PARN: Position-Aware Relation Networks for Few-Shot Learning

Ziyang Wu[1], Yuwei Li[2], Lihua Guo[3] and Kui Jia[4]

School of Electronic and Information Engineering,
South China University of Technology, Guangzhou, China
{eezywu[1], 201821010824[2]}@mail.scut.edu.cn, {guolihua[3], kuijia[4]}@scut.edu.cn

Abstract

Few-shot learning presents a challenge that a classifier must quickly adapt to new classes that do not appear in the training set, given only a few labeled examples of each new class. This paper proposes a position-aware relation network (PARN) to learn a more flexible and robust metric ability for few-shot learning. Relation networks (RNs), a kind of architectures for relational reasoning, can acquire a deep metric ability for images by just being designed as a simple convolutional neural network (CNN) [23]. However, due to the inherent local connectivity of CNN, the CNN-based relation network (RN) can be sensitive to the spatial position relationship of semantic objects in two compared images. To address this problem, we introduce a deformable feature extractor (DFE) to extract more efficient features, and design a dual correlation attention mechanism (DCA) to deal with its inherent local connectivity. Successfully, our proposed approach extents the potential of RN to be position-aware of semantic objects by introducing only a small number of parameters. We evaluate our approach on two major benchmark datasets, i.e., Omniglot and Mini-Imagenet, and on both of the datasets our approach achieves state-of-the-art performance. It's worth noting that our 5-way 1-shot result on Omniglot even outperforms the previous 5-way 5-shot results.

1. Introduction

Humans can effectively utilize prior knowledge to easily learn new concepts given just a few examples. Few-shot learning [11, 20, 15] aims to acquire some transferable knowledge like humans, where a classifier is able to generalize to new classes when given only one or a few labeled examples of each class, *i.e.*, one- or few-shot. In this paper, we focus on the ability of learning how to compare, namely metric-based methods. Metric-based methods [2, 11, 22, 23, 25] often consist of a feature extractor and a metric module. Given an unlabeled query image and a few labeled sample images, the feature extractor first generates embeddings for all input images, and then the metric module measures distances between the query embedding and sample embeddings to give a recognition result.

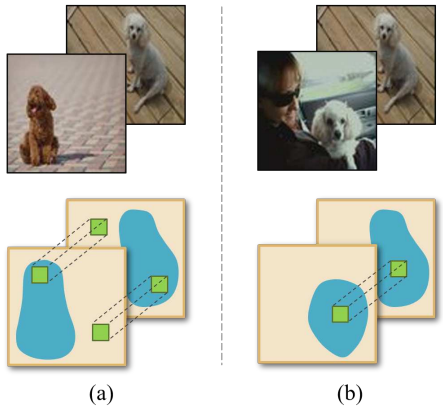


Figure 1: Two situations where the comparison ability of RN will be limited. The top row shows the two compared images, and the bottom row shows their extracted features, where blue areas represent the response of corresponding semantic objects. (a) The convolutional kernel fails to involve the two objects. (b) The convolutional kernel fails to involve the same fine-grained features.

Most existing metric-based methods for few-shot learning focused on constructing a learned embedding space to better adapt to some pre-specified distance metric functions, *e.g.*, cosine similarity [25] or Euclidean distance [22]. These studies expected to learn a distance metric for images, but actually only the feature embedding is learnable. As a result, the fixed but sub-optimal metric functions would limit the feature extractor to produce discriminative representations. Based on this problem, recently Sung *et al*. [23] introduced a relation network, which was designed as a simple CNN, to make the metric learnable and flexible in a data-driven way (in this paper we denote the *simply*

CNN-based relation network as RN), and they achieved impressive performance in few-shot learning. However, according to our analysis, the comparison ability of RN is still limited due to its inherent local connectivity.

As we all know, convolutional operations naturally have the translation invariance to extract features from images, meaning that higher responses of extracted features mainly locate in positions corresponding to the semantic objects [27]. There are two situations: (i) two semantic objects of images are in totally different spatial positions, as shown in Figure 1(a); (ii) they are in close spatial positions while their fine-grained features do not, as shown in Figure 1(b). We note that these two situations commonly occur in the datasets, especially the situation (ii), which should not be overlooked. For these two situations, Sung *et al*. [23] simply concatenated two compared features together and used RN to learn their relationship. However, we argue that the comparison ability of RN is inherently constrained due to its local receptive fields. In situation (i), as shown in Figure 1(a), each convolution step can only involve a same local spatial region, which rarely contains two objects at the same time. In situation (ii), even if the convolutional kernel involves two objects simultaneously, it may also fail to involve their related fine-grained semantic features, *e.g*., in Figure 1(b) it involves body features of the sample and head features of the query, which is not optimal and reasonable as a comparison operation. These two situations motivate us to promote RN aware of objects and fine-grained features in different positions.

In this paper, we propose a position-aware relation network (PARN), where the convolution operator can overcome its local connectivity to be position-aware of related semantic objects and fine-grained features in images. Compared with RN [23], our proposed model provides a more efficient feature extractor and a more robust deep metric network, which enhances the generalization capability of the model to deal with the above two situations. The overall framework is shown in Figure 2. Our main contributions are as follows:

- During the feature extraction phase, we introduce the deformable feature extractor (DFE) to extract more efficient features, which contain fewer low-response or unrelated semantic features, for effectively alleviating the problem in the situation (i).
- Our another important contribution is that we further exploit the potential of RN to be position-aware to learn a more robust and general metric ability. During the comparison phase, we propose a dual correlation attention mechanism (DCA) that utilizes position-wise relationships of two compared features to capture their global information, and then densely aggregate the captured information into each position of outputs. In this way, the subsequent convolutional layer can sense related fine-grained features in all positions, and adaptively compare them despite of the local connectivity.
- With the setting of using a shallow feature extraction network, our method achieves state-of-the-art results with a comparable margin on two major benchmarks, *i.e*., Omniglot and Mini-Imagenet. It's worth noting that our 5-way 1-shot result on Omniglot even outperforms the previous 5-way 5-shot results.

