Contents lists available at ScienceDirect

Pattern Recognition

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

SRML: Structure-relation mutual learning network for few-shot image classification

Xiaoxu Li [a,c], Lang Wang [a], Rui Zhu [b,*], Zhanyu Ma [c], Jie Cao [a,d], Jing-Hao Xue [e]

[a] *School of Computer and Communication, Lanzhou University of Technology, Lanzhou, 730050, China*
[b] *Faculty of Actuarial Science and Insurance, Bayes Business School, City St George's, University of London, London, EC1Y 8TZ, UK*
[c] *Pattern Recognition and Intelligent System Laboratory, School of Artificial Intelligence, Beijing University of Posts and Telecommunications, Beijing, 100876, China*
[d] *Lanzhou City University, Lanzhou, 730050, China*
[e] *Department of Statistical Science, University College London, London, WC1E 6BT, UK*

ARTICLE INFO

Keywords:
Few-shot image classification
Self-correlation
Cross-correlation
Mutual learning

ABSTRACT

Few-shot image classification aims at tackling a challenging but practical classification setting, where only few labelled images are available for training. Metric-based methods are main-stream solutions for few-shot image classification, but many of them extract features that are either irrelevant to target objects in the query images or insufficient to describe the local shape or structural patterns within images, which can lead to mis-identification of the target objects, especially when the images are of multiple objects. To resolve this issue, we propose the structure-relation mutual learning (SRML) network, which first learns both the intra-image structural features and the inter-image relational features in a parallel fashion via two parallel branches, the structural feature extractor (SFE) and the relational feature extractor (RFE), and then harnesses mutual learning to enable knowledge exchange between them. In such a manner, the structural features learnt from the SFE branch not only contain the structural patterns within the images, but also focus more on the target objects, guided by the relational knowledge from the RFE branch. In return, the RFE branch can exploit the more-focused structural knowledge to better match the target objects in the support and query images. We conduct extensive experiments on four few-shot classification benchmark datasets to showcase the superior classification of the proposed SRML network, achieving a 3.17% improvement in classification accuracy over the leading competitor, RENet Kang et al. (2021). The code of this work can be found in https://github.com/Rilliant7/SRML.

1. Introduction

Traditional deep learning methods for image classification rely on a substantial amount of annotated images to train models that can be well generalised to images unseen in the training phase. However, the cost of annotating such an amount of images is usually prohibitive in practice. Hence, few-shot image classification methods methods have emerged to take up this challenge: to learn from few labelled images and make accurate predictions on images from classes never seen during training [1,2]. Few-shot image classification has broad application potential. For example, in hyperspectral image classification, labelling thousands of pixels per image is impractical [3]. Similarly, annotated medical images for classification are often limited [4].

Deep metric learning is popular in few-shot image classification methods, which aims to learn discriminative feature embeddings with a proper metric space for classifying test images [5]. Classical metric-based methods include the matching networks (MatchNet) [6] and the prototypical networks (ProtoNet) [7] based on the pre-defined cosine similarity and Euclidean distance, respectively. Metrics can also be learnt in more advanced ways. For example, the relation network (RelationNet) [8] learns the metric function from a relation module. The deep nearest neighbour neural network (DN4) [9] utilises a novel image-to-class metric based on local descriptors. The bi-similarity network (BSNet) [10] adopts a dual similarity network to learn different types of similarities. To enhance the discriminative power of the extracted features, extensive works focus on designing novel attention schemes to assign higher weights to discriminative spatial features or channels. For instance, Lee et al. [11] propose the task discrepancy maximisation (TDM) module to learn task-wise channel weights to highlight discriminative channels. Song et al. [12] introduce a fusion spatial attention method to fuse discriminative information in both the image space and the embedded space.

[*] Corresponding author.
E-mail address: rui.zhu@city.ac.uk (R. Zhu).

https://doi.org/10.1016/j.patcog.2025.111822
Received 19 August 2024; Received in revised form 17 April 2025; Accepted 5 May 2025
Available online 17 May 2025

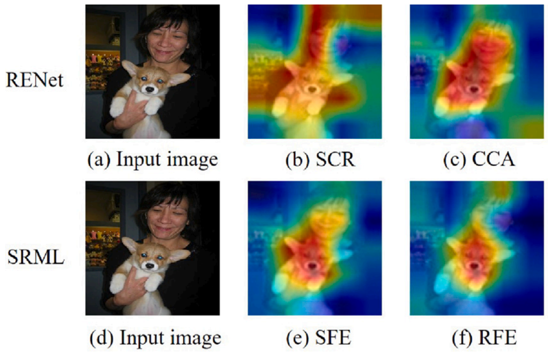

Fig. 1. An example of the features captured (upper row:) by the sequentially combined SCR and CCA modules in RENet [16] and (lower row:) by the parallely combined SFE and RFE branches in our proposed SRML network. The class of the input image is dog.

Although involving attention schemes can raise few-shot classification accuracy, many of these methods extract features of support and query sets separately. This results in potential mis-identification of objects in the meta-test phase, especially when the test images contain multiple objects. Recall that in few-shot image classification, test images are from classes different from those of the training images. Thus the multi-object test images containing some objects also in the training images can be easily mis-classified, since the extracted features tend to stress the objects that have been seen in the training phase. To alleviate this issue, Hou et al. [13] propose the cross attention network (CAN) to match the semantically related objects in the support and query sets and assign more attention to the relevant objects. Such matching in CAN relies on the correlation layer to calculate the cosine similarities between each spatial position of support and query features and identify the related spatial positions. Following a similar logic, Doersch et al. [14] propose the CrossTransformers (CTX) that can spatially align support and query images, while enhancing the base representation via self-supervised learning.

However, barely learning the spatial relationships between support and query images from cross-attention modules ignores the vital shape patterns or structural information within images to assist object identification. The same object in different images can have different textures and colours, but their shapes or structures should remain similar. For example, the same complex action can be performed by different people wearing the different clothes in different backgrounds [15]. To extract the crucial structural patterns for object detection, Shechtman and Irani [15] introduce the local self-similarity descriptor via calculating the correlations between each pixel and its surrounding neighbouring area. Utilising the idea of local self-similarity, Kang et al. [16] propose the relational embedding network (RENet) to extract the self-correlational representations (SCR) to describe the structural patterns within images, and employ these features to match the related objects across support and query images through the cross-correlational attention (CCA) module. RENet demonstrates improvements on classification accuracy compared with CAN and CTX with only cross-attentions.

In RENet, the structural patterns captured in the SCR module usually describe all objects and backgrounds within one image and can contain misleading patterns that are irrelevant to the object to be classified. This could bring difficulties to the CCA module to explore the relationship between related objects, because of the sequential combination of the SCR and CCA modules. For example, in Fig. 1(b) and (c), the features stressed by the SCR module in RENet not only include the target dog, but also pay attention to the human face and background. This leads to an ambiguously focused area in the CCA module that is not entirely on the dog.

To resolve this problem, we propose the structure-relation mutual learning (SRML) network to encourage interactions between the structural patterns learnt within images and the relational features to match target objects across images. To this end, rather than sequentially combining the self- and cross-correlational modules, we structure two parallel branches to learn the within-image and cross-image information, while allowing knowledge exchange between them via mutual learning [17]. Specifically, we employ the bidirectional knowledge exchange (BKE) learning to facilitate the interactions between the features extracted by the structural feature extractor (SFE) and relational feature extractor (RFE) branches. To illustrate the effectiveness of our proposed framework, we adopt the readily available SCR and CCA modules in RENet as feature extractors for the two branches, respectively. The two parallel branches are supervised by their own classification losses to learn the corresponding discriminative features separately. Moreover, the BKE learning strategy forces the outputs of the two branches gradually approach each other by minimising the Kullback–Leibler (KL) divergence. This makes the within-image structural information focuses more on the target objects, which can further help the RFE branches to identify the correct objects. As illustrated in Fig. 1(e) and (f), the SFE branch mainly extracts the structural features of the dog and the RFE branch further limits the focusing area for correct classification. Extensive experiments and ablation studies on four publicly available datasets showcase the superior classification performance of our method.

