol. (2022).
[45] A. Afrasiyabi, H. Larochelle, J.-F. Lalonde, C. Gagné, Matching feature sets for few-shot image classification, in: IEEE Conference on Computer Vision and Pattern Recognition, 2022, pp. 9014–9024.
[46] Y. Liu, W. Zhang, C. Xiang, T. Zheng, D. Cai, X. He, Learning to affiliate: Mutual centralized learning for few-shot classification, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2022, pp. 14411–14420.
[47] Y. Liu, T. Zheng, J. Song, D. Cai, X. He, Dmn4: Few-shot learning via discriminative mutual nearest neighbor neural network, in: Proceedings of the AAAI Conference on Artificial Intelligence, Vol. 36, 2022, pp. 1828–1836.
[48] B. Shi, W. Li, J. Huo, P. Zhu, L. Wang, Y. Gao, Global-and local-aware feature augmentation with semantic orthogonality for few-shot image classification, Pattern Recognit. 142 (2023) 109702.
[49] D. dan Guo, L. Tian, H. Zhao, M. Zhou, H. Zha, Adaptive distribution calibration for few-shot learning with hierarchical optimal transport, in: Neural Information Processing Systems, 2022.
[50] B. Zhang, X. Li, S. Feng, Y. Ye, R. Ye, MetaNODE: Prototype optimization as a neural ODE for few-shot learning, in: AAAI Conference on Artificial Intelligence, 2022.
[51] J. Cheng, F. Hao, L. Liu, D. Tao, Imposing semantic consistency of local descriptors for few-shot learning, IEEE Trans. Image Process. 31 (2022) 1587–1600.
[52] H. Tang, C. Yuan, Z. Li, J. Tang, Learning attention-guided pyramidal features for few-shot fine-grained recognition, Pattern Recognit. 130 (2022) 108792.
[53] Q. Guo, H. Gong, X. Wei, Y. Fu, W. Ge, Y. Yu, W. Zhang, RankDNN: Learning to rank for few-shot learning, 2022, arXiv preprint arXiv:2211.15320.
[54] B. Munjal, A. Flaborea, S. Amin, F. Tombari, F. Galasso, Query-guided networks for few-shot fine-grained classification and person search, Pattern Recognit. 133 (2023) 109049.
[55] Q. Liu, W. Cao, Z. He, Cycle optimization metric learning for few-shot classification, Pattern Recognit. 139 (2023) 109468.
[56] J. Krause, M. Stark, J. Deng, L. Fei-Fei, 3D object representations for fine-grained categorization, in: IEEE International Conference on Computer Vision, 2013, pp. 554–561.
[57] J. Oh, H. Yoo, C. Kim, S.-Y. Yun, BOIL: Towards representation change for few-shot learning, in: International Conference on Learning Representations, 2021.
[58] R. Das, Y.-X. Wang, J.M. Moura, On the importance of distractors for few-shot classification, in: IEEE International Conference on Computer Vision, 2021, pp. 9030–9040.
[59] H. Liang, Q. Zhang, P. Dai, J. Lu, Boosting the generalization capability in cross-domain few-shot learning via noise-enhanced supervised autoencoder, in: IEEE International Conference on Computer Vision, 2021, pp. 9424–9434.
[60] H. Wang, Z.-H. Deng, Cross-domain few-shot classification via adversarial task augmentation, in: International Joint Conference on Artificial Intelligence, 2021.
[61] R. Xu, L. Xing, B. Liu, D. Tao, W. Cao, W. Liu, Cross-domain few-shot classification via class-shared and class-specific dictionaries, Pattern Recognit. 144 (2023) 109811.
[62] S. Woo, J. Park, J.-Y. Lee, I.S. Kweon, Cbam: Convolutional block attention module, in: European Conference on Computer Vision, 2018, pp. 3–19.
[63] A. Chattopadhay, A. Sarkar, P. Howlader, V.N. Balasubramanian, Grad-cam++: Generalized gradient-based visual explanations for deep convolutional networks, in: 2018 IEEE Winter Conference on Applications of Computer Vision, WACV, IEEE, 2018, pp. 839–847.

Wenxiao Wu received the B.E. degree in School of Artificial Intelligence and Automation, Huazhong University of Science and Technology, Wuhan, China, in 2021. He is currently pursuing the M.E. degree with the School of Artificial Intelligence and Automation, Huazhong University of Science and Technology, Wuhan, China. His research interests include deep learning and computer vision.

Yuanjie Shao received the B.S. and M.S degree in college of mechanical and electronic information, China University of Geosciences in 2010 and 2013, Wuhan, China, and Ph.D. degree in Control science and Engineering from Huazhong University of Science and Technology in 2018. He is currently a lecturer with the School of Electronic Information and Communications, Huazhong University of Science and Technology, Wuhan, China. His research interests include pattern recognition, computer vision.

Changxin Gao received the Ph.D. degree in pattern recognition and intelligent systems from Huazhong University of Science and Technology in 2010. He is currently a Professor with the School of Artificial Intelligence and Automation, Huazhong University of Science and Technology. His research interests are pattern recognition and surveillance video analysis.

Jing-Hao Xue received the Dr.Eng. degree in signal and information processing from Tsinghua University in 1998 and the Ph.D. degree in statistics from the University of Glasgow in 2008. He is a Professor in the Department of Statistical Science, University College London. His research interests include statistical classification, high-dimensional data analysis, pattern recognition and image processing. He is an Associate Editor of IEEE Transactions on Circuits and Systems for Video Technology, IEEE Transactions on Cybernetics, and IEEE Transactions on Neural Networks and Learning Systems.

Nong Sang received the B.E. degree in computer science and engineering, the M.S. degree in pattern recognition and intelligent control, and the Ph.D. degree in pattern recognition and intelligent systems from Huazhong University of Science and Technology in 1990, 1993, and 2000, respectively. He is currently a Professor with the School of Artificial Intelligence and Automation, Huazhong University of Science and Technology. His research interests include object detection, object tracking, image/video segmentation, and analysis of surveillance videos.