2. Related Work

Recent methods for few-shot learning usually adopted the *episode*-based strategy [25] to learn meta-knowledge from a set of episodes, where each episode/task/mini-batch contains C classes and K samples of each class, *i.e*., C-way K-shot. The acquired meta-knowledge could enable the model to adapt to new tasks that contain unseen classes with only a few samples. According to the variety of meta-knowledge, recent methods could be summarized into the three categories, *i.e*., optimization-based (learning to optimize the model quickly) [6, 18, 28, 29], memory-based (learning to accumulate and generalize experience) [3, 16, 19] and metric-based (learning a general metric) [2, 11, 22, 23, 25] methods.

Briefly, optimization-based methods usually associated with the concept of meta-learning/learning to learn [7, 24], *e.g*., learning a meta-optimizer [18] or taking some wise optimization strategies [6, 28, 29], to better and faster update the model for new tasks. Memory-based methods generally introduced memory components to accumulate experience when learning old tasks and generalize them when performing new tasks [3, 16, 19]. Our experimental results show that our method outperforms them without the need for updating the model for new tasks or introducing complicated memory structure.

Metric-based methods, where our approach belongs to, can perform new tasks in a feed-forward manner, which often consist of a feature extractor and a metric module. The feature extractor first generates embeddings for the unlabeled query image and a few labeled sample images, and then the recognition result is given by measuring distances between the query embedding and sample embeddings in the metric module. Earlier works [2, 11, 22, 25] mostly focused on designing embedding methods or some well-performed but fixed metric mechanism. For example, Bertinetto *et al*. [2] designed a task-adaptive feature extractor for new tasks by utilizing a trained network to predict parameters. And Vinyals *et al*. [25] proposed a learnable attention mechanism by introducing LSTM to calculate fully context embeddings (FCE), and applying softmax over the cosine similarity in the embedding space, which developed the idea of a fully differentiable neural neighbors algorithm. Yet their approach was somewhat complicated. Snell *et al*. [22] then further exceeded them with prototypical

networks by simply learning an embedding space, where prototypical representations of classes could be obtained by directly calculating the mean of samples, and they used Bregman divergences [1] to measure distance, which outperforms the cosine similarity used in [25].

In the above metric-based methods, embeddings would be limited to produce discriminative representations in order to meet the fixed but sub-optimal metric methods. Some approaches [4, 14] tried to adopt the Mahalanobis metric, while still inadequate in the high-dimensional embedding space. To solve this problem, Sung *et al*. [23] introduced relation networks (RNs) for few-shot learning, which are a kind of architectures for relational reasoning and successfully applied in visual question answering tasks [17, 20, 30]. They achieved impressive performance by designing a simply CNN-based relation network (RN) to develop a learnable non-linear metric module, which is simple but flexible enough for the embedding network. However, due to the local connectivity of CNN, RN would be sensitive to the spatial position relationship of compared objects. Therefore, we further exploit the potential of RN to learn a more robust metric ability, which avoids this problem.

3. Approach

In this section, we give the details of the proposed position-aware relation network (PARN) for few-shot learning. At first, we will present the overall framework of PARN. Then we will introduce our deformable feature extractor (DFE) which could extract more efficient features. At last, to promote RN position-aware of fine-grained features in images, we propose a dual correlation attention mechanism (DCA).

3.1. Overall

The network architecture is given in Figure 2. At first, a sample and a query image are fed into a feature extraction network, which is designed as a DFE. With DFE, extracted features $\boldsymbol{f_1}$ and $\boldsymbol{f_2}$ can be more focused on the semantic objects, which is beneficial to improve the subsequent comparison efficiency and precision.

Then, in order to make a robust comparison between $\boldsymbol{f_1}$ and $\boldsymbol{f_2}$, we apply the dual correlation attention module (DCA) over them, so that each position of the output feature map $\boldsymbol{f_{mn}}(\boldsymbol{m}, \boldsymbol{n} \in \{1, 2\})$ contains global cross- or self-correlation information, where $\boldsymbol{f_{mn}}$ means that each position of $\boldsymbol{f_m}$ attends to all positions of $\boldsymbol{f_n}$. In this way, even if the subsequent convolution operations are locally connected, each convolution step can adaptively sense related fine-grained semantic features in all positions.

Finally, we concatenate the above output features $\boldsymbol{f_{mn}}(\boldsymbol{m}, \boldsymbol{n} \in \{1, 2\})$, and feed them into a standard CNN to learn the relation score.

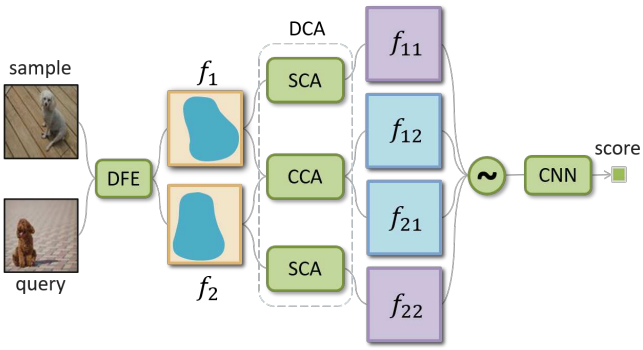


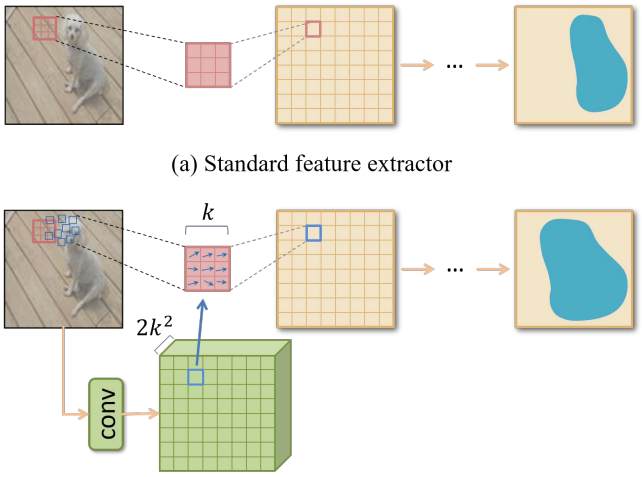
Figure 2: Overview of our proposed PARN for few-shot learning. DFE is the deformable feature extractor. DCA is the dual correlation attention module, which consists of a cross-correlation attention module (CCA) and a self-correlation attention module (SCA). The two SCA blocks are a shared module. The symbol '$\sim$' represents a concatenating operation.

