ReNAP: Relation Network with Adaptive Prototypical Learning for Few-Shot Classification

Xiaoxu Li[a], Yalan Li[a], Yixiao Zheng[c], Rui Zhu[d], Zhanyu Ma[c,*], Jing-Hao Xue[b] and Jie Cao[a]

[a]School of Computer and Communication, Lanzhou University of Technology, China.
[b]Department of Statistical Science, University College London, U.K.
[c]Pattern Recognition and Intelligent System Laboratory, School of Artificial Intelligence, Beijing University of Posts and Telecommunications, China.
[d]Faculty of Actuarial Science and Insurance, City, University of London, U.K.

ARTICLE INFO

Keywords:
Few-shot learning
Relation network
Prototypical learning
Convolutional neural networks

ABSTRACT

Traditional deep learning-based image classification methods often fail to recognize a new class that does not exist in the training dataset, particularly when the new class only has a small number of samples. Such a challenging and new learning problem is referred to as few-shot learning. In few-shot learning, the relation network (RelationNet) is a powerful method. However, in RelationNet and its state-of-the-art variants, the prototype of each class is obtained by a simple summation or average over the labeled samples. These simple sample statistics cannot accurately capture the distinct characteristics of the diverse classes of real-world images. To address this problem, in this paper, we propose the *Relation Network with Adaptive Prototypical Learning* method (ReNAP), which can learn the class prototypes adaptively and provide more accurate representations of the classes. More specifically, ReNAP embeds an adaptive prototypical learning module constructed by a convolutional network into RelationNet. Our ReNAP achieves superior classification performances to RelationNet and other state-of-the-art methods on four widely used benchmark datasets, FC100, CUB-200-2011, Stanford-Cars, and Stanford-Dogs.

1. Introduction

Deep learning has been developed rapidly in recent years for image classification and recognition [17, 18, 4, 24]. To train these deep learning models, however, we need large amounts of labeled data [30, 20]. Due to the incompetence of conventional deep learning methods on classification with few labeled examples per class, few-shot learning emerges [7, 12, 26, 41, 23].

In few-shot classification, there have been several ways to alleviate the deficiency of samples [20, 22]. For example, meta-learning-based methods [21, 19, 8, 37, 27, 26] aim to learn how to learn, such as learning an initialization. Metric-based methods [34, 25, 31, 33, 19] learn the similarity metric of samples to conduct few-shot classification. Data augmentation-based methods [2] focus on augmenting training samples based on few labeled samples. Transfer learning-based methods [5] aim to transfer knowledge from similar learning tasks and fine-tune the pre-trained models to enhance few-shot learning.

Among these methods, metric-based methods have attracted extensive attention, since they can learn metrics that adapt to data automatically and some of them achieve state-of-the-art classification performances, such as DN4 [19] and EGNN [14]. There are two types of metric-based methods, depending on whether the metrics are learned from data.

The first type is to learn a feature embedding with a predefined metric. For example, the matching networks [34] use the attention mechanism for feature embedding and the cosine distance for measuring the similarities between samples. The prototypical networks [31] assume that a class can be represented by a single feature embedding, i.e. prototype, which is defined as the averaged features of support samples; the Euclidean distance is adopted as the metric to measure the similarities between a test sample and class prototypes. Built on the prototype network, the infinite mixture prototype [1] introduces multiple class prototypes. Wu et al. [40] propose an effective non-parametric classifier termed attentive prototypes to replace the simple prototypes, where the instances are weighted by their reconstruction errors

*Corresponding author
mazhanyu@bupt.edu.cn (Z. Ma)
ORCID(s):

for a given query. Instead of using the feature maps-based measures, DN4 [19] adopts local descriptors and the cosine distance to measure the image-to-class distances.

The second type is to learn a metric and feature embedding from data simultaneously. In few-shot learning, the relation network (RelationNet) [33] is the first to learn a metric to measure the similarities between images by a convolutional neural network module: the input of this module is the concatenation of two images and the output is their relation score. RelationNet also represents each class by a prototype defined as the sum of the embedded features of support samples. Similarly to RelationNet, the graph neural networks (GNN) based few-shot learning methods [14] also use neural network modules to measure the similarities between images. However, one important difference between them is that RelationNet only uses similarity information between support images and query images in a task, while GNN-based methods employ similarity information of any two images in the task. Position-aware relational network (PARN) [42] solves the problem that the semantically related-objects in two images are not properly extracted when calculating similarity. Self-attention relation network (SARN) [11] involves an attention module to model the non-local features of an image.

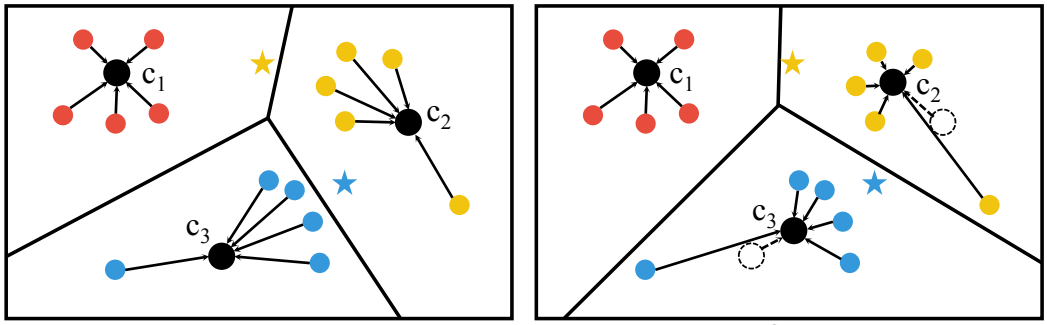

Figure 1: (Left) RelationNet: The prototype of a class is obtained by summing the embedded features of the support samples in the class. (Right) The proposed method: The prototype of a class is learned from a convolutional network module with the embedded features of the support samples in the class as input. The prototype is automatically adjusted from the dashed circle to the solid dot by the convolutional network module. On both left-hand and right-hand panels, black dots represent prototypes; other-colored dots are support samples; stars indicate query samples; and different colors are for different classes.

Besides its successful application in few-shot learning, prototypical learning has also been involved in other tasks. Zhu and Yang [44] used label independent memory (LIM) to adaptively generate an aggregated embedding prototype for each class to solve semi-supervised few-shot video classification problem. Xiao et al. [43] proposed an adaptive mixture mechanism to integrate the existing label information into support features of each class to get more interactive class prototypes to solve the few-shot relation classification task. Wang et al. [38] proposed an end-to-end interactive prototype learning framework to learn better active object representations by leveraging the motion cues from the actor for egocentric action recognition task. These methods all use delicate and refined design to more accurately represent the prototype of each category. Ding et al. [6] used the instance-wise prototype as a non-parametric classifier for unsupervised domain adaptation in person re-identification.