To sum up, the contributions of our work are three-fold:

- We propose the novel SRML network for few-shot image classification, by leveraging the complementary information between the structural patterns within images and the relational features between the matched objects across images. The parallel integration of the SFE and the RFE with knowledge exchange can generate discriminative features with precise identification of vital spatial regions for classification.
- To encourage the interactions between the two modules, we propose to utilise the BKE learning that can help calibrate biases introduced by individual branches.
- We conduct extensive experiments to validate the effectiveness of SRML. The results demonstrate that the parallel integration of the SFE and RFE branches with BKE learning leads to improved classification performance compared with the state-of-the-art methods for few-shot image classification.

The rest of the paper is organised as follows. In Section 2, we discuss the existing methods that are closely related to our work. In Section 3, we present the technical details of the proposed SRML network. We then show the extensive experimental results and ablation studies in Section 4. Lastly, we draw conclusions in Section 5.

2. Related work

In this section, we first present the state-of-the-art metric-based methods for few-shot image classification. We then discuss literature utilising self-correlational attentions and cross-correlational attentions. Finally, we introduce mutual learning for knowledge exchange.

2.1. Metric-based few-shot image classification

Metric-based methods calculate the distance between the query image and the support set based on a given metric, and assign the query image to the nearest class [18]. For instance, the matching networks (MatchNet) [6] utilise the external memory to enhance the neural network and compute the cosine between support samples and query samples for classification. The prototypical networks (ProtoNet) [7] compute the mean of support images for each class as the class prototypes and classify the query image based on its Euclidean distances to the class prototypes. These models are based on the pre-defined metrics; however, the pre-defined metrics are usually not suitable for real-world tasks.

To address this issue, Sung et al. [8] propose the relation network (RelationNet) based on a convolutional module to find the metric function. Li et al. [10] argue that a single metric may not be sufficient to learn discriminative features, thus they propose the bi-similarity network (BSNet), which combines two different distance metric modules to generate a more compact and discriminative feature space. Cross-view deep nearest neighbour neural network (CDN4) [18] designs four cross-view metric pairs to make features more discriminative. Dong et al. [2] propose to use cross-image semantic alignment to reduce intra-class variation.

Besides learning the metric adaptively from data, more works aim to incorporate attention mechanisms to improve the classification performance. For example, Lee et al. [11] proposed the task discrepancy maximisation (TDM) module for fine-grained few-shot classification, which highlights channels that encode class-specific information to locate the discriminative regions. Lai et al. [19] notice that the CNN structure produces inaccurate attention maps based on local features, and thus they develop a novel SpatialFormer structure that generates more accurate attention regions based on global features.

To make the extracted features more discriminative, the relationship between samples is exploited in a plug-and-play module, enhancing transfer learning and meta-learning based few-shot classification frameworks [20]. A feature re-abstraction embedding (FRaE) module is also developed to effectively amplify the difference between the feature information of different categories [1].

2.2. Cross-attentions and self-attentions

Cross attentions usually aim to exploit the spatial similarities between support and query images that can help to match the target objects. Hou et al. [13] introduce the cross attention network (CAN) to identify the target objects in the unseen classes. It generates cross-attention maps to match support features and query features, making the extracted features more discriminative. Doersch et al. [14] propose a transformers-based neural network named CrossTransformers to find the coarse spatial correspondences between the query and support features, and perform classification by computing the distances between the corresponding features.

However, only considering cross-correlational attentions ignores the structural patterns within images and can lead to mis-identification. To solve this problem, self-correlational attentions are involved, which measure the similarity between a local region in an image and the other parts of the same image in terms of structure or shape. Some works only utilise the self-correlational attentions. For example, Afrasiyabi et al. [21] improve the traditional encoder architectures by embedding the self-attention mechanisms to extract a set of feature vectors for images, enabling better representation of images. Recently, Huang and Choi [22] use the self-attention module to obtain more representative class prototypes than ProtoNet.

Furthermore, there are other works combining the two types of attentions to improve the few-shot classification performance. For instance, RENet [16] combines the two types of attentions in a sequential manner and utilises the structural patterns to obtain the cross-attention maps. Moreover, Huang et al. [23] utilise the self-correlational attention to extract prototype features for each class from the support set. They then combine the self-attention of support images and the mutual attention between prototypes and query images to jointly attend to features of different samples with the same class, expanding class prototypes for more stable feature representations.

In this paper, we propose a novel structure that involves the two types of attentions in a parallel manner and allows interactions between them to obtain more discriminative features. Compared with CAN and CTX, our method incorporates structural patterns through self-correlation attention. Unlike RENet, we adopt a parallel structure to improve the classification of multi-object images.

2.3. Mutual learning for knowledge exchange

Knowledge distillation is a model compression technique that aims to train a small student model under the guidance of a large teacher model, allowing the small student model to achieve comparable performance with low computational costs and storage requirements. Rajasegaran et al. [24] propose the self-supervised knowledge distillation approach to learn representative feature embeddings that can encode inter-class relationships for few-shot image classification. Instead of transferring knowledge from the teacher model to the student model, Zhang et al. [17] propose to let students learn collaboratively and teach each other throughout the training process.

Inspired by this idea, in this paper, we propose to utilise bidirectional knowledge distillation to facilitate mutual knowledge transfer between the SFE and RFE branches, allowing them to learn useful knowledge from each other while learning their own specific knowledge.

3. Methodology

In this section, we first formulate the few-shot image classification problem in Section 3.1. We then describe the technical details of the proposed SRML network, including the overall structure in Section 3.2, the SFE in Section 3.3, the RFE in Section 3.4 and the bidirectional knowledge exchange (BKE) learning in Section 3.5.

3.1. Problem formulation

This paper follows the N-way K-shot setting for few-shot image classification, where we train the model by a series of classification tasks with N classes and each with K images. Specifically, we denote the dataset as $\mathcal{D} = \{\left(\mathbf{x}_i, y_i\right)_{i=1}^{N_T}, y_i \in \mathcal{Y}\}$, where x_i denotes the ith image, y_i is its corresponding label, N_T represents the total number of classes and $\mathcal{Y}$ represents the set of all possible class labels. We randomly divide it to a training set $\mathcal{D}_{\text{train}}$, a validation set $\mathcal{D}_{\text{val}}$ and a test set $\mathcal{D}_{\text{test}}$. The label sets of the three subsets do not intersect with each other. To form a task, we randomly divide the training set to a support set $\mathcal{S}$ with N classes of K images and a query set $\mathcal{Q}$ with N classes of q images. Note that $\mathcal{S}$ and $\mathcal{Q}$ share the same label sets. The same procedure follows for the validation set to choose the best performed model and for the test set to evaluate the classification performance of the chosen model on the unseen classes.

3.2. The overall structure of the SRML network

We illustrate the overall structure of the SRML network in Fig. 2. The support and query images are input to a shared embedding module to extract the base representations $\mathbf{Z}_S$ and $\mathbf{Z}_Q$, respectively. Hereafter, we adopt the subscripts S and Q to denote the quantities related to the support and query images, respectively. $\mathbf{Z}_S$ and $\mathbf{Z}_Q$ are then fed to two parallel branches: the SFE branch to extract the structural patterns within images and the RFE branch to exploit the relationship between the matched objects across images. Note that in this paper, the SFE and RFE branches follow the same structures of the SCR and CCA modules in RENet [16], respectively.

In SFE, we utilise the local self-correlation calculation to obtain the similarities between each pixel and its surrounding area for each image separately, and extract the self-correlation features, $\mathbf{F}_S^{\text{intra}}$ and $\mathbf{F}_Q^{\text{intra}}$, through a convolutional block. In RFE, we identify the correlated spatial regions across the support and query images through a cross attention operation and obtain $\mathbf{f}_S^{\text{inter}}$ and $\mathbf{f}_Q^{\text{inter}}$. Note that $\mathbf{f}_S^{\text{inter}}$ and $\mathbf{f}_Q^{\text{inter}}$ are paired; that is, for each pair of support and query images, we have a specific pair of features.