3.2. Deformable Feature Extractor

Figure 3(a) shows a standard feature extractor (SFE). Due to the translation invariance of convolutional operations, the output feature extracted by SFE would only present high responses in spatial positions corresponding to the object. Other positions are low-response or unrelated features that may induce the metric module to perform some redundant comparison operations on them, which affects the efficiency of the comparison. In the worst scenario like Figure 1(a), it is difficult to accurately compare the two objects.

Inspired by the idea of deformable convolutional networks [5, 9] for object detection tasks, we try to deploy deformable convolutional layers for the feature extraction network to extract more efficient features that contain fewer low-response or unrelated semantic features. As shown in Figure 3(b), the convolutional kernel of a deformable convolutional layer is not a regular $k \times k$ grid, but k^2 parameters with 2D offsets. Each parameter $\boldsymbol{w_i}(0 \leq i \leq k^2)$ of the kernel should take an offset coordinate $(\Delta x, \Delta y)$, transforming the original operation from $\boldsymbol{w_i} * \boldsymbol{f_{(x,y)}}$ to $\boldsymbol{w_i} * \boldsymbol{f_{(x+\Delta x, y+\Delta y)}}$, where $\boldsymbol{f_{(x,y)}}$ refers to a spatial point at the coordinate (x, y) of $\boldsymbol{f}$. In our work, the offsets are learned by applying a convolutional layer over the input feature map following Dai *et al*. [5]. And the offsets map has the same spatial resolution as the output map, while its channel dimension is $2k^2$, since for every spatial position of the output map there are $k \times k \times 2 = 2k^2$ offset scalars.

Comparing the features extracted by SFE and DFE in Figure 3(a)(b), we can learn that DFE can filter out unrelated information to some extent, and extract a more efficient feature, which is expected to improve the subsequent comparison efficiency and performance.



(a) Standard feature extractor

(b) Deformable feature extractor

Figure 3: Two feature extractors. Feature maps are shown in spatial shapes. Blue areas on output features represent the response of corresponding semantic objects.

3.3. Dual Correlation Attention Module

Despite of more efficient features, as mentioned in Section 1, if we just use convolutional operations to implement the subsequent comparison procedure, the comparison ability is still limited, since it is somewhat difficult to involve related fine-grained semantic features of the two images at each convolution step. To deal with this problem, one immediate idea is to use a larger receptive field by enlarging the size of the convolutional kernel, or stacking several convolutional layers. However, with more parameters and deeper layers, the model will fall into overfitting problems more easily.

Inspired by the non-local networks [26] that captures long-term dependencies for video classification task, we propose a dual correlation attention mechanism (DCA) for the two-input deep relation network. The proposed attention mechanism uses just a small number of parameters to capture relationships between any two positions of features, regardless of their spatial distance, and then utilizes the captured position-wise relationships to aggregate global information at each spatial position of outputs. In this way, even if the subsequent convolutional kernel is small, each convolution step can involve global information of the two input features, and adaptively perform the comparison on them.

As shown in Figure 2, the proposed DCA consists of a cross-correlation attention module (CCA) and a self-correlation attention module (SCA), where CCA calculates $\boldsymbol{f_{12}}$ (or $\boldsymbol{f_{21}}$) by attending every spatial position of $\boldsymbol{f_1}$ (or $\boldsymbol{f_2}$) to the global information of $\boldsymbol{f_2}$ (or $\boldsymbol{f_1}$), and SCA calculates $\boldsymbol{f_{11}}$ (or $\boldsymbol{f_{22}}$) by attending every spatial position of $\boldsymbol{f_1}$ (or $\boldsymbol{f_2}$) to the global information of its own. We will give their details respectively below.

Cross-correlation attention module As shown in Figure 4, given two extracted features $\boldsymbol{f_1} \in \mathbb{R}^{C\times H_1\times W_1}$ and $\boldsymbol{f_2} \in \mathbb{R}^{C\times H_2\times W_2}$[1], CCA first applies two shared 1×1 convolutional layers over them respectively to make a embedding over the channel dimension, and then generates two feature maps $\boldsymbol{f_1'} \in \mathbb{R}^{C'\times H_1\times W_1}$ and $\boldsymbol{f_2'} \in \mathbb{R}^{C'\times H_2\times W_2}$, where C' is less than C. We reshape them into $\boldsymbol{f_1'} \in \mathbb{R}^{H_1W_1\times C'}$ and $\boldsymbol{f_2'} \in \mathbb{R}^{H_2W_2\times C'}$. Then we apply a cross-interrelation operation $\boldsymbol{g(f_1', f_2')}$ to calculate their relationships of any two positions into the cross-attention map $\boldsymbol{A^c}$. From the spatial position $\boldsymbol{i}$ of $\boldsymbol{f_1'}$ and $\boldsymbol{j}$ of $\boldsymbol{f_2'}$, we can respectively get two spatial points/vectors $\{\boldsymbol{f_{1i}'}, \boldsymbol{f_{2j}'}\} \in \mathbb{R}^{C'}$, where $\boldsymbol{i} \in \{1, ..., H_1W_1\}, \boldsymbol{j} \in \{1, ..., H_2W_2\}$. The pointwise calculation of $\boldsymbol{g(f_1', f_2')}$ is denoted as $\boldsymbol{g_{ij}(f_{1i}', f_{2j}')}$, *i.e.*, $\boldsymbol{g_{ij}}$ computes the value of $\boldsymbol{A^c_{ij}}$, which indicates the relationship between $\boldsymbol{f_{1i}'}$ and $\boldsymbol{f_{2j}'}$. Here we choose the cosine similarity function for $\boldsymbol{g_{ij}}$ to calculate their relationships, then $\boldsymbol{A^c_{ij}}$ can be computed as follows:

$$\boldsymbol{A^c_{ij} = g_{ij}(f_{1i}', f_{2j}') = \overline{f}_{1i}' \overline{f}_{2j}'^{T}} \tag{1}$$

where $\boldsymbol{\overline{f}_{1i}' = \frac{f_{1i}'}{\|f_{1i}'\|}}$ and $\boldsymbol{\overline{f}_{2j}' = \frac{f_{2j}'}{\|f_{2j}'\|}}$ are the l_2-normalized vectors. We denote $\boldsymbol{\overline{f}_1' = [\overline{f}_{1i}'] \in \mathbb{R}^{H_1W_1\times C'}}$ and $\boldsymbol{\overline{f}_2' = [\overline{f}_{2j}'] \in \mathbb{R}^{H_2W_2\times C'}}$, meaning that $\boldsymbol{\overline{f}_1'}$ and $\boldsymbol{\overline{f}_2'}$ are obtained by performing l_2-normalization over $\boldsymbol{f_1'}$ and $\boldsymbol{f_2'}$ respectively along their channel dimension. Then Eq. (1) can be rewritten in matrix form:

$$\boldsymbol{A^c = g(f_1', f_2') = \overline{f}_1' \overline{f}_2'^{T}} \tag{2}$$

where $\boldsymbol{A^c} \in \mathbb{R}^{H_1W_1\times H_2W_2}$ contains all the correlationships between every spatial position of $\boldsymbol{f_1'}$ and $\boldsymbol{f_2'}$.

After obtaining the cross-attention map $\boldsymbol{A^c}$, as shown in Figure 4, the next step is the distribution operation that performs dot-product between each sub-map of $\boldsymbol{A^c}$ with $\boldsymbol{f_1'}$ and $\boldsymbol{f_2'}$ respectively. We perform the distribution as follows:

$$\begin{cases} \boldsymbol{f_{21} = A^{cT} f_1'} \\ \boldsymbol{f_{12} = A^c f_2'} \end{cases} \tag{3}$$

where $\boldsymbol{f_{mn}}$ means that $\boldsymbol{f_m}$ attends to the global information of $\boldsymbol{f_n}$ $(\boldsymbol{m, n} \in \{1, 2\}, \boldsymbol{m \neq n})$. Specifically, we can learn from Figure 4 that the output feature $\boldsymbol{f_{21}}$ captures the global information of $\boldsymbol{f_1}$ into each its spatial position, and so does $\boldsymbol{f_{12}}$ to $\boldsymbol{f_2}$. In this way, the subsequent convolutional layer can sense all the positions, and compare

[1] Actually H_1 and W_1 are equal to H_2 and W_2. Here we denote them as different notations for clear explanation.

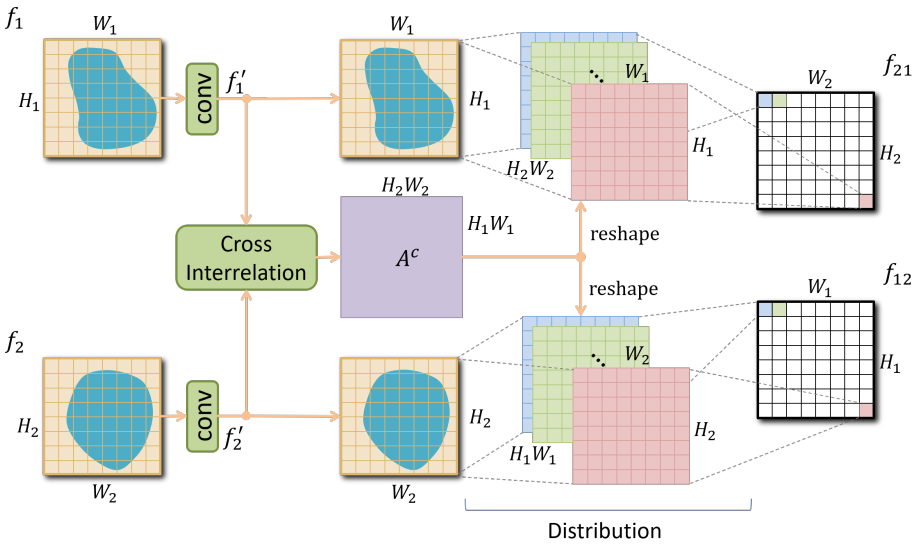


Figure 4: The cross-correlation attention module (CCA). Feature maps are shown in spatial shapes. Weights of the two 1×1 convolutional layers are shared. The cross-correlation attention map $\boldsymbol{A^c}$ contains all the position-wise correlationships of the two inputs. During the distribution operation, $\boldsymbol{A^c}$ will be reshaped into shapes corresponding to the spatial shape of $\boldsymbol{f_1}$ (or $\boldsymbol{f_2}$). Each sub-map of $\boldsymbol{A^c}$ is then performed dot-product with $\boldsymbol{f_1'}$ (or $\boldsymbol{f_2'}$) to aggregate cross-global information into each spatial position of the output $\boldsymbol{f_{21}}$ (or $\boldsymbol{f_{12}}$).