We note that the single prototype representation of one class in previous few-shot image classification approaches, e.g. the prototypical networks and RelationNet, is calculated by averaging or summing over the embedded features of the supporting samples. However, these simple summary statistics cannot well capture the distinct and sophisticated properties of real-world image classes.

Therefore, to provide more accurate representations of the classes, we propose the *Relation Network with Adaptive Prototype Learning* method (ReNAP), which designs a convolutional network block that can learn the prototype of a class adaptively As shown in Figure 1, when the support samples contain outliers, the prototype calculated by simple averaging or summing over all samples will be biased towards the outliers, which cannot accurately represent the potential class prototypes, resulting in misclassified query samples. With adaptive prototypical learning, prototypes are dynamically adjusted during the training phase so that the learned prototypes can better represent real class prototypes for more accurate classification. To be more specific, we embed into RelationNet an additional convolutional module to learn the single class prototypes. The prototypes learned from the convolutional module are expected to capture

more accurate characteristics of the classes than simple summary statistics, although it involves more parameters and requires additional computation. Experimental results show that, on the FC100, CUB-200-2011, Stanford-Cars, and Stanford-Dogs datasets, the proposed ReNAP method achieves better classification performances than RelationNet and other state-of-the-art methods for few-shot image classification.

In summary, our contributions are twofold:

1. A novel neural network for few-shot classification, *Relation Network with Adaptive Prototype Learning* method (ReNAP), is proposed. It allows prototypical representations of classes to be learned adaptively. Existing methods to calculate class prototypes by averaging or summing operations can be seen as special cases of the proposed ReNAP. At the same time, it can alleviate the problem of inaccurate prototype representation caused by outliers in simple summary statistics.
2. Experimental results on four commonly used small-sample image datasets demonstrate the superior classification performance of the proposed ReNAP. Visualizations of features and relation scores also indicate that, compared with RelationNet, ReNAP can learn more precise prototypical representations of classes.

2. The Proposed ReNAP Method

In few-shot classification, the training stage involves many tasks. Each task is constructed by randomly selecting C classes from the base data, with K samples from each class (totally $C \times K$ samples) to form the support set, and Q randomly selected samples from the C classes to form the query set. The model is continuously trained on many tasks sampled from the base data, so that it can learn how to distinguish the C classes based on the $C \times K$ support samples on the novel data. Note that there is no overlapping between the base and novel datasets. Such few-shot classification is also called C-way K-shot problem.

Before introducing our proposed method, we first review two related state-of-the-art few-shot learning methods, the prototypical networks [31] and relation network [33].

2.1. Prototypical Networks

The prototypical networks are the first method to introduce the "prototype" concept to few-shot learning. The prototypical networks can recognize new classes that have never been seen in the training stage, and only need a few samples for each class to train the model. We illustrate the prototypical networks in Figure 2 for a 3-way 3-shot classification. We can observe that the prototypical networks map the samples in each class to a feature embedding space, and then calculate the mean of the embedded features as the prototype to represent the class. To obtain the label of the query image, a classifier is used to calculate the Euclidean distance between the query image and the prototype of each class.

Let $\mathbf{p}_k$ denote the prototype representation of the kth class ($k = 1, 2, \ldots, C$), h_ϕ denote the feature embedding function, S_k denote the support dataset in the kth class and $\mathbf{x}_{ik}$ denote the ith original image in S_k. The prototype $\mathbf{p}_k^{\mathrm{PN}}$ is calculated as the average of all embedded features in the support set:

$$\mathbf{p}_k^{\mathrm{PN}} = \frac{1}{|S_k|} \sum_{\mathbf{x}_{ik} \in S_k} h_\phi(\mathbf{x}_{ik}), \tag{1}$$

where $|S_k|$ is the number of samples in the support set S_k.

The Euclidean distance between the jth query image $\mathbf{x}_j$ and the kth prototype is then calculated as

$$d_{jk} = \sqrt{(h_\phi(\mathbf{x}_j) - \mathbf{p}_k^{\mathrm{PN}})^T (h_\phi(\mathbf{x}_j) - \mathbf{p}_k^{\mathrm{PN}})}. \tag{2}$$

2.2. Relation Network

Prior to the Relational Network (RelationNet) [33], the methods in few-shot learning usually use pre-determined distance metrics, such as the Euclidean distance or cosine distance, to measure the image-to-image or image-to-class similarities. However, the pre-determined distance metrics cannot well capture the intrinsic structures of different data. The Relational Network is the first method to learn the feature embedding and non-linear similarity metric simultaneously. The learned metric can automatically adapt to different data and capture their distinct intrinsic structures.