To encourage knowledge exchange between the two branches, we involve the BKE learning strategy to connect them. The two branches

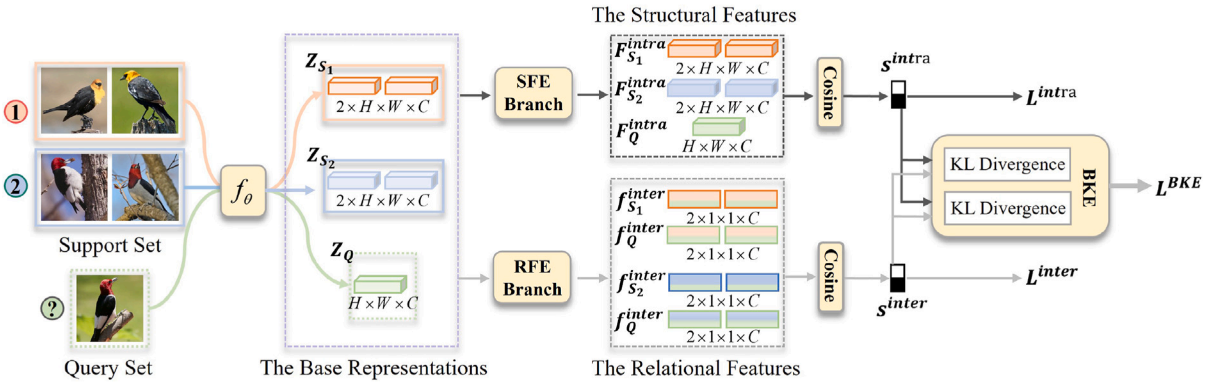

Fig. 2. The overall structure of the SRML network. The support and query images pass through a shared embedding block to generate the base representations $\mathbf{Z}_S$ and $\mathbf{Z}_Q$, respectively. These base representations are then fed to the RFE branch and the SFE branch separately. The RFE branch captures the relational features $\mathbf{f}^{\text{inter}}_S$ and $\mathbf{f}^{\text{inter}}_Q$ to assist identify target objects, while the SFE branch learns the structural features $\mathbf{F}^{\text{inter}}_S$ and $\mathbf{F}^{\text{intra}}_Q$ to better describe the images. The two branches are supervised by their own classification losses, L^{inter} and L^{intra}, respectively. To encourage BKE between the two branches, we calculate the cosine similarities between the query feature and the class prototypes for both branches, and require their distributions to approach each other via minimising the mimicry loss L^{BKE} based on the KL divergence.

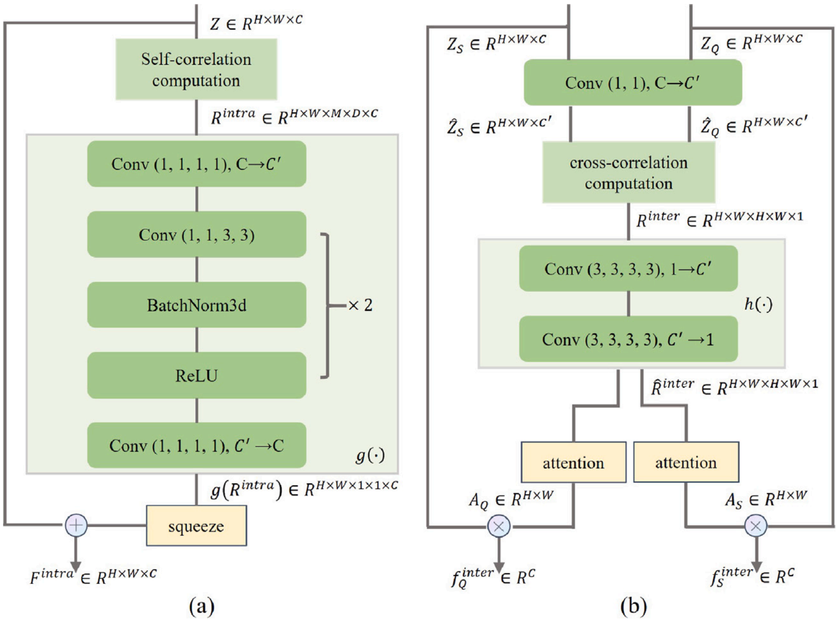
$Z \in R^{H\times W\times C}$
Self-correlation computation
$R^{intra} \in R^{H\times W\times M\times D\times C}$
Conv (1, 1, 1, 1), C→C'
Conv (1, 1, 3, 3)
BatchNorm3d
ReLU
× 2
Conv (1, 1, 1, 1), C' →C
$g(\cdot)$
$g(R^{intra}) \in R^{H\times W\times 1\times 1\times C}$
squeeze
$F^{intra} \in R^{H\times W\times C}$
(a)
$Z_S \in R^{H\times W\times C}$
$Z_Q \in R^{H\times W\times C}$
Conv (1, 1), C→C'
$\hat{Z}_S \in R^{H\times W\times C'}$
$\hat{Z}_Q \in R^{H\times W\times C'}$
cross-correlation computation
$R^{inter} \in R^{H\times W\times H\times W\times 1}$
Conv (3, 3, 3, 3), 1→C'
$h(\cdot)$
Conv (3, 3, 3, 3), C' →1
$\hat{R}^{inter} \in R^{H\times W\times H\times W\times 1}$
attention
attention
$A_Q \in R^{H\times W}$
$A_S \in R^{H\times W}$
$f_Q^{inter} \in R^C$
$f_S^{inter} \in R^C$
(b)

Fig. 3. The structures of the (a) SFE and (b) RFE branches.

are trained by two different classification losses, L^{intra} and L^{inter}, respectively, and the BKE learning is guided by L^{BKE}.

In the end, we adopt the cosine similarity as the metric to compare the query feature and the class prototypes. In the meta-test phase, the support and query images of the unseen classes are fed to the trained model. The cosine similarities between the query image and the class prototypes of the two branches, $\mathbf{s}^{\text{intra}}$ and $\mathbf{s}^{\text{inter}}$, are calculated. The sum of $\mathbf{s}^{\text{intra}}$ and $\mathbf{s}^{\text{inter}}$ is adopted as the final score, i.e. the query image is assigned to the class with the highest sum of similarities.

3.3. The SFE branch

The SFE branch aims to learn feature representations within the images, with a focus on capturing the internal structural patterns. In this section, we omit the subscripts S and Q, as the operations are the same for support and query images.

For a given base representation $\mathbf{Z} \in \mathbb{R}^{H\times W\times C}$, where H and W represent the height and width of the feature map, respectively, and C represents the number of channels, we process its spatial positions as follows. First, we calculate the cosine similarities between each position and its neighbouring window of size $M \times D$ to capture the local correlations, where M and D are the height and width of the neighbouring window, respectively. To handle positions on the edges of the feature map, zero-padding is applied.

Next, these cosine similarities are aggregated to a tensor $\mathbf{R}^{\text{intra}} \in \mathbb{R}^{H\times W\times M\times D\times C}$, containing rich information about the structural patterns within images. Then, the self-correlation learner $g : \mathbb{R}^{H\times W\times M\times D\times C} \to \mathbb{R}^{H\times W\times 1\times 1\times C}$ is applied to $\mathbf{R}^{\text{intra}}$ to extract the structural features, as illustrated in Fig. 3(a). This learner applies a series of convolutions along the $M\times D$ dimensions of $\mathbf{R}^{\text{intra}}$ as follows. Firstly, a point-wise convolutional layer is applied to map $\mathbf{R}^{\text{intra}}$ to a lower-dimensional space with less channels, i.e. $C' < C$ in Fig. 3(a). Secondly, two two-dimensional convolutions are applied to extract the high-level convolved features. Finally, a point-wise convolution is used again to restore the channel dimension to C. By doing so, the self-correlation learner can aggregate the local self-correlation patterns, resulting in $g\left(\mathbf{R}^{\text{intra}}\right) \in \mathbb{R}^{H\times W\times 1\times 1\times C}$. To incorporate the appearance details of the images, we combine the base representation $\mathbf{Z}$ with $g\left(\mathbf{R}^{\text{intra}}\right)$ to obtain the structural features:

$$\mathbf{F}^{\text{intra}} = \hat{g}\left(\mathbf{R}^{\text{intra}}\right) + \mathbf{Z} \in \mathbb{R}^{H\times W\times C}, \tag{1}$$

where $\hat{g}\left(\mathbf{R}^{\text{intra}}\right) \in \mathbb{R}^{H\times W\times C}$ is $g\left(\mathbf{R}^{\text{intra}}\right)$ after squeezing.