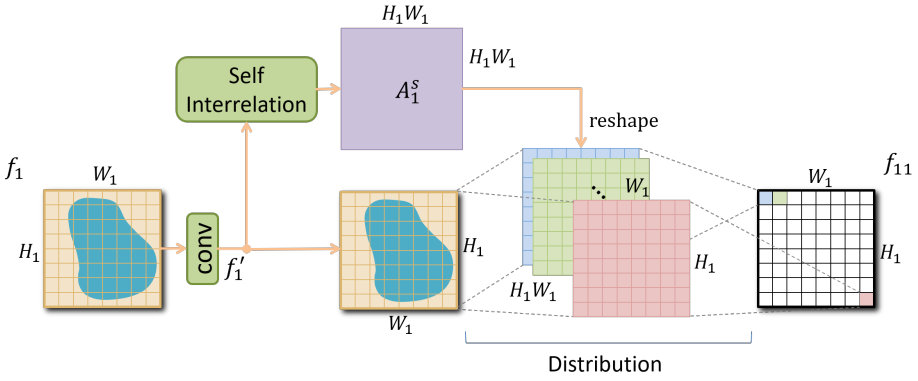


Figure 5: The self-correlation attention module (SCA). Feature maps are shown in spatial shapes. Weights of the 1×1 convolutional layer are shared with that in CCA. The self-correlation attention map $\boldsymbol{A_1^s}$ contains all the position-wise relationships in $\boldsymbol{f_1}$. Each sub-map of $\boldsymbol{A_1^s}$ is then performed dot-product with $\boldsymbol{f_1'}$ to aggregate global information into each spatial position of the output $\boldsymbol{f_{11}}$.

them even with a small convolutional kernel. At last $\boldsymbol{f_{21}}$ and $\boldsymbol{f_{12}}$ will be reshaped into $\boldsymbol{f_{21}} \in \mathbb{R}^{C' \times H_2 \times W_2}$ and $\boldsymbol{f_{12}} \in \mathbb{R}^{C' \times H_1 \times W_1}$ respectively, and then pass through a 1×1 convolutional layer to increase the channel dimension to C.

Self-correlation attention module As shown in Figure 5, SCA is similar to CCA in Figure 4, except that the self-interrelation operation in SCA accept only one input to generate a self-attention map $\boldsymbol{A^s}$, which is actually the case when two inputs of the cross-interrelation operation are the same in our implementation. Besides, the weights of the two 1×1 convolutional layers in SCA are shared with that in CCA. Therefore, referring to Eq. (2)(3), given the input feature $\boldsymbol{f_1}$, we can also get the output $\boldsymbol{f_{11}}$:

$$\boldsymbol{A_1^s} = g(\boldsymbol{f_1'}, \boldsymbol{f_1'}) = \overline{\boldsymbol{f}}_1' \overline{\boldsymbol{f}}_1'^{\boldsymbol{T}} \tag{4}$$

$$\boldsymbol{f_{11}} = \boldsymbol{A_1^{sT}} \boldsymbol{f_1'} \tag{5}$$

where $\boldsymbol{f_{11}}$ means $\boldsymbol{f_1}$ attends to itself, and captures the global information to aggregate into each its spatial position. By inputting $\boldsymbol{f_2}$ and performing the same operations, we can also get $\boldsymbol{A_2^s}$ and $\boldsymbol{f_{22}}$. The next step for $\boldsymbol{f_{11}}$ and $\boldsymbol{f_{22}}$ is the same as for $\boldsymbol{f_{12}}$ and $\boldsymbol{f_{21}}$.

Then the computations of DCA are completed, where all the introducing parameters are only one shared 1×1 con-

volutional layer for embedding input features and another shared 1×1 convolutional layer for increasing the channel dimension. After that, we concatenate these four globally related features $\boldsymbol{f}_{mn}(\boldsymbol{m}, \boldsymbol{n} \in \{1, 2\})$[2] and pass through a CNN to learn the final relation score.

4. Experiments

In this section, we first introduce two benchmark datasets and implementation details. Then we conduct a series of ablation studies to analyze the effectiveness of our proposed model. Finally we compare our proposed model with previous state-of-the-art methods on these two datasets.

4.1. Datasets

Omniglot [12] is a common benchmark for few-shot learning, which contains 1,623 different handwritten characters/classes from 50 different alphabets, and each class has a maximum of 20 samples of size 28×28. We follow the standard splits [22, 23, 25] that there are 1,200 classes for meta-training and 423 classes for meta-testing. In addition, we follow [19, 22, 25] to augment the dataset with random rotations by multiples of 90 degrees during training.

Mini-Imagenet [25] is a subset of Imagenet, consisting of 100 classes, each of which contains 600 images of size 84×84. We follow [6, 18, 22, 23, 25] in the exactly same way to split the dataset, *i.e.*, 64 classes for meta-training, 16 classes for meta-validation and 20 classes for meta-testing.

4.2. Implementation Details

Network architectures Following the previous works [22, 23, 25], our basic feature extraction network, the standard feature extractor (SFE), consists of 4 convolutional modules, each of which contains a 64-filter of 3×3 convolutions, followed by batch normalization [8] and ReLU nonlinearity. Besides, we apply 2×2 max-pooling in the last two layers. As for the basic relation network (RN), we follow the same architecture in [23], namely two convolutional modules with 64-filter, followed by two fully connected layers, and the final output is mapped into 0-1 as the relation score through a sigmoid function.

Training and testing details We implement all the experiments in Pytorch with a GeForce GTX 1080 Ti GPU. We use Adam [10] to optimize the network end-to-end, starting with a learning rate of 0.001 and reducing it by a factor of 10 when the validation accuracy stopped improving. We use the mean square error (MSE) loss to train the network as a regression task, where the label is 1 when the two input categories are the same, otherwise 0. No regularization techniques such as dropout or l_2 regularisation are applied during training. We follow Sung *et al*. [23] to arrange the number of sample and query images for the 1-shot and 5-shot tasks. The classification result is given by the category with the highest score.

[2]In our experiments we also concatenate the two input features.

4.3. Ablation Study

In this subsection, we do some ablation experiments on Mini-Imagenet to examine the effectiveness of DFE and DCA.

Deformable feature extractor In Section 3.2, we propose DFE to extract more efficient features, which is expected to improve the subsequent comparison efficiency and precision. To validate the expectation, we observe the results of using SFE with 4 convolutional layers (SFE-4) or DFE with 4 convolutional layers (DFE-4) to extract features for the subsequent comparison. The structures of SFE-4 and DFE-4 are the same, except that the last two convolutional layers of DFE-4 are deformable convolutional layers. To eliminate the influence of extra parameters introduced by DFE-4, we set up SFE with 6 convolutional layers (SFE-6) for comparison. In this ablation experiment, we just use RN without DCA as the metric network. As we find that the learning of deformable convolutional layers tends to be unstable at the begining, we initialize the parameters of the convolutional layer that learns offsets to be 0 and start training them after about 10000 episodes of warm-up.