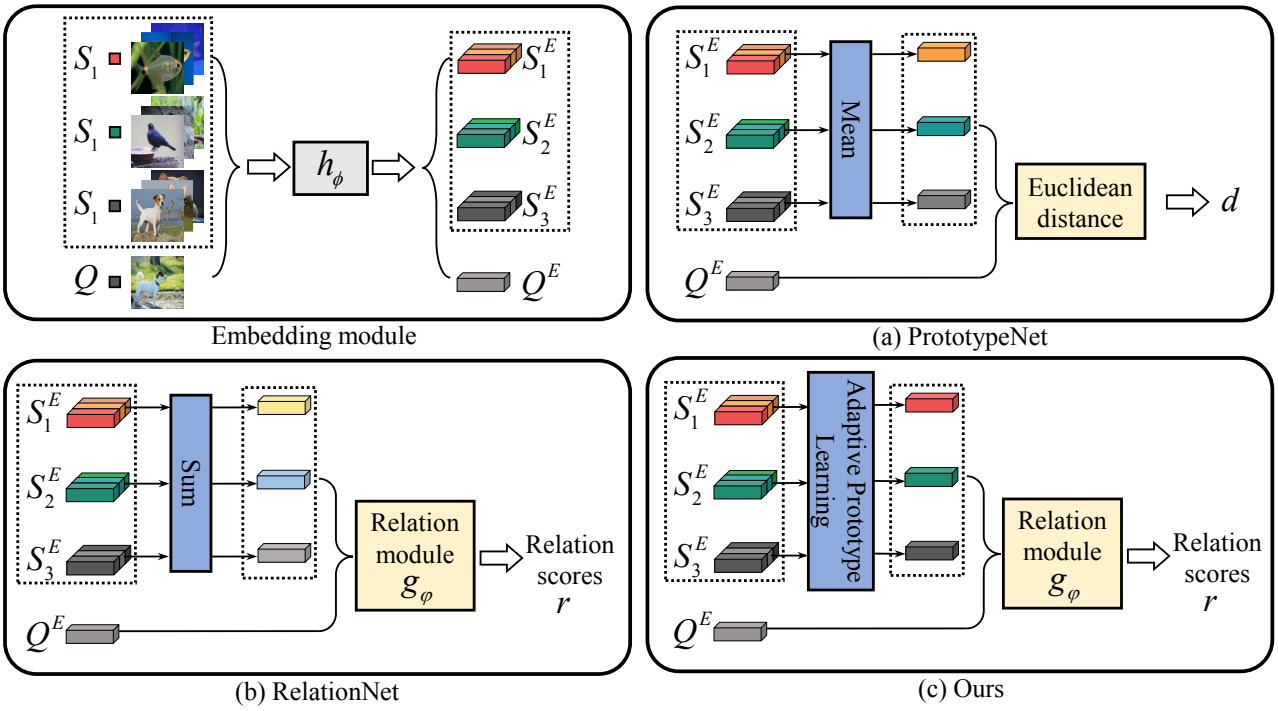

Figure 2: Illustrations of Prototypical Networks, Relation Network, and the proposed ReNAP method. On the top-left panel, the embedding module is the common part of the three methods, where S_k $(k = 1, 2, 3)$ denotes the kth support set while S_k^E $(k = 1, 2, 3)$ denotes the embedded kth support set via feature embedding function h_ϕ; and Q is the query image while Q^E denotes the embedded query image via h_ϕ. (a), (b) and (c) illustrate the structures of Prototypical Networks, Relation Network, and the proposed method (Ours), respectively, excluding the common embedding and classification modules.

As shown in Figure 2, RelationNet consists of two modules: a feature embedding module h_ϕ and a relation module g_φ. The embedding module is used to extract features, while the relation module is used to learn the distance metric and calculate the similarities or relation scores between the query image and the class prototypes.

The prototype of a class is calculated by summing over the embedded features of support samples in the class as

$$\mathbf{p}_k^{\mathrm{RN}} = \sum_{\mathbf{x}_{ik} \in S_k} h_\phi(\mathbf{x}_{ik}). \tag{3}$$

The relation score, r_{jk}^{RN}, between the jth query image $\mathbf{x}_j$ and the kth class prototype in RelationNet is calculated by

$$r_{jk}^{\mathrm{RN}} = g_\varphi(\mathbf{p}_k^{\mathrm{RN}}, h_\phi(\mathbf{x}_j)). \tag{4}$$

2.3. Relation Network with Adaptive Prototypical Learning

From previous discussion, we can see that the prototypical networks and RelationNet both use simple summary statistics (average or sum) to construct the prototype of a class, which cannot accurately capture the intrinsic characteristics of the classes. Simple summary statistics use the same weights for all support samples to compute prototypes. However, this is not always suitable for real-world data. For example, when outliers present, they should receive less weights to make the class prototype a good representation of the majority of the class; otherwise, the prototype will be biased towards the outliers. Here we propose the *Relation Network with Adaptive Prototypical Learning* method (ReNAP), which can provide more accurate representations of the classes by utilizing an adaptive prototypical learning module. As illustrated in Figure 2, the proposed network contains three modules, the embedding module h_ϕ, the relation module g_φ, and the adaptive prototypical learning module f_ψ.

Instead of simply summing over the embedded features as that in RelationNet, the prototypes are now learned by the adaptive prototypical learning module which is constructed by a convolutional neural network as

$$\mathbf{p}_k^{\mathrm{APL}} = f_\psi(h_\phi(\mathbf{x}_{1k}), h_\phi(\mathbf{x}_{2k}), \ldots, h_\phi(\mathbf{x}_{|S_k|k})), \tag{5}$$

where f_ψ adaptively learns class prototypes using support features from the same class. When in some cases, f_ψ can degrade into a sum or average operation. The relation score, r_{jk}, between the jth query image and the kth class prototype in the proposed method is calculated by

$$r_{jk} = g_\varphi(\mathbf{p}_k^{\mathrm{APL}}, h_\phi(\mathbf{x}_j)). \tag{6}$$

3. Experimental Results and Discussions

The experiments in this section are designed to mainly answer the following five questions, in sections 3.3, 3.4, 3.5, 3.6, and 3.7, respectively:

- How does the classification performance of the proposed method compare with state-of-the-art methods for few-shot image classification?
- Is the improvement provided by the proposed method statistically significant?
- How does the proposed adaptive prototypical learning module affect the proposed method?
- Is the proposed adaptive prototypical learning module applicable to other backbone networks?
- Does the proposed method learn a more accurate prototypical representation for each class?

3.1. Datasets

We select four widely adopted few-shot classification benchmark datasets: FC100 [3], CUB-200-2011 [39], Stanford-Cars [15] and Stanford-Dogs [13].

Few-Shot CIFAR (FC100) [3] is reorganized from CIFAR100 [16] dataset, which consists of 100 classes with 600 samples in each class. In this paper, we divide FC100 into base, validation, and novel sets with 60, 16, and 24 classes, separately, following the origin paper [3].

The CUB-200-2011 [36] dataset contains 200 bird species with 11,788 images in total. Following [9], we randomly split the dataset into a base dataset with 100 classes, a validation dataset with 50 classes, and a novel dataset with 50 classes. All images are resized to 84×84.

The Stanford-Cars dataset [15] contains 196 classes of cars, with totally 16,185 images. It is a benchmark for fine-grained classification tasks. We divide the dataset into a base dataset with 98 classes, a validation dataset with 49 classes, and a novel dataset with 49 classes. All images are resized to 84×84.

The Stanford-Dogs dataset [13] contains 120 types of dogs with 20,580 images. The base, validation, and novel datasets contain 60, 30, and 30 classes, respectively. All images are resized to 84×84.

3.2. Implementation Details

To evaluate the classification performance of the proposed method, we compare it with the following six state-of-the-art methods: Baseline [5], MAML [8], Matching Networks [35], Prototypical Networks [32] and Relation Network [33]. For all state-of-the-art methods, we use the code published by [5], which is also the base of the code of the proposed method.