In the last step of SFE, we compute the cosine similarity between the query feature and the nth class prototype:

$$s_n^{\text{intra}} = \frac{(\mathbf{p}_n^{\text{intra}})^T \mathbf{f}_Q^{\text{intra}}}{\|\mathbf{p}_n^{\text{intra}}\|_2 \|\mathbf{f}_Q^{\text{intra}}\|_2}, \quad n \in \{1,2,\ldots,N\}, \tag{2}$$

where $\mathbf{p}_n^{\text{intra}}$ is the flattened vector of the nth class prototype, i.e. the simple average of all features $\mathbf{F}_S^{\text{intra}}$ in the nth class of the support set, and $\mathbf{f}_Q^{\text{intra}}$ is the flattened vector of $\mathbf{F}_Q^{\text{intra}}$.

3.4. The RFE branch

The RFE branch aims to match the target object information across support and query images and facilitate subsequent classification. RFE computes the cross correlations for all spatial positions between support and query features and assign higher weights to those positions related to the target objects.

We illustrate the structure of the RFE branch in Fig. 3(b). First, similarly to the SFE branch, we reduce the number of channels of the input features using a 1 × 1 convolution, resulting in $\hat{\mathbf{Z}}_S \in \mathbb{R}^{H\times W\times C'}$ and $\hat{\mathbf{Z}}_Q \in \mathbb{R}^{H\times W\times C'}$, where $C' < C$. Then, we obtain the spatial correlations between $\hat{\mathbf{Z}}_S$ and $\hat{\mathbf{Z}}_Q$ by calculating the cosine similarities between the C' dimensional vectors of each spatial position and storing them in the cross correlation tensor $\mathbf{R}^{\text{inter}} \in \mathbb{R}^{H\times W\times H\times W\times 1}$. Next, we utilise two four-dimensional convolution operations to extract informative features from $\mathbf{R}^{\text{inter}}$:

$$\hat{\mathbf{R}}^{\text{inter}} = h(\mathbf{R}^{\text{inter}}) \in \mathbb{R}^{H\times W\times H\times W\times 1}. \tag{3}$$

Now we are prepared to obtain the cross-correlation attention maps to highlight the most discriminative spatial regions to identify the target objects. For the lth spatial position of the query feature map, we take its correlations with all spatial positions in the support feature map from $\hat{\mathbf{R}}^{\text{inter}}$ and denote them as a vector $\mathbf{q}_l \in \mathbb{R}^{(H\times W)}$ with $l \in \{1,2,\ldots,H\times W\}$. The attention of the lth spatial position of the query feature is then calculated as

$$a_{Q,l} = \frac{1}{HW}\sum_{t=1}^{H\times W} \frac{\exp\left(q_{l,t}/\gamma\right)}{\sum_{l=1}^{H\times W}\exp\left(q_{l,t}/\gamma\right)}, \tag{4}$$

where $q_{l,t}$ is the tth value in $\mathbf{q}_l$ and $t \in \{1,2,\ldots,H\times W\}$ and γ is the parameter for the soft-max function. This operation aims to obtain the average probability that each position of the query feature matches to the overall support feature. We aggregate all $a_{Q,l}$ to form the cross-correlation attention map, $\mathbf{A}_Q \in \mathbb{R}^{H\times W}$ for the query feature. Following a similar strategy but changing the positions of query and support features, we can obtain the attention map $\mathbf{A}_S \in \mathbb{R}^{H\times W}$ for the support set as well.

The final embeddings of the RFE branch, $\mathbf{f}_S^{\text{inter}} \in \mathbb{R}^C$ and $\mathbf{f}_Q^{\text{inter}} \in \mathbb{R}^C$, are obtained via assigning higher weights to spatial positions corresponding to the target object based on the attention maps:

$$\begin{aligned} f_{S,c}^{\text{inter}} &= \sum \text{vec}(\mathbf{A}_S \odot \mathbf{Z}_S^c), \\ f_{Q,c}^{\text{inter}} &= \sum \text{vec}(\mathbf{A}_Q \odot \mathbf{Z}_Q^c), \end{aligned} \tag{5}$$

where $c \in \{1,2,\ldots,C\}$ denotes the cth channel of the feature map, vec is the vectorisation operation, ⊙ is the element-wise Hadamard product, and the sum operation is over all elements in the vector.

Lastly, similarly to the SFE branch, we obtain the similarities between the query feature and the nth class prototype:

$$s_n^{\text{inter}} = \frac{(\mathbf{p}_n^{\text{inter}})^T \mathbf{f}_Q^{\text{inter}}}{\|\mathbf{p}_n^{\text{inter}}\|_2 \|\mathbf{f}_Q^{\text{inter}}\|_2}, \quad n \in \{1,2,\ldots,N\}, \tag{6}$$

where $\mathbf{p}_n^{\text{inter}}$ is the simple average of all features $\mathbf{f}_S^{\text{inter}}$ in the nth class of the support set.

3.5. The BKE learning

To encourage the interactions between the features obtained from the SFE and RFE branches, we propose to tailor the mutual learning strategy to enable knowledge transfer in both directions. In this way, the within-image structural features of the SFE branch can shift their focuses to the target objects, because of the participation of the relational information from the RFE branch. In return, the relational features can be better learnt and the target objects can be precisely matched based on the more focused structural features.

Specifically, we force the distributions of $\mathbf{s}^{\text{intra}}$ in (2) and $\mathbf{s}^{\text{inter}}$ in (6) to be as similar as possible based on the Kullback–Leibler (KL) divergence:

$$\begin{aligned} D_{KL}\left(\mathbf{s}^{\text{intra}} \parallel \mathbf{s}^{\text{inter}}\right) &= \sum_{n=1}^{N} s_n^{\text{intra}} \log \frac{s_n^{\text{intra}}}{s_n^{\text{inter}}}, \\ D_{KL}\left(\mathbf{s}^{\text{inter}} \parallel \mathbf{s}^{\text{intra}}\right) &= \sum_{n=1}^{N} s_n^{\text{inter}} \log \frac{s_n^{\text{inter}}}{s_n^{\text{intra}}}. \end{aligned} \tag{7}$$

By minimising the mimicry loss:

$$L^{\text{BKE}} = D_{KL}(\mathbf{s}^{\text{intra}} \parallel \mathbf{s}^{\text{inter}}) + D_{KL}(\mathbf{s}^{\text{inter}} \parallel \mathbf{s}^{\text{intra}}), \tag{8}$$

we expect that the outputs of the SFE and RFE branches gradually approach each other, thereby exchange the knowledge between each other.

3.6. Loss function

Besides the mimicry loss to match the distributions of the similarities to the class prototypes, the SFE and RFE branches are also supervised by their own classification losses, L^{intra} and L^{inter}, respectively. Each classification loss consists of two parts, the metric-based loss and the anchor-based loss [16]:

$$L^{\text{m}} = \lambda L_{\text{metric}}^{\text{m}} + L_{\text{aux}}^{\text{m}}, \tag{9}$$

where $\lambda = 1.5$ and $\text{m} \in \{\text{intra}, \text{inter}\}$. The metric loss and the anchor loss are calculated as

$$L_{\text{metric}}^{\text{m}} = -\sum_{r=1}^{N\times q} \log P_{\text{metric}}(\hat{y}_l = y_l | \mathbf{x}_{Q,r}), \tag{10}$$

and

$$L_{\text{aux}}^{\text{m}} = -\sum_{r=1}^{N\times q} \log P_{\text{aux}}(\hat{y}_l = y_l | \mathbf{x}_{Q,r}). \tag{11}$$

Here

$$P_{\text{metric}}(\hat{y} = n | \mathbf{x}_Q) = \frac{\exp(s_n^{\text{m}}/\tau)}{\sum_{n'=1}^{N}\exp(s_{n'}^{\text{m}}/\tau)}, \tag{12}$$

and

$$P_{\text{aux}}(\hat{y} = n | \mathbf{x}_Q) = \frac{\exp(\mathbf{w}_n^T \mathbf{f}_Q^{\text{m}} + \mathbf{b}_n)}{\sum_{n'=1}^{N}\exp(\mathbf{w}_{n'}^T \mathbf{f}_Q^{\text{m}} + \mathbf{b}_{n'})}, \tag{13}$$

where $\tau = 0.2$ and $\mathbf{w}$ and $\mathbf{b}$ are hyperparameters to learn.