The results are shown in Table 1. It can be seen that by using DFE, the accuracies are improved from 51.64% to 52.07% in the 5-way 1-shot task and 66.08% to 67.53% in the 5-way 5-shot task, and slightly better than SFE-6 that holds more parameters, which indicates the effectiveness of DFE. In Figure 6, we further visualize the effective

Model	5-way 1-shot	5-way 5-shot	params	depth
SFE-4	$51.64 \pm 0.83\%$	$66.08 \pm 0.69\%$	0.424M	4
SFE-6	$51.74 \pm 0.84\%$	$67.13 \pm 0.67\%$	**0.498M**	**6**
DFE-4	$\mathbf{52.07 \pm 0.82\%}$	$\mathbf{67.53 \pm 0.67\%}$	0.445M	4

Table 1: The ablation study of DFE on Mini-Imagenet. Results are obtained by averaging over 600 test episodes with 95% confidence intervals.

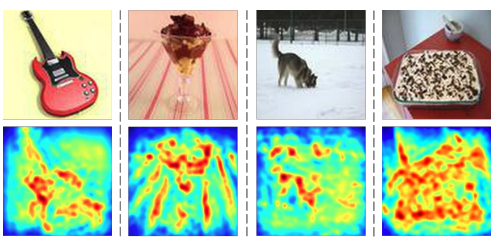


Figure 6: Visualization of the effective receptive fields (ERF) [13] of DFE. DFE can filter out some useless information, such as the background.

receptive field (ERF) [13] of DFE on the input images. The visualization shows that the learned offsets in the deformable convolutional layers can potentially adapt to the image object, meaning that DFE can filter out some useless information to extract more efficient features, which helps the subsequent comparison procedure. Note that ERF does not represent the response of extracted features, but just represents the effective area in the receptive field, that is, the network is watching at these places. So it is acceptable if DFE just filters out some background information, but does not exactly focus on desirable objects.

Dual correlation attention mechanism In this ablation experiment, we take SFE as the feature extractor and RN as the basic metric network. So when no proposed attention module is used, the overall network is our reimplementation of RN in [23]. To verify our proposed DCA, we conduct experiments on whether RN is applied with CCA, SCA or their combination DCA. For fair comparison, a simple 1×1 convolutional layer will be added before RN as the baseline of the proposed attention modules.

The results are shown in Table 2. We can see that in 1-shot and 5-shot tasks, the proposed CCA and SCA both improve the performance. Especially when combining the two modules as DCA, the accuracies increase to 54.36% in the 1-shot task and 70.50% in the 5-shot task, which outperforms the baseline by a clear margin. Besides, we find that during training the network converges much faster with DCA, indicating that DCA successfully allows RN to perceive related semantic features in different positions, and makes it easier to learn to compare.

To more intuitively observe the effectiveness of DCA, we use the gradient-weighted class activation mapping (Grad-CAM) introduced in [21] to visualize the output result activations on the two compared images. As shown in Figure 7, when the related fine-grained semantic features of two objects are in different positions, RN fails to compare them without our proposed DCA, while with DCA it can successfully do it. In other words, with the proposed DCA, RN become more robust and general to learn metrics.

It is worthy to notice that CCA works much better than SCA as shown in Table 2. We analyze that the main reason may attribute to the certain ability of preliminary comparison of CCA, while SCA does not have it. As mentioned in Section 3.3, the cross-attention map $\boldsymbol{A^c}$ of CCA is calculated by the cross-interrelation operation $\boldsymbol{g(f_1, f_2)}$, which is actually implemented by a similarity function. Therefore, when two input features come from different categories, most values of $\boldsymbol{A^c}$ will tend to be smaller. Then in Eq. (3), since $\boldsymbol{f_1^{'}}$ and $\boldsymbol{f_2^{'}}$ are relatively stable after the BN [8] layer of SFE, we can infer that the response of $\boldsymbol{f_{12}}$ and $\boldsymbol{f_{21}}$ will tend to be lower due to the small $\boldsymbol{A^c}$. In other words, inputs of different categories lead to small outputs. While the situation is opposite when $\boldsymbol{f_1}$ and $\boldsymbol{f_2}$ come from the same category. So we can learn that the outputs of CCA have preliminarily represented the relationship between the two inputs, which can help the subsequent RN to make further comparisons.

Method	5-way Acc.	
	1-shot	5-shot
RN	$51.64 \pm 0.83\%$	$66.08 \pm 0.69\%$
baseline	$51.29 \pm 0.82\%$	$66.00 \pm 0.70\%$
SCA	$52.64 \pm 0.91\%$	$67.14 \pm 0.70\%$
CCA	$53.88 \pm 0.87\%$	$69.49 \pm 0.69\%$
CCA&SCA	$\mathbf{54.36 \pm 0.84\%}$	$\mathbf{70.50 \pm 0.64\%}$

Table 2: The ablation study of DCA on Mini-Imagenet. The baseline is a 1×1 convolutional layer with RN. The combination of SCA and CCA is the proposed DCA. Results are obtained by averaging over 600 test episodes with 95% confidence intervals.

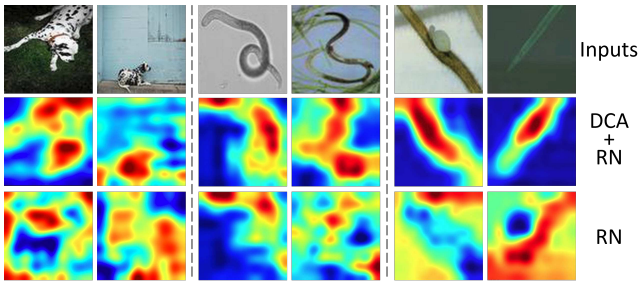


Figure 7: Three Visualization examples of the Gradient-weighted Class Activation Mapping (Grad-CAM) [21] on two input images for RN with or without DCA. With DCA, RN successfully compares related semantic features of two images in different positions, while without DCA it fails to do it.