All methods use the same embedding module, which includes four convolutional blocks. Each convolutional block of *Conv4* (VGG style) has 3×3 convolution of 64 filters, followed by batch normalization and a *ReLU* activation function. The first two blocks contain a 2×2 max-pooling layer while the latter two blocks do not have max-pooling layers.

The matching networks are composed of the embedding module and a matching module which consists of a memory network and an attention module. The memory network uses a bidirectional LSTM and the attention module contains a softmax layer. The similarity metric between images is the cosine distance.

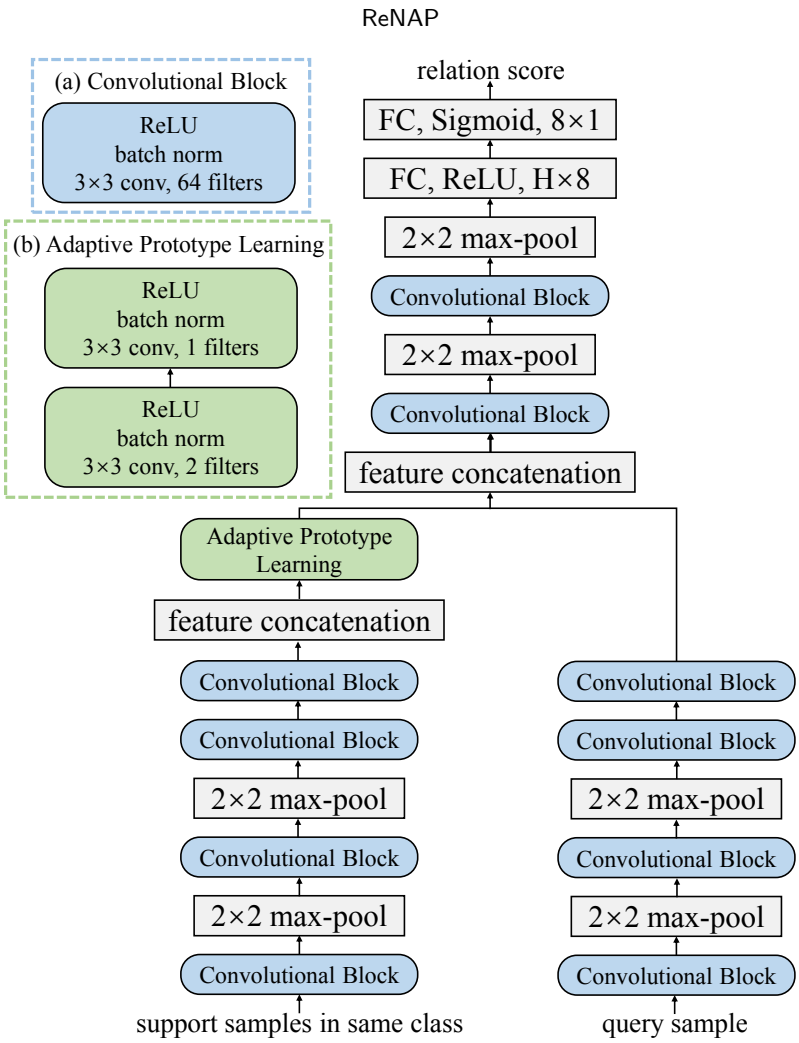

Figure 3: The detailed implementation of the proposed relation network with adaptive prototypical learning. The sub-figure (a) details the convolutional block, and the sub-figure (b) details of the adaptive prototype learning module.

The prototype networks use the negative Euclidean distance to measure the similarity between the flatten image features extracted from the feature embedding module. For multiple shots, e.g. 5-shot, the mean of the embedded support image features of a class is the prototype of the class.

For MAML, we construct a liner classifier on the feature embedding. MAML is a meta-learning algorithm based on optimization and aims to learn the common model parameters which can adapt different tasks quickly. We set 1,600 iterations, and each iteration contains 100 multiple episodes. In the task of each episode, task-specific model parameters are initialized by the common model parameters and are updated by minimizing the loss function of the model based on the samples in the task. For every 100 episodes, the common model parameters are updated once by minimizing the summation of the loss functions of all tasks.

For Baseline, we build a C-class classification network by combining the embedding module and a linear classification layer, where C is the total number of classes in the base data. The network is optimized based on the base data, and then transfer the feature embedding to fine-tune the task-specific network for each task on the test data.

The relation network places a relation module after the embedding module. The relation module consists of two convolutional blocks and two fully connected layers. Each convolutional block has four parts: 3×3 convolution of 64 filters, batch normalization, a *ReLU* non-linearity function and a 2×2 max-pooling.

For the proposed method, the adaptive prototype network f_ψ contains two convolutional layers, with the first layer containing 3×3 convolution of 2-filter and the second layer containing 3×3 convolution of 1-filter. Our relational module is the same as that in RelationNet. The detailed implementation of the proposed method is shown in Figure 3.

Table 1
5-way 5-shot classification accuracies on the FC100, CUB-200-2011, Stanford-Cars, and Stanford-Dogs datasets. The compared methods include Baseline, MAML, Matching Networks (MatchingNet), Prototypical Networks (PrototypeNet), Relation Network (RelationNet), and the proposed ReNAP (Ours). The best results are in **bold** and the second best results are marked with underline.

Dataset	*Methods*					
	Baseline	MAML	MatchingNet	PrototypeNet	RelationNet	Ours
FC100	68.66±0.79	70.29±0.77	68.85±0.77	66.49±0.89	69.61±0.77	**70.45±0.82**
CUB-200-2011	69.14±0.63	76.02±0.67	74.62±0.69	76.09±0.65	77.96±0.67	**78.48±0.65**
Stanford-Cars	53.25±0.66	64.92±0.73	64.56±0.75	62.05±0.74	67.24±0.79	**68.52±0.79**
Stanford-Dogs	52.44±0.65	61.95±0.80	61.25±0.71	59.88±0.70	65.53±0.71	**66.86±0.70**

Table 2
The p-values of the paired t-test based on the 5-way 5-shot classification accuracies on the FC100, CUB-200-2011, Stanford-Cars, and Stanford-Dogs datasets. "$\surd$" denotes the p-values <0.05 and "×" denotes the p-values >0.05. That is, $\surd$ indicates that the proposed method (Ours) is significantly different from the compared method at the significance level of 5%, in terms of mean classification accuracy.