Finally, the proposed SRML network is trained end-to-end by minimising the overall loss L:

$$L = \alpha L^{\text{intra}} + \beta L^{\text{inter}} + \omega L^{\text{BKE}}, \tag{14}$$

where α, β, and ω are the learnable hyperparameters that control the relative importance of each component.

Hence, in SRML, each branch learns their own discriminative features through the corresponding classification losses, while exploiting the useful information from each other through the BKE loss. The convergence behaviour of the loss function is visualised in Fig. 4, together with the corresponding training accuracy.

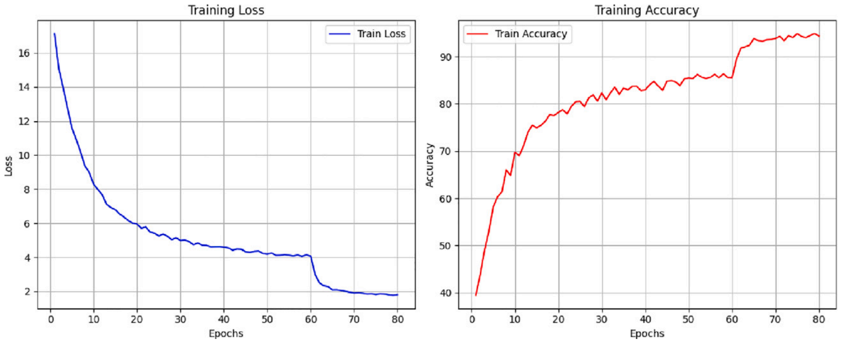

Fig. 4. The training loss and training accuracy of SRML on CUB-200-2011.

Table 1
The 5-way few-shot classification accuracies on the CUB-200–2011, Stanford-Cars, Stanford-Dogs and Flowers datasets for the Conv-4 backbone.

Methods	CUB-200–2011		Stanford-Dogs		Stanford-Cars		Flowers	
	1-shot	5-shot	1-shot	5-shot	1-shot	5-shot	1-shot	5-shot
MatchNet [6]	60.06 ± 0.88	74.57 ± 0.73	46.10 ± 0.86	59.79 ± 0.72	44.73 ± 0.77	64.74 ± 0.72	71.89 ± 0.90	85.46 ± 0.59
ProtoNet [7]	68.27 ± 0.52	83.89 ± 0.33	48.85 ± 0.45	66.89 ± 0.40	59.35 ± 0.47	77.82 ± 0.34	69.34 ± 0.56	83.51 ± 0.41
RelationNet [8]	63.94 ± 0.92	77.87 ± 0.64	47.11 ± 0.90	65.56 ± 0.74	45.83 ± 0.87	68.01 ± 0.78	69.50 ± 0.96	83.91 ± 0.63
DN4 [9]	57.45 ± 0.89	84.41 ± 0.58	39.08 ± 0.76	69.81 ± 0.69	34.12 ± 0.68	87.47 ± 0.47	71.15 ± 0.94	88.86 ± 0.56
Baseline++ [29]	62.36 ± 0.84	79.08 ± 0.61	44.49 ± 0.70	64.48 ± 0.66	46.82 ± 0.76	68.20 ± 0.72	70.54 ± 0.84	86.63 ± 0.58
DeepEMD [30]	64.08 ± 0.50	80.55 ± 0.71	46.71 ± 0.49	65.74 ± 0.63	61.63 ± 0.27	72.95 ± 0.38	–	–
DSN [31]	71.57 ± 0.92	83.51 ± 0.60	44.33 ± 0.81	60.04 ± 0.68	48.16 ± 0.86	60.77 ± 0.75	67.71 ± 0.92	84.58 ± 0.70
CTX [14]	72.61 ± 0.21	86.23 ± 0.14	57.86 ± 0.21	73.59 ± 0.16	66.35 ± 0.21	82.25 ± 0.14	–	–
BSNet [10]	62.84 ± 0.95	85.39 ± 0.56	43.42 ± 0.86	71.90 ± 0.68	40.89 ± 0.77	73.47 ± 0.75	72.79 ± 0.91	84.93 ± 0.64
FRN [32]	73.38 ± 0.21	88.23 ± 0.13	58.48 ± 0.23	76.29 ± 0.16	59.41 ± 0.22	80.60 ± 0.15	72.91 ± 0.22	88.89 ± 0.13
TDM [11]	74.39 ± 0.21	88.89 ± 0.13	60.62 ± 0.22	77.39 ± 0.16	69.05 ± 0.22	87.79 ± 0.12	73.57 ± 0.23	88.66 ± 0.14
LCCRN [33]	75.72 ± 0.21	88.42 ± 0.13	63.08 ± 0.22	78.38 ± 0.15	72.92 ± 0.21	89.34 ± 0.11	74.28 ± 0.22	**89.39 ± 0.14**
Ours	**79.84 ± 0.45**	**90.68 ± 0.26**	**65.72 ± 0.50**	**80.80 ± 0.34**	**78.73 ± 0.42**	**90.89 ± 0.23**	**75.07 ± 0.51**	88.66 ± 0.31

4. Experiments

4.1. Datasets

In the experiments, we evaluate the effectiveness of our method on four fine-grained datasets, CUB-200-2011 [25], Stanford-Dogs [26], Stanford-Cars [27] and Flowers [28].

The CUB-200-2011 dataset consists of images of 200 bird species, with a total of 11,788 images. We randomly divide it to a training set with 100 classes, a validation set with 50 classes and a test set with 50 classes.

The Stanford-Dogs dataset includes images of 120 different dog breeds from around the world, with a total of 20,580 images. We randomly divide it to a training set with 60 classes, a validation set with 30 classes and a test set with 30 classes.

The Stanford-Cars dataset consists of images of 196 different car categories, including various years, brands, and models, with a total of 16,185 images. We randomly split it to a training set with 130 classes, a validation set with 17 classes and a test set with 49 classes.

The Flowers dataset contains 8189 images of 102 different flower categories. We randomly split it to a training set with 51 classes, a validation set with 26 classes and a test set with 25 classes.

4.2. Implementation details

In all experiments, the input images are resized to 84 × 84. Random resized cropping and random horizontal flipping are applied during training. In line with previous works on few-shot classification [6–8], we employ ResNet-12 and Conv-4 as the backbones. We train the model based on the PyTorch deep learning framework using NVIDIA RTX 3090. We conduct experiments using 5-way 1-shot ($N = 5, K = 1$) and 5-way 5-shot ($N = 5, K = 5$) settings, with $q = 15$ query images per class. During training, the loss is minimised using an SGD optimiser with a momentum of 0.9 and a weight decay of $5e{-}4$. The initial learning rate is set to 0.1. Following RENet, we train the network for 80 epochs in 1-shot settings and 60 epochs in 5-shot settings. In the testing phase, we perform few-shot classification tasks on 2000 randomly sampled episodes and report the average classification accuracies within the corresponding 95% confidence intervals. Our model is trained end-to-end without pretraining and required no fine-tuning during testing.

4.3. Comparison with the state-of-the-art methods

We compare the proposed SRML network with the following state-of-the-art (SOTA) methods: MatchNet [6], ProtoNet [7], RelationNet [8], DN4 [9], Baseline++ [29], DeepEMD [30], DSN [31], CAN [13], CTX [14], RENet [16], BSNet [10], FRN [32], TDM [11], FEAT [34], MixtFSL [35], VFD [36], AGPF [37] and LCCRN [33]. The classification accuracies and the corresponding 95% confidence intervals [38] are reported in Tables 1 and 2 for the fine-grained datasets under the Conv-4 and ResNet-12 backbones, respectively.