Besides, as mentioned in Section 1, we propose DFE to handle the situation (i) where two objects are in different positions, and DCA to deal with the situation (ii) where related fine-grained features are in different positions. Comparing the results of DFE in Table 1 and DCA in Table 2, we can find that DCA contributes much more than DFE. According to our analysis, one reason is that in datasets the situation (ii) occurs more commonly than the situation (i), so the effect of DCA can be more apparent. Another reason is that since DCA can compare related features in any position, it naturally has a certain ability to deal with the situation (i). In other words, DCA is general for the two situations.

4.4. Comparison with the State-of-the-arts

In this subsection, we combine DFE and RN with DCA as our proposed position-aware relation network (PARN) to compare with previous state-of-the-art approaches on Mini-Imagenet and Omniglot.

Mini-Imagenet The results on Mini-Imagenet are summarized in Table 4. The first three methods in Table 4

Method	5-way Acc.		20-way Acc.	
	1-shot	5-shot	1-shot	5-shot
MANN [19]	82.8%	94.9%	-	-
Matching Nets [25]	98.1%	98.9%	93.8%	98.5%
Siamese Nets [11]	98.4%	99.6%	95.0%	98.6%
Meta Nets [16]	98.95%	-	97.0%	-
Proto Nets [22]	97.4%	99.3%	95.4%	98.7%
MAML [6]	$98.7 \pm 0.4\%$	$99.9 \pm 0.1\%$	$95.8 \pm 0.3\%$	$98.9 \pm 0.2\%$
MMNet [3]	$99.28 \pm 0.08\%$	$99.77 \pm 0.04\%$	$97.16 \pm 0.10\%$	$98.93 \pm 0.05\%$
RN [23]	$99.6 \pm 0.2\%$	$99.8 \pm 0.1\%$	$97.6 \pm 0.2\%$	$99.1 \pm 0.1\%$
Meta-GAN [28]	$99.67 \pm 0.18\%$	$99.86 \pm 0.11\%$	$97.64 \pm 0.17\%$	$99.21 \pm 0.1\%$
PARN(ours)	$\mathbf{99.91 \pm 0.08\%}$	$\mathbf{99.93 \pm 0.03\%}$	$\mathbf{98.55 \pm 0.18\%}$	$\mathbf{99.48 \pm 0.05\%}$

Table 3: Few-shot classification accuracies on Omniglot. Results are mean accuracies over 1000 test episodes with 95% confidence intervals. '-': not reported

Method	5-way Acc.	
	1-shot	5-shot
Meta-LSTM [18]	$43.44 \pm 0.77\%$	$60.60 \pm 0.71\%$
MAML [6]	$48.70 \pm 1.84\%$	$63.11 \pm 0.92\%$
Meta-GAN [28]	$52.71 \pm 0.64\%$	$68.63 \pm 0.67\%$
MMNets [3]	$53.37 \pm 0.48\%$	$66.97 \pm 0.35\%$
Matching Nets [25]	$43.40 \pm 0.78\%$	$51.09 \pm 0.71\%$
Matching Nets FCE [25]	$43.56 \pm 0.84\%$	$55.31 \pm 0.73\%$
Proto Nets [22][1]	$44.53 \pm 0.76\%$	$65.77 \pm 0.70\%$
Proto Nets [22][2]	$49.42 \pm 0.78\%$	$68.20 \pm 0.66\%$
RN [23]	$50.44 \pm 0.82\%$	$65.32 \pm 0.70\%$
RN[3]	$51.64 \pm 0.83\%$	$66.08 \pm 0.69\%$
PARN(ours)	$\mathbf{55.22 \pm 0.84\%}$	$\mathbf{71.55 \pm 0.66\%}$

[1] Trained with 5-way 15 queries per episode task, which is the same as us.
[2] Trained with 30-way 15 queries per episode task.
[3] Our reimplementation of RN [23].

Table 4: Few-shot classification accuracies on Mini-Imagenet. Results are mean accuracies over 600 test episodes with 95% confidence intervals.

are optimization-based, and the fourth method (MMNets) is memory-based. Others methods, including ours, are metric-based. The result of our reimplementation of RN [23] is better than the reported because our 2×2 max-pooling layers are applied in the last two layers but not the first two, and avoid premature loss of information. Compared with the optimization-based [6, 18, 28] and memory-based methods [3], our proposed PARN achieves better accuracies without the need for updating the model for new tasks or introducing complicated memory structure. As for metric-based methods, after combining DFE and DCA, PARN improves RN from 51.64% to 55.22% in the 1-shot task and 66.08% to 71.55% in the 5-shot task, and defeats all the other metric-based methods by a clear margin. In summary, our proposed method achieves state-of-the-art performance.

Omniglot The experimental results on Omniglot are shown in Table 3. Most previous methods have performed quite well on the Omniglot dataset. However, in all 1-shot and 5-shot tasks, our method still outperforms them by a comparable margin and reaches state-of-the-art results. It is worthy to notice that our 5-way 1-shot result even outperforms the previous 5-way 5-shot results.

5. Conclusion

In this paper, we propose the position-aware relation network (PARN), a more effective and robust deep metric network for few-shot learning. Firstly, we introduce the deformable feature extractor (DFE) to extract more efficient features, which is beneficial for the subsequent comparison efficiency and precision. Secondly, by introducing only a small number of parameters, our proposed dual correlation attention mechanism (DCA) helps RN overcome its inherent local connectivity to compare related semantic objects or fine-grained features in different positions. Therefore, our model is more flexible and robust to learn metrics. Last but not least, we validate our proposed approach on Omniglot and Mini-Imagenet, which achieves state-of-the-art performance.