Dataset	** vs. Ours*				
	Baseline	MAML	MatchingNet	PrototypeNet	RelationNet
FC100	$\surd$	$\surd$	$\surd$	$\surd$	$\surd$
CUB-200-2011	$\surd$	$\surd$	$\surd$	$\surd$	$\surd$
Stanford-Cars	$\surd$	$\surd$	$\surd$	$\surd$	$\surd$
Stanford-Dogs	$\surd$	$\surd$	$\surd$	$\surd$	$\surd$

We use the Adam optimizer with the initial learning rate of 10^{-3}. For the adaptive prototype learning module in the proposed method, the initial parameters of each convolutional layer are sampled from normal distributions with means and standard deviations of ${c_i}^{-1}$ and $\sqrt{2}(c_o k^2)^{-1/2}$, respectively, where c_i and c_o are the number of input channels and the number of output channels of the convolutional layer, respectively, and k is the kernel size.

Existing methods to calculate class prototypes by averaging or summing operations can be seen as special cases of our method. When the convolution kernel size of the two convolution blocks in the adaptive Prototype Learning module is set to 1×1, the parameters of the first and the second convolution block are all fixed to 1 and 0.5, respectively, and batch norm and ReLU are removed at the same time, the adaptive Prototype Learning module is degraded into a sum operation. If the parameters of the first and the second convolution block are all fixed to ${c_i}^{-1}$ and 0.5, respectively, the adaptive Prototype Learning module is degraded into an average operation.

3.3. Classification Results

We conduct 5-way 5-shot classification on the FC100, CUB-200-2011, Stanford-Cars, and Stanford-Dogs datasets. For each method, we report the mean accuracy of 600 episodes on the novel dataset, along with its 95% confidence interval in Table 1.

From Table 1, we can see that the proposed method outperforms all other methods in terms of mean accuracy on all datasets. This result shows that by involving the APL module, we can achieve higher accuracies and learn more suitable prototypes for the tested real datasets compared with RelationNet. More analysis on the effect of the APL module is discussed in section 3.5.

3.4. Student's t-test

In the experimental results, the proposed method obtained higher mean classification accuracies than Baseline, MAML, MatchingNet, PrototypeNet, and RelationNet on the FC100, CUB-200-2011, Stanford-Cars, and Stanford-Dogs datasets. To further show that the superior classification performance of the proposed method is not by chance, we perform paired t-test [10] to formally test whether the classification accuracies are different. The null-hypothesis is that the proposed method and the other methods have the same mean classification accuracy. The p-values from paired t-tests are listed in Table 2. We set the significance level as 0.05. The p-values less than 0.05 are denoted as "$\surd$" and the p-values larger than 0.05 are denoted as "$\times$".

From Table 2, we observe that all p-values are less than 0.05, except for those of comparing RelationNet with the proposed method and comparing MatchingNet with the proposed method on the Ominiglot dataset. This indicates that the classification accuracies obtained by the proposed method are statistically significantly different from the compared methods in most cases.

Query Image										
Ground-truth	Coupe IPL	Toyota	smart	Geo Metro	Audi S5	Coupe IPL	Toyota	smart	Geo Metro	Audi S5
RelationNet	Coupe IPL	Toyota	smart	Geo Metro	Audi S5	Audi S5	Audi S5	Coupe IPL	smart	Dodge
Ours	Coupe IPL	Toyota	smart	Geo Metro	Audi S5	Coupe IPL	Toyota	smart	Geo Metro	Audi S5

Figure 4: Example predictions of RelationNet and ReNAP (Ours) on the test data of the Stanford-Cars dataset. Query images are selected from 5 classes, i.e. Infiniti G Coupe IPL 2012 (Coupe IPL), Toyota 4Runner SUV 2012 (Toyota), smart fortwo Convertible 2012 (smart), Geo Metro Convertible 1993 (Geo Metro), and Audi S5 Coupe 2012 (Audi S5). Wrong predictions are labelled by the red color.

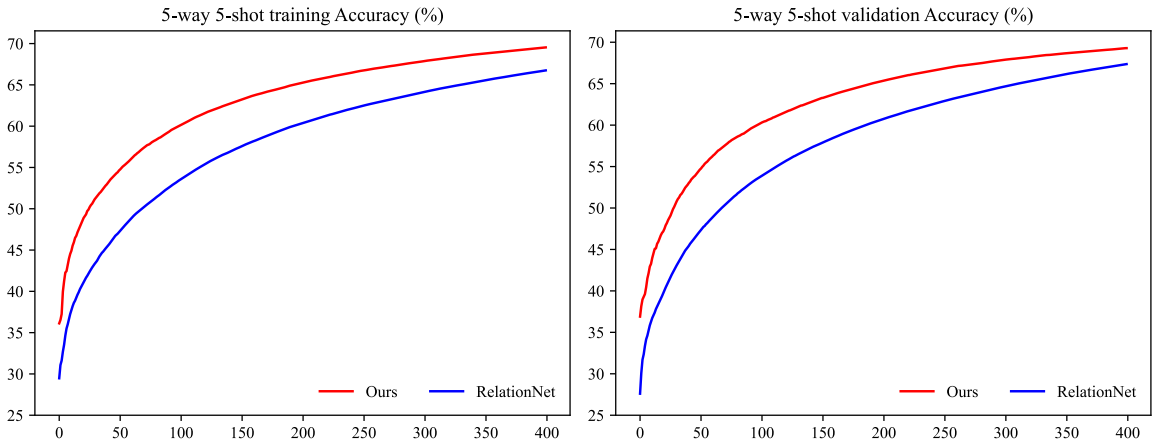

Figure 5: Moving average curve of classification accuracy during training processing on CUB-200-2011 dataset. The figure shows the moving average classification accuracy of each epoch in the 5-way 5-shot classification task, and the backbone adopts Conv4.

3.5. The Effect of Embedding the APL Module in ReNAP

The proposed method can be considered as embedding an adaptive prototypical learning (APL) module into RelationNet. To further verify the effectiveness of the proposed APL module, we compare our ReNAP with RelationNet

Table 3
Few-shot classification accuracies of the Relation Network (RelationNet) and the proposed ReNAP (Ours) on the FC100, CUB-200-2011, Stanford-Cars, and Stanford-Dogs datasets. The best results are in bold.