It is obvious that our proposed SRML dominates other state-of-the-art methods for all scenarios, except for 1-shot classification of Stanford-Dogs and 5-shot classification of Flowers. Specifically, we have the following observations. First, methods matching target objects in support and query sets, such as RENet and SRML, usually perform better than those without this relational information, such as MatchNet, ProtoNet and RelationNet, which demonstrates the importance of involving the relational information in classification. Second, SRML tends to outperform methods that only learn the relational information about the target objects, such as CTX and CAN, which suggests that utilising the structural information within images is equally important for classification. Lastly, SRML can achieve higher classification accuracies than RENet, which demonstrates that it is crucial to encourage mutual learning between the two types of information.

To formally verify the statistically significance of our results, we conduct one-sided paired t test with $H_0 : \mu_{\text{Ours}} \leq \mu_{\text{SOTA}}$ and H_1 :

Table 2
The 5-way few-shot classification accuracies on the CUB-200–2011, Stanford-Cars, Stanford-Dogs and Flowers datasets for the ResNet-12 backbone.

Methods	CUB-200–2011		Stanford-Dogs		Stanford-Cars		Flowers	
	1-shot	5-shot	1-shot	5-shot	1-shot	5-shot	1-shot	5-shot
MatchNet [6]	71.87 ± 0.85	85.08 ± 0.57	66.48 ± 0.88	79.57 ± 0.63	73.32 ± 0.93	87.61 ± 0.55	75.70 ± 0.88	87.61 ± 0.55
ProtoNet [7]	77.96 ± 0.47	89.27 ± 0.27	66.23 ± 0.49	81.60 ± 0.34	82.23 ± 0.42	92.11 ± 0.22	73.91 ± 0.52	86.24 ± 0.36
RelationNet [8]	63.94 ± 0.92	77.87 ± 0.64	47.35 ± 0.88	66.20 ± 0.74	69.67 ± 1.01	84.29 ± 0.68	69.51 ± 1.01	86.84 ± 0.56
CTX [14]	80.39 ± 0.20	91.01 ± 0.11	**73.22 ± 0.22**	85.90 ± 0.13	85.03 ± 0.19	92.63 ± 0.11	–	–
CAN [13]	77.42 ± 0.49	87.61 ± 0.30	49.14 ± 0.52	63.08 ± 0.43	25.64 ± 0.32	35.31 ± 0.25	61.36 ± 0.58	73.12 ± 0.49
FEAT [34]	73.27 ± 0.22	85.77 ± 0.14	–	–	–	–	–	–
DeepEMD [30]	75.65 ± 0.83	88.69 ± 0.50	67.59 ± 0.30	83.13 ± 0.20	73.30 ± 0.29	88.37 ± 0.17	70.00 ± 0.35	83.63 ± 0.26
RENet [16]	80.50 ± 0.44	91.11 ± 0.24	71.53 ± 0.48	85.92 ± 0.30	86.04 ± 0.39	94.43 ± 0.18	77.81 ± 0.46	89.45 ± 0.30
MixtFSL [35]	67.86 ± 0.94	82.18 ± 0.66	67.26 ± 0.90	82.05 ± 0.56	58.15 ± 0.87	80.54 ± 0.63	72.60 ± 0.91	86.52 ± 0.65
VFD [36]	79.12 ± 0.83	91.48 ± 0.39	63.65 ± 0.92	78.13 ± 0.62	77.52 ± 0.85	90.76 ± 0.46	76.20 ± 0.92	89.90 ± 0.53
AGPF [37]	78.73 ± 0.84	89.77 ± 0.47	72.34 ± 0.86	84.02 ± 0.57	85.34 ± 0.74	94.79 ± 0.35	–	–
Ours	**83.05 ± 0.43**	**92.74 ± 0.23**	72.97 ± 0.47	**86.01 ± 0.30**	**87.49 ± 0.36**	**95.34 ± 0.16**	**79.82 ± 0.47**	**91.97 ± 0.26**

Table 3
The results of the one-sided paired t-test to compare our method with SOTA methods. ✔ indicates $p < 0.05$.

Ours vs. *	CUB-200–2011		Stanford-Cars		Stanford-Dogs		Flowers	
	1-shot	5-shot	1-shot	5-shot	1-shot	5-shot	1-shot	5-shot
ProtoNet [7]	✔	✔	✔	✔	✔	✔	✔	✔
CAN [13]	✔	✔	✔	✔	✔	✔	✔	✔
RENet [16]	✔	✔	✔	✔	✔	✔	✔	✔
VFD [36]	✔	✔	✔	✔	✔	✔	✔	✔

Table 4
The ablation studies on the SFE and RFE branches and the BKE learning strategy.

	SFE	RFE	BKE	CUB-200–2011		Stanford-Cars		Stanford-Dogs		Flowers	
				1-shot	5-shot	1-shot	5-shot	1-shot	5-shot	1-shot	5-shot
(a)	✗	✗	✗	80.41	90.85	84.67	93.16	70.54	84.11	75.60	87.58
(b)	✔	✗	✗	81.50	92.04	85.84	94.73	72.31	85.30	78.66	91.14
(c)	✗	✔	✗	80.56	91.35	86.82	94.88	71.33	84.89	78.07	89.46
(d)	✔	✔	✗	82.55	92.45	86.97	95.05	72.49	85.60	79.06	91.37
SRML	✔	✔	✔	**83.05**	**92.74**	**87.49**	**95.34**	**72.97**	**86.01**	**79.82**	**91.97**

$\mu_{\text{Ours}} > \mu_{\text{SOTA}}$, where μ is the mean accuracy. We compare with the major competitors in Table 2. The results of these tests are reported in Table 3, with✔indicating $p < 0.05$, i.e. we reject H_0 with strong confidence. Clearly, our method is significantly better than ProtoNet, CAN, RENet and VFD for all datasets.

To sum up, SRML demonstrates superior classification performance for fine-grained datasets over the state-of-the-art methods.

4.4. Ablation studies

4.4.1. The impact of sfe, RFE and BKE

In this section, we conduct a series of ablation experiments to evaluate the impact of different components of the proposed method on its classification performance, and summarise the results in Table 4. It is clear that utilising all three components, the SFE and RFE branches and the BKE learning strategy, achieves the best classification accuracies on all four datasets. In addition, we also observe that scenario-(b) has higher classification accuracies than scenario-(c) in most cases, which indicates that simply exploiting the object relationships across images but ignoring the within-image structural patterns is not ideal for image classification. Moreover, scenario-(d) is only slightly better than scenario-(b), which demonstrates the value of involving the BKE learning for knowledge exchange between the two branches.

4.4.2. The impact of the number of epochs

To study the impact of the number of epochs, we depict the change of classification accuracy against the number of epochs in Fig. 5 on the CUB-200-2011 dataset. For both 1-shot and 5-shot settings, we can observe an upward trend of accuracy as the number epochs increases. In addition, SRML performs better than RENet for all number of epochs.

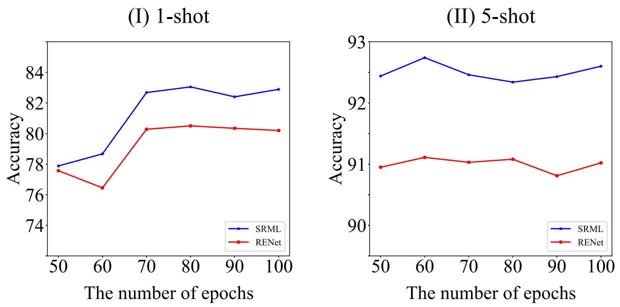

Fig. 5. The impact of the number of epochs on the classification accuracy of RENet and SRML for the CUB-200-2011 dataset.

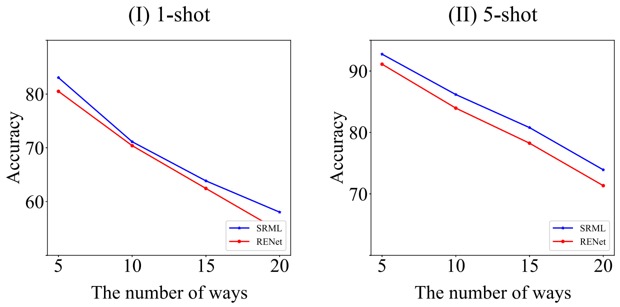

Fig. 6. The impact of the number of ways on the classification accuracy of RENet and SRML for the CUB-200-2011 dataset.