6. Acknowledgments

This work is supported in part by the Guangzhou Science and Technology Program key projects (No. 201707010141, 201704020134), GD-NSF (no.2017A030312006), the National Natural Science Foundation of China (Grant No.: 61771201), the Program for Guangdong Introducing Innovative and Enterpreneurial Teams (Grant No.: 2017ZT07X183).

References

[1] Arindam Banerjee, Srujana Merugu, Inderjit S. Dhillon, and Joydeep Ghosh. Clustering with bregman divergences. *Journal of Machine Learning Research (JMLR)*, 2005.

[2] Luca Bertinetto, João F. Henriques, Jack Valmadre, Philip H. S. Torr, and Andrea Vedaldi. Learning feed-forward one-shot learners. In *Advances in Neural Information Processing Systems (NIPS)*, 2016.

[3] Qi Cai, Yingwei Pan, Ting Yao, Chenggang Yan, and Tao Mei. Memory matching networks for one-shot image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.

[4] Dong Chen, Xudong Cao, Liwei Wang, Fang Wen, and Jian Sun. Bayesian face revisited: A joint formulation. In *European Conference on Computer Vision (ECCV)*, 2012.

[5] Jifeng Dai, Haozhi Qi, Yuwen Xiong, Yi Li, Guodong Zhang, Han Hu, and Yichen Wei. Deformable convolutional networks. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2017.

[6] Chelsea Finn, Pieter Abbeel, and Sergey Levine. Model-agnostic meta-learning for fast adaptation of deep networks. In *Proceedings of the 34th International Conference on Machine Learning (ICML)*, 2017.

[7] Sepp Hochreiter, A. Steven Younger, and Peter R. Conwell. Learning to learn using gradient descent. In *International Conference on Artificial Neural Networks (ICANN)*, 2001.

[8] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *Proceedings of the 32nd International Conference on Machine Learning (ICML)*, 2015.

[9] Max Jaderberg, Karen Simonyan, Andrew Zisserman, and Koray Kavukcuoglu. Spatial transformer networks. In *Advances in Neural Information Processing Systems (NIPS)*, 2015.

[10] Diederik Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *Proceedings of International Conference on Learning Representations (ICLR)*, 2015.

[11] Gregory Koch, Richard Zemel, and Ruslan Salakhutdinov. Siamese neural networks for one-shot image recognition. In *Proceedings of the 32nd International Conference on Machine Learning Workshops (ICMLW)*, 2015.

[12] Brenden M. Lake, Ruslan Salakhutdinov, Jason Gross, and Joshua B. Tenenbaum. One shot learning of simple visual concepts. In *Proceedings of the 33th Annual Meeting of the Cognitive Science Society (CogSci)*, 2011.

[13] Wenjie Luo, Yujia Li, Raquel Urtasun, and Richard S. Zemel. Understanding the effective receptive field in deep convolutional neural networks. In *Advances in Neural Information Processing Systems (NIPS)*, 2016.

[14] Thomas Mensink, Jakob J. Verbeek, Florent Perronnin, and Gabriela Csurka. Metric learning for large scale image classification: Generalizing to new classes at near-zero cost. In *European Conference on Computer Vision (ECCV)*, 2012.

[15] Erik G. Miller, Nicholas E. Matsakis, and Paul A. Viola. Learning from one example through shared densities on transforms. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2000.

[16] Tsendsuren Munkhdalai and Hong Yu. Meta networks. In *Proceedings of the 34th International Conference on Machine Learning (ICML)*, 2017.

[17] Rasmus Berg Palm, Ulrich Paquet, and Ole Winther. Recurrent relational networks. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2018.

[18] Sachin Ravi and Hugo Larochelle. Optimization as a model for few-shot learning. In *Proceedings of International Conference on Learning Representations (ICLR)*, 2017.

[19] Adam Santoro, Sergey Bartunov, Matthew Botvinick, Daan Wierstra, and Timothy P. Lillicrap. Meta-learning with memory-augmented neural networks. In *Proceedings of the 33nd International Conference on Machine Learning (ICML)*, 2016.

[20] Adam Santoro, David Raposo, David G. T. Barrett, Mateusz Malinowski, Razvan Pascanu, Peter Battaglia, and Tim Lillicrap. A simple neural network module for relational reasoning. In *Advances in Neural Information Processing Systems (NIPS)*, 2017.

[21] Ramprasaath R. Selvaraju, Michael Cogswell, Abhishek Das, Ramakrishna Vedantam, Devi Parikh, and Dhruv Batra. Grad-cam: Visual explanations from deep networks via gradient-based localization. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2017.

[22] Jake Snell, Kevin Swersky, and Richard S. Zemel. Prototypical networks for few-shot learning. In *Advances in Neural Information Processing Systems (NIPS)*, 2017.

[23] Flood Sung, Yongxin Yang, Li Zhang, Tao Xiang, Philip H. S. Torr, and Timothy M. Hospedales. Learning to compare: Relation network for few-shot learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.

[24] Sebastian Thrun and Lorien Pratt. *Learning to learn*. Springer Science & Business Media, 2012.

[25] Oriol Vinyals, Charles Blundell, Tim Lillicrap, Koray Kavukcuoglu, and Daan Wierstra. Matching networks for one shot learning. In *Advances in Neural Information Processing Systems (NIPS)*, 2016.

[26] Xiaolong Wang, Ross B. Girshick, Abhinav Gupta, and Kaiming He. Non-local neural networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.

[27] Matthew D. Zeiler and Rob Fergus. Visualizing and understanding convolutional networks. In *European Conference on Computer Vision (ECCV)*, 2014.

[28] Ruixiang Zhang, Tong Che, Zoubin Ghahramani, Yoshua Bengio, and Yangqiu Song. Metagan: An adversarial approach to few-shot learning. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2018.

[29] Yabin Zhang, Hui Tang, and Kui Jia. Fine-grained visual categorization using meta-learning optimization with sample selection of auxiliary data. In *European Conference on Computer Vision (ECCV)*, 2018.

[30] Bolei Zhou, Alex Andonian, Aude Oliva, and Antonio Torralba. Temporal relational reasoning in videos. In *European Conference on Computer Vision (ECCV)*, 2018.