Datasets	*C-way K-shot*	*Methods*	
		RelationNet	Ours
FC100	5-way 5-shot	69.61±0.77	**69.89±0.78**
	5-way 10-shot	76.30±0.76	**76.49±0.71**
	10-way 5-shot	70.95±0.85	**71.07±0.80**
	10-way 10-shot	76.79±0.70	**77.40±0.75**
CUB-200-2011	5-way 5-shot	77.96±0.67	**78.48±0.65**
	5-way 10-shot	82.18±0.54	**83.09±0.54**
	10-way 5-shot	79.59±0.64	**80.90±0.62**
	10-way 10-shot	82.52±0.55	**82.78±0.55**
Stanford-Cars	5-way 5-shot	67.24±0.79	**68.52±0.79**
	5-way 10-shot	74.81±0.72	**77.96±0.69**
	10-way 5-shot	71.46±0.81	**71.84±0.74**
	10-way 10-shot	77.26±0.63	**78.50±0.67**
Stanford-Dogs	5-way 5-shot	65.53±0.71	**66.86±0.70**
	5-way 10-shot	72.13±0.66	**73.20±0.66**
	10-way 5-shot	66.64±0.73	**69.01±0.73**
	10-way 10-shot	74.37±0.64	**75.64±0.61**

by varying the number of ways and the number of shots on the FC100, CUB-200-2011, Stanford-Cars, and Stanford-Dogs datasets. The setting of C-way K-shot image classification is the same as that is section 3.2. The mean accuracy for each method along with its 95% confidence interval are listed in Table 3.

It is obvious that ReNAP performs better than RelationNet in all classification settings and on all datasets. These results suggest that, by embedding the APL module in ReNAP, we can obtain better classification results in most C-way K-shot problems on the four benchmark datasets, compared with RelationNet.

Moreover, we plot the classification accuracy curves during the training process on the CUB-200-2011 dataset in Figure 5. Clearly, our proposed ReNAP dominants RelationNet during the whole process in terms of training and validation accuracies. Similar patterns can be found on all datasets test in the paper, and thus we only show the result of the CUB-200-2011 dataset as an example. This result further demonstrates the advantage of our ReNAP to get class prototypes: rather than using the simple sum in RelationNet, utilising the convolutional module in ReNAP is more effective to obtain representative prototypes for few-shot classification tasks.

In addition, we compare some example classifications of RelationNet and ReNAP in Figure 4. It is clear that when an obvious sign, e.g. the front view or profile, of a car appears in the image, RelationNet and ReNAP can both predict correctly. However, when the obvious signs of cars have not been presented in the images, ReNAP can predict more precisely than RelationNet. These results also demonstrate that the APL module in ReNAP can help to identify images that are more difficult to classify.

3.6. The Effect of Changing the Backbone Network in ReNAP

In the experiments in Section 3.5, all methods adopt the same feature embedding method described in Section 3.2. In order to further verify that the APL module is also applicable to different backbone networks, we change the style and depth of the backbone network and compare the classification performances of ReNAP and RelationNet. The design of all the backbones follows [5]. The average classification accuracies with their corresponding 95% confidence intervals are listed in Table 4.

Table 4
5-way 5-shot classification accuracies on the FC100, CUB-200-2011, Stanford-Cars, and Stanford-Dogs datasets. The backbones include Conv4 (VGG style), Conv6 (VGG style), ResNet-10, ResNet-18, and ResNet-34.

Backbone	*Datasets*	*Methods*	
		RelationNet	Ours
Conv4	FC100	69.61±0.77	**69.89±0.78**
	CUB-200-2011	77.96±0.67	**78.48±0.65**
	Stanford-Cars	67.24±0.79	**68.52±0.79**
	Stanford-Dogs	65.53±0.71	**66.86±0.70**
Conv6	FC100	71.26±0.79	**71.63±0.78**
	CUB-200-2011	79.93±0.60	**80.07±0.62**
	Stanford-Cars	70.98±0.84	**73.84±0.71**
	Stanford-Dogs	66.28±0.69	**67.72±0.73**
ResNet-10	FC100	74.49±0.80	**76.64±0.82**
	CUB-200-2011	80.94±0.59	**83.67±0.58**
	Stanford-Cars	82.82±0.65	**83.02±0.67**
	Stanford-Dogs	77.20±0.70	**78.04±0.66**
ResNet-18	FC100	76.72±0.81	**77.89±0.78**
	CUB-200-2011	83.01±0.58	**83.03±0.58**
	Stanford-Cars	82.41±0.65	**84.26±0.62**
	Stanford-Dogs	78.96±0.66	**80.48±0.59**
ResNet-34	FC100	**79.34±0.76**	77.28±0.79
	CUB-200-2011	83.72±0.57	**84.15±0.58**
	Stanford-Cars	83.36±0.67	**83.45±0.67**
	Stanford-Dogs	**81.42±0.61**	81.31±0.58

As shown in Table 4, when the backbone network is set as Conv4 (VGG style), Conv6 (VGG style), ResNet-10, ResNet-18 and ResNet-34, the proposed ReNAP method has better classification performance than RelationNet on the CUB-200-2011 and Stanford-Cars datasets. Except for ResNet-34, the proposed method performs better than RelationNet on the FC100 and Stanford-Dogs dataset.

3.7. Visualizations of Features and Relation Scores

To further explore properties of the proposed ReNAP and explain why ReNAP can provide superior classification results, we visualize important feature regions based on the gradient-based technique, *Grad-CAM* [29] in Figure 6. The relation scores based on confusion matrices are shown in Figure 7.

Visualization of Features: We randomly select five query images in the same training task on the *Stanford-Dogs* and *Stanford-Cars* datasets, respectively, and use Grad-Cam[28] to visualize the important regions of these images when the backbone is set as Conv4. As shown in Figure 6, the important regions learned by the proposed ReNAP are more concentrated in the key parts of objects than those learned by RelationNet. That is, the proposed ReNAP method can learn more class-discriminative features than RelationNet.

Visualization of Relation Scores: In Figure 7, on the CUB-200-2011, Stanford-Cars, and Stanford-Dogs datasets, we visualize the predicted relation scores of the proposed ReNAP method and RelationNet in the same training task, respectively. From Figure 7, we can observe that, on the diagonals of the confusion matrices for the CUB-200-2011 dataset, the number of warmer colour bars of ReNAP (Ours) is greater than that of RelationNet, which means that ReNAP can predict more precisely than RelationNet. Similar patterns can also be found on other two datasets. These

Figure 6: Feature visualization under RelationNet and ReNAP (Ours) on the *Stanford-Dogs* and *Stanford-Cars* datasets via the gradient-based technique, *Grad-CAM* [29]. The warmer the region is, the more important the region is.