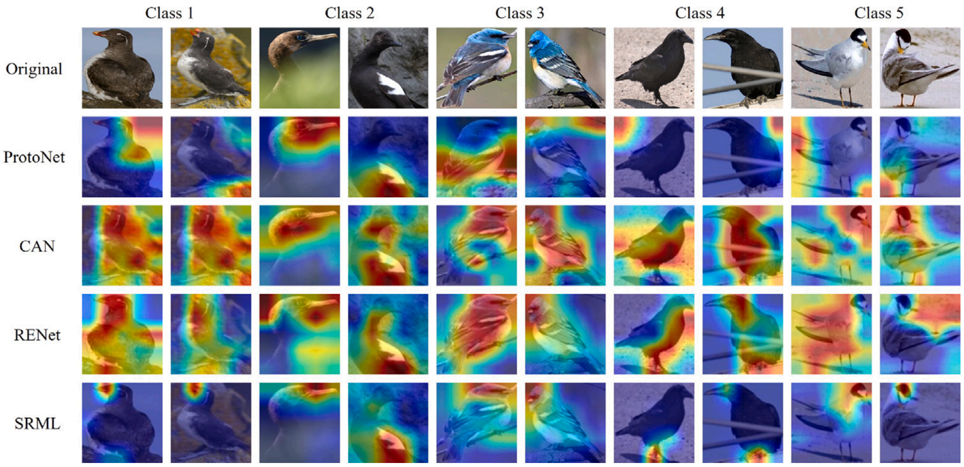

Fig. 7. Visualisations of the discriminative features captured by ProtoNet, CAN, RENet and SRML for five classes in the CUB-200-2011 dataset.

Table 5
Classification accuracies of CUB-200–2011 with different values of α, β and ω.

	α	β	ω	1-shot	5-shot
(1)	0.5	0.5	0.1	82.86	92.40
(2)	1.5	0.5	0.1	82.87	92.36
(3)	1.0	0.5	0.1	**83.05**	**92.74**
(4)	1.0	0.1	0.1	82.78	92.48
(5)	1.0	1.0	0.1	82.55	92.03
(6)	1.0	0.5	0.0	82.55	92.45
(7)	1.0	0.5	0.5	82.89	92.37

4.4.3. *The impact of the number of ways*

To explore the impact of the number of ways, in Fig. 6 we present the classification accuracy of different number of ways under the 1-shot and 5-shot settings on the CUB dataset. Overall, our method demonstrates superior classification performance compared to RENet over all number of ways.

4.4.4. *The impact of hyperparameters in the loss function*

We conduct a sensitivity analysis on three key weighting hyperparameters of the loss function: α, β and ω. Table 5 presents the test performance on the CUB-200-2011 dataset for various values of these hyperparameters. The best combination is $\alpha = 1$, $\beta = 0.5$ and $\omega = 0.1$. Notably, even the worst combination outperforms the SOTA methods in Table 2 in both 1-shot and 5-shot classification on CUB-200-2011.

4.5. *Discussion of limited performance of SRML on coarse-grained images*

We further evaluate the performance of SRML network on two coarse-grained datasets, *tiered*ImageNet [39] and *mini*ImageNet [6]. The *tiered*ImageNet dataset contains 34 super-categories and 608 object classes in total. The training set contains 20 super-categories with a total of 351 object classes. The validation set is consisting of 6 super-categories with 97 object classes. The test set consists of 8 super-categories with 160 object classes. The *mini*ImageNet dataset consists of 60,000 images with 100 object classes, and each class contains 600 images. The training, validation and test sets have 64, 16, and 20 classes, respectively.

We can observe from Table 6 that, SRML tends to perform better than RENet for higher number of shots, while competitive or worse than RENet for the 1-shot scenario. For the *mini*ImageNet dataset, the accuracy of SRML is only slightly worse than RENet but competitive for the 1-shot scenario, while for the *tiered*ImageNet dataset, the accuracy of SRML is over 1% lower than that of RENet. One potential explanation to this is that, *tiered*ImageNet is a more challenging task with substantially larger within-class variations due to the construction of super-categories. Therefore, when the support set only contains one image, our calculations of $\mathbf{s}^{\text{inter}}$ and $\mathbf{s}^{\text{intra}}$ are not reliable to evaluate the similarity between the query image and support classes. Imposing L^{BKE} based on these similarities in the training process for such dataset is not desirable.

4.6. *Qualitative analysis via visualisations*

4.6.1. *Visualisation of the discriminative features*

To intuitively verify the effectiveness of the proposed SRML network, we generate the CAM based feature visualisations [40] for five classes from the CUB-200-2011 dataset in Fig. 7. Clearly, SRML can provide the most precise concentrations on the discriminative areas to distinguish birds, e.g. beaks, heads and wings. CAN and RENet can also provide concentrations in these areas, but without precise focuses. Moreover, we note that RENet mis-identifies the target bird and focuses on the background in the last column of class 5. The drastic difference between the visualised discriminative features between SRML and RENet demonstrates the effectiveness of using the BKE learning strategy to allow interactions between the within-image structural information and the cross-image relational information.

In addition, in Fig. 8, we generate visualisations of multi-object images in the Stanford-Dogs dataset. Clearly, when nuisance objects are presented, our method accurately focuses on the target object. Furthermore, when two target objects are presented, our method successfully identifies both.

4.6.2. *Visualisation of the attention maps*

In Fig. 9, we further visualise the areas highlighted by the attention maps $\mathbf{A}_S$ and $\mathbf{A}_Q$ in Eq. (5). The attention maps with the largest weights are highlighted by blue for few test examples from the CUB-200-2011, Stanford-Cars, Stanford-Dogs and Flowers datasets. Obviously, the semantically similar areas are identified as important across query and support sets, e.g. the neck of the bird, the brand logo of the car, the nose of the dog and the pistil of the flower.

4.7. *Computational complexity*

Lastly, we discuss the computational complexity of SRML. The model parameters (Params.) and floating-point operations (FLOPs) are presented in Table 7. The FLOPs of our model is 5.92G, slightly higher than 3.56G of RENet. However, compared to the parameter count of RENet, there is no change because our BKE learning strategy does not introduce any additional parameters. With a slightly higher FLOPs, SRML can provide superior classification performance, especially for multi-object images.

Table 6
The 5-way few-shot classification accuracies on the *mini*ImageNet and *tiered*ImageNet datasets for the ResNet-12 backbone.

Methods	*miniImageNet*				*tieredImageNet*			
	1-shot	5-shot	7-shot	10-shot	1-shot	5-shot	7-shot	10-shot
ProtoNet [7]	62.99	79.32	81.23	82.93	68.23	84.03	85.05	86.34
CAN [13]	63.85	79.44	81.48	82.77	69.89	84.23	85.05	86.22
RENet [16]	**67.60**	82.58	84.21	86.09	**71.61**	**85.28**	85.73	86.97
Ours	67.37	**82.86**	**84.28**	**86.36**	70.40	84.92	**86.31**	**87.58**

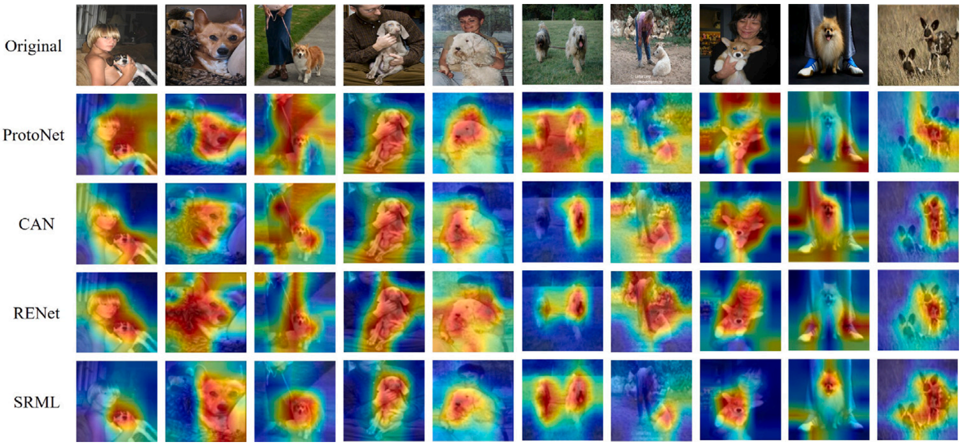

Fig. 8. Visualisations of the discriminative features captured by ProtoNet, CAN, RENet, and SRML for multi-object images in the Stanford-Dogs dataset.