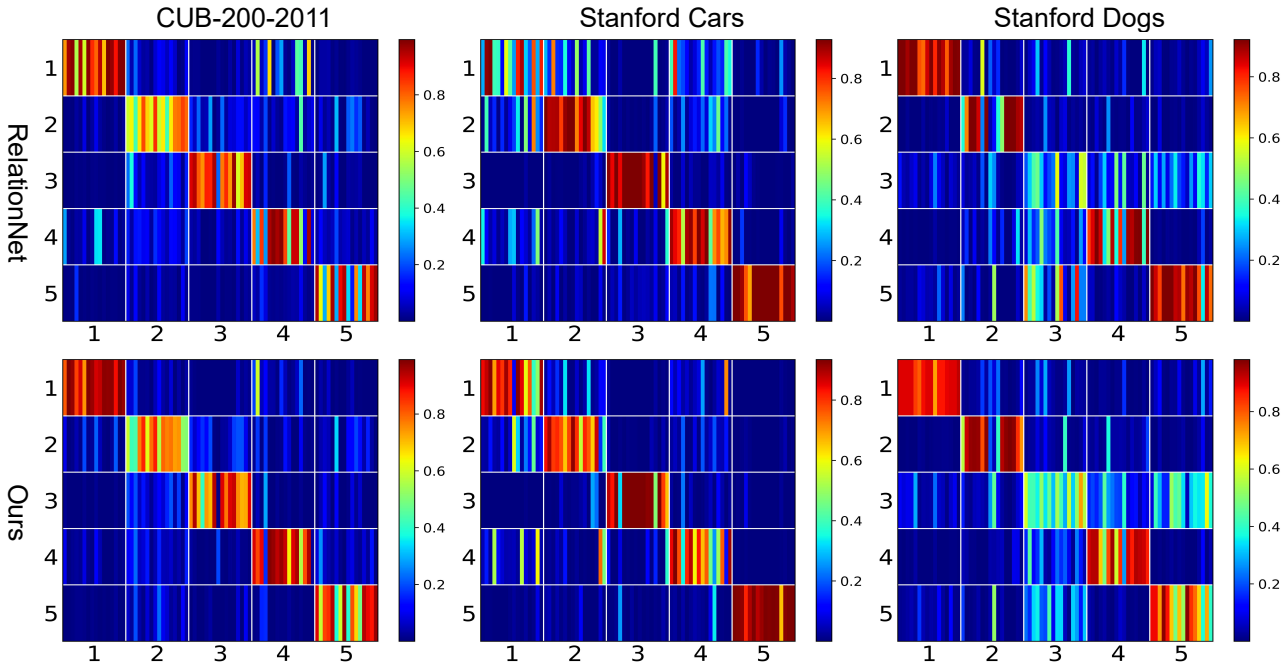

Figure 7: Visualization of relation scores predicted by RelationNet and the proposed ReNAP (Ours) on the *CUB-200-2011*, *Stanford-Cars*, and *Stanford-Dogs* datasets. For each confusion matrix, the vertical axis denotes the five classes in a task while the horizontal axis denotes the query samples in the 5 classes, with each class containing 16 query samples. The color-bars indicate the predicted relation scores.

results suggest that the prototypes learned by ReNAP can represent classes more precisely than RelationNet, which then lead to better relation scores and classification accuracies.

4. Conclusion

In this paper, we proposed a simple yet effective network, the *Relation Network with Adaptive Prototypical Learning* (ReNAP), for few-shot image classification, by embedding an adaptive prototypical learning network module into RelationNet, between the feature embedding module and the relation module. The experimental results on four benchmark datasets demonstrated that: firstly, the proposed network outperforms the compared state-of-the-art methods and the improvement is statistically significant. Secondly, the proposed adaptive prototypical learning network module

is applicable to different backbone networks. Thirdly, the prototype learned by ReNAP can represent each class more precisely than RelationNet. The proposed adaptive prototypical learning network module can also be used in a variety of potential applications, such as person re-identification and domain adaptation, which will be explored in subsequent work.

Acknowledgement

This work is in part by the National Natural Science Foundation of China (NSFC) under Grant 62176110, 62111530146, 61906080, Young Doctoral Fund of Education Department of Gansu Province under Grant 2021QB-038, Hong-liu Distinguished Young Talents Foundation of Lanzhou University of Technology..

References

[1] Allen, K., Shelhamer, E., Shin, H., Tenenbaum, J., 2019. Infinite mixture prototypes for few-shot learning, in: International Conference on Machine Learning, pp. 232–241.
[2] Antoniou, A., Storkey, A., Edwards, H., 2017. Data augmentation generative adversarial networks. arXiv preprint arXiv:1711.04340 .
[3] Bertinetto, L., Henriques, J.F., Torr, P., Vedaldi, A., 2019. Meta-learning with differentiable closed-form solvers, in: International Conference on Learning Representations.
[4] Chang, D., Ding, Y., Xie, J., Bhunia, A.K., Li, X., Ma, Z., Wu, M., Guo, J., Song, Y.Z., 2020. The devil is in the channels: Mutual-channel loss for fine-grained image classification. IEEE Transactions on Image Processing 29, 4683–4695.
[5] Chen, W.Y., Liu, Y.C., Kira, Z., Wang, Y.C.F., Huang, J.B., 2019. A closer look at few-shot classification, in: International Conference on Learning Representations.
[6] Ding, Y., Fan, H., Xu, M., Yang, Y., 2020. Adaptive exploration for unsupervised person re-identification. ACM Transactions on Multimedia Computing, Communications, and Applications (TOMM) 16, 1 – 19.
[7] Edwards, H., Storkey, A., 2016. Towards a neural statistician. arXiv preprint arXiv:1606.02185 .
[8] Finn, C., Abbeel, P., Levine, S., 2017. Model-agnostic meta-learning for fast adaptation of deep networks, in: Proceedings of the 34th International Conference on Machine Learning, pp. 1126–1135.
[9] Hilliard, N., Phillips, L., Howland, S., Yankov, A., Corley, C.D., Hodas, N.O., 2018. Few-shot learning with metric-agnostic conditional embeddings. arXiv preprint arXiv:1802.04376 .
[10] Hsu, H., Lachenbruch, P.A., 2005. Paired t test. Encyclopedia of Biostatistics 6.
[11] Hui, B., Zhu, P., Hu, Q., Wang, Q., 2019. Self-attention relation network for few-shot learning, in: 2019 IEEE international conference on multimedia & expo workshops (ICMEW), IEEE. pp. 198–203.
[12] Kaiser, Ł., Nachum, O., Roy, A., Bengio, S., 2017. Learning to remember rare events. arXiv preprint arXiv:1703.03129 .
[13] Khosla, A., Jayadevaprakash, N., Yao, B., Fei-Fei, L., 2011. Novel dataset for fine-grained image categorization: Stanford Dogs, in: Proc. CVPR Workshop on Fine-Grained Visual Categorization (FGVC).
[14] Kim, J., Kim, T., Kim, S., Yoo, C.D., 2019. Edge-labeling graph neural network for few-shot learning, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 11–20.
[15] Krause, J., Stark, M., Deng, J., Fei-Fei, L., 2013. 3D object representations for fine-grained categorization, in: Proceedings of the IEEE International Conference on Computer Vision Workshops, pp. 554–561.
[16] Krizhevsky, A., 2009. Learning multiple layers of features from tiny images. Technical Report.
[17] Krizhevsky, A., Sutskever, I., Hinton, G.E., 2012. ImageNet classification with deep convolutional neural networks, in: Advances in Neural Information Processing Systems, pp. 1097–1105.
[18] Lake, B.M., Salakhutdinov, R., Gross, J., Tenenbaum, J.B., 2011. One shot learning of simple visual concepts, in: CogSci.
[19] Li, W., Wang, L., Xu, J., Huo, J., Gao, Y., Luo, J., 2019. Revisiting local descriptor based image-to-class measure for few-shot learning, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 7260–7268.
[20] Li, X., Chang, D., Ma, Z., Tan, Z.H., Xue, J.H., Cao, J., Yu, J., Guo, J., 2020a. OSLNet: Deep small-sample classification with an orthogonal softmax layer. IEEE Transactions on Image Processing 29, 6482–6495.
[21] Li, X., Sun, Z., Xue, J.H., Ma, Z., 2020b. A concise review of recent few-shot meta-learning methods. Neurocomputing .
[22] Li, X., Yu, L., Yang, X., Ma, Z., Xue, J.H., Cao, J., Guo, J., 2020c. ReMarNet: Conjoint relation and margin learning for small-sample image classification. IEEE Transactions on Circuits and Systems for Video Technology .
[23] Lu, J., Gong, P., Ye, J., Zhang, C., 2020. Learning from very few samples: A survey. arXiv preprint arXiv:2009.02653 .
[24] Ma, Z., Chang, D., Xie, J., Ding, Y., Wen, S., Li, X., Si, Z., Guo, J., 2019. Fine-grained vehicle classification with channel max pooling modified cnns. IEEE Transactions on Vehicular Technology 68, 3224–3233.
[25] Mai, S., Hu, H., Xu, J., 2019. Attentive matching network for few-shot learning. Computer Vision and Image Understanding 187, 102781.
[26] Ravi, S., Larochelle, H., 2017. Optimization as a model for few-shot learning, in: International Conference on Learning Representations.
[27] Santoro, A., Bartunov, S., Botvinick, M., Wierstra, D., Lillicrap, T., 2016. Meta-learning with memory-augmented neural networks, in: International Conference on Machine Learning, pp. 1842–1850.
[28] Selvaraju, R.R., Cogswell, M., Das, A., Vedantam, R., Batra, D., 2019. Grad-CAM: Visual explanations from deep networks via gradient-based localization. International Journal of Computer Vision 128.
[29] Selvaraju, R.R., Cogswell, M., Das, A., Vedantam, R., Parikh, D., Batra, D., 2017. Grad-CAM: Visual explanations from deep networks via gradient-based localization, in: IEEE International Conference on Computer Vision, pp. 618–626.
[30] Simonyan, K., Zisserman, A., 2014. Very deep convolutional networks for large-scale image recognition. arXiv preprint arXiv:1409.1556 .

[31] Snell, J., Swersky, K., Zemel, R., 2017a. Prototypical networks for few-shot learning, in: Advances in Neural Information Processing Systems, pp. 4077–4087.
[32] Snell, J., Swersky, K., Zemel, R., 2017b. Prototypical networks for few-shot learning, in: Advances in Neural Information Processing Systems, pp. 4077–4087.
[33] Sung, F., Yang, Y., Zhang, L., Xiang, T., Torr, P.H., Hospedales, T.M., 2018. Learning to compare: Relation network for few-shot learning, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR).
[34] Vinyals, O., Blundell, C., Lillicrap, T., Kavukcuoglu, K., Wierstra, D., 2016a. Matching networks for one shot learning, in: Advances in Neural Information Processing Systems, pp. 3630–3638.
[35] Vinyals, O., Blundell, C., Lillicrap, T., Wierstra, D., et al., 2016b. Matching networks for one shot learning, in: Advances in Neural Information Processing Systems, pp. 3630–3638.
[36] Wah, C., Branson, S., Welinder, P., Perona, P., Belongie, S., 2011. The Caltech-UCSD Birds-200-2011 dataset .
[37] Wang, D., Cheng, Y., Yu, M., Guo, X., Zhang, T., 2019. A hybrid approach with optimization-based and metric-based meta-learner for few-shot learning. Neurocomputing 349, 202–211.
[38] Wang, X., Zhu, L., Wang, H., Yang, Y., 2021. Interactive prototype learning for egocentric action recognition, in: Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV), pp. 8148–8157.
[39] Welinder, P., Branson, S., Mita, T., Wah, C., Schroff, F., Belongie, S., Perona, P., 2010. Caltech-UCSD Birds 200 .
[40] Wu, F., Smith, J.S., Lu, W., Pang, C., Zhang, B., 2020a. Attentive prototype few-shot learning with capsule network-based embedding, in: European Conference on Computer Vision, Springer. pp. 237–253.
[41] Wu, X., Sahoo, D., Hoi, S., 2020b. Meta-RCNN: Meta learning for few-shot object detection, in: Proceedings of the 28th ACM International Conference on Multimedia, pp. 1679–1687.
[42] Wu, Z., Li, Y., Guo, L., Jia, K., 2019. PARN: Position-aware relation networks for few-shot learning, in: Proceedings of the IEEE/CVF International Conference on Computer Vision, pp. 6659–6667.
[43] Xiao, Y., Jin, Y., Hao, K., 2021. Adaptive prototypical networks with label words and joint representation learning for few-shot relation classification. IEEE Transactions on Neural Networks and Learning Systems , 1–12.
[44] Zhu, L., Yang, Y., 2022. Label independent memory for semi-supervised few-shot video classification. IEEE Transactions on Pattern Analysis and Machine Intelligence 44, 273–285.