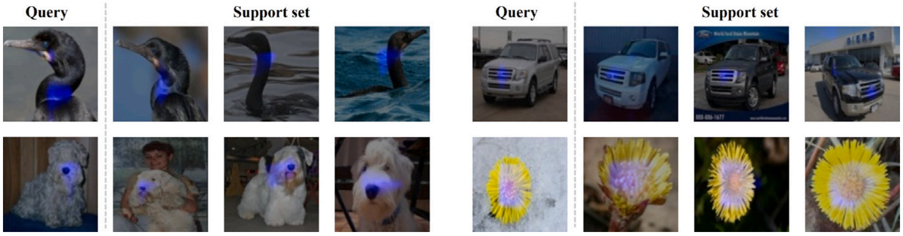

Fig. 9. Visualisations of the attention maps $\mathbf{A}_S$ and $\mathbf{A}_Q$ in Eq. (5). The attention maps with the largest weights are highlighted by blue for few test examples from CUB-200-2011 (top left), Stanford-Cars (top right), Stanford-Dogs (bottom left) and Flowers (bottom right).

Table 7
Model parameters (Param.) and FLOPs of SRML and SOTA methods.

Methods	FLOPs	Params.
CAN [13]	1.28G	8.04M
RENet [16]	3.56G	12.63M
FRN [32]	3.52G	12.42M
DeepEMD [30]	3.52G	12.42M
Ours	5.92G	12.63M

5. Conclusion

In this paper, we propose the SRML network for few-shot image classification. SRML excels in classifying a target object within a multi-object image. This capability is highly valuable in real-world scenarios, such as identifying a pedestrian in a busy street for autonomous driving or detecting a tumor in an MRI scan with multiple anatomical structures. The network architecture includes two parallel branches: SFE and RFE, effectively leveraging the structural features within-images and the relational features across-images. Additionally, we introduce the BKE learning strategy to facilitate knowledge exchange between the two branches, allowing the SFE branch to focus more on the target objects while the RFE branch better matches objects across images. Extensive experiments on benchmark datasets demonstrate the superior performance of the SRML network in few-shot image classification.

In the future, we aim to further enhance SRML to achieve high classification accuracy on challenging coarse-grained datasets when the number of shots is extremely limited. Furthermore, we will explore the potential of our method in real-world scenarios, particularly when noise, occlusion, or highly varied object scales are present.

CRediT authorship contribution statement

Xiaoxu Li: Supervision, Methodology. **Lang Wang:** Writing – original draft, Visualization, Validation, Methodology, Investigation, Formal analysis, Conceptualization. **Rui Zhu:** Writing – review & editing, Writing – original draft, Supervision. **Zhanyu Ma:** Supervision. **Jie Cao:** Supervision. **Jing-Hao Xue:** Writing – review & editing, Supervision.

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.

Acknowledgement

This work was partly supported by the National Nature Science Foundation of China (Grants 62176110, 62225601, U23B2052), Key Talent Program of Gansu Province, China under Grant 2025RCXM002, Science and Technology Program of Hebei, China under Grant SZX20 20034, Hong-Liu Distinguished Young Talents Foundation of Lanzhou University of Technology, Beijing Natural Science Foundation Project No. L242025, and the Royal Society under International Exchanges Award IEC\NSFC\201071.

Data availability

Data are publicly available.

References

[1] W. Zhang, Y. Zhao, Y. Gao, C. Sun, Re-abstraction and perturbing support pair network for few-shot fine-grained image classification, Pattern Recognit. 148 (2024) 110158.

[2] M. Dong, F. Li, Z. Li, X. Liu, PRSN: Prototype resynthesis network with cross-image semantic alignment for few-shot image classification, Pattern Recognit. 159 (2025) 111122.

[3] Z. Li, H. Guo, Y. Chen, C. Liu, Q. Du, Z. Fang, Y. Wang, Few-shot hyperspectral image classification with self-supervised learning, IEEE Trans. Geosci. Remote Sens. 61 (2023) 1–17.

[4] Z. Dai, J. Yi, L. Yan, Q. Xu, L. Hu, Q. Zhang, J. Li, G. Wang, Pfemed: Few-shot medical image classification using prior guided feature enhancement, Pattern Recognit. 134 (2023) 109108.

[5] F. Liu, S. Yang, D. Chen, H. Huang, J. Zhou, Few-shot classification guided by generalization error bound, Pattern Recognit. 145 (2024) 109904.

[6] O. Vinyals, C. Blundell, T. Lillicrap, D. Wierstra, et al., Matching networks for one shot learning, Adv. Neural Inf. Process. Syst. 29 (2016).

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

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

[9] 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: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2019, pp. 7260–7268.

[10] X. Li, J. Wu, Z. Sun, Z. Ma, J. Cao, J.-H. Xue, BSNet: Bi-similarity network for few-shot fine-grained image classification, IEEE Trans. Image Process. (2021).

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

[12] H. Song, B. Deng, M. Pound, E. Özcan, I. Triguero, A fusion spatial attention approach for few-shot learning, Inf. Fusion (2022).

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

[14] C. Doersch, A. Gupta, A. Zisserman, CrossTransformers: Spatially-aware few-shot transfer, 2020, ArXiv arXiv:2007.11498.

[15] E. Shechtman, M. Irani, Matching local self-similarities across images and videos, in: 2007 IEEE Conference on Computer Vision and Pattern Recognition, IEEE, 2007, pp. 1–8.

[16] D. Kang, H. Kwon, J. Min, M. Cho, Relational embedding for few-shot classification, in: Proceedings of the IEEE/CVF International Conference on Computer Vision, 2021, pp. 8822–8833.

[17] Y. Zhang, T. Xiang, T.M. Hospedales, H. Lu, Deep mutual learning, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2018, pp. 4320–4328.

[18] X. Li, S. Ding, J. Xie, X. Yang, Z. Ma, J.-H. Xue, CDN4: A cross-view deep nearest neighbor neural network for fine-grained few-shot classification, Pattern Recognit. 163 (2025) 111466.

[19] J. Lai, S. Yang, W. Wu, T. Wu, G. Jiang, X. Wang, J. Liu, B.-B. Gao, W. Zhang, Y. Xie, C. Wang, SpatialFormer: Semantic and target aware attentions for few-shot learning, 2023, Arxiv, arXiv:2303.09281.

[20] X. Chen, W. Wu, L. Ma, X. You, C. Gao, N. Sang, Y. Shao, Exploring sample relationship for few-shot classification, Pattern Recognit. 159 (2025) 111089.

[21] A. Afrasiyabi, H. Larochelle, J.-F. Lalonde, C. Gagné, Matching feature sets for few-shot image classification, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2022, pp. 9014–9024.

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

[23] K. Huang, X. Deng, J. Geng, W. Jiang, Self-attention and mutual-attention for few-shot hyperspectral image classification, in: 2021 IEEE International Geoscience and Remote Sensing Symposium IGARSS, IEEE, 2021, pp. 2230–2233.

[24] J. Rajasegaran, S. Khan, M. Hayat, F.S. Khan, M. Shah, Self-supervised knowledge distillation for few-shot learning, 2020, arXiv preprint arXiv:2006.09785.

[25] C. Wah, S. Branson, P. Welinder, P. Perona, S.J. Belongie, The Caltech-UCSD Birds-200–2011 Dataset, Tech. Rep. CNS-TR-2011-001, California Institute of Technology, 2011.

[26] A. Khosla, N. Jayadevaprakash, B. Yao, L. Fei-Fei, Novel dataset for fine-grained image categorization : Stanford dogs, in: ArXiv, 2012.

[27] J. Krause, M. Stark, J. Deng, L. Fei-Fei, 3D object representations for fine-grained categorization, 2013, pp. 554–561,

[28] M.-E. Nilsback, A. Zisserman, Automated flower classification over a large number of classes, 2008 Sixth Indian Conf. Comput. Vis. Graph. & Image Process. (2008) 722–729.

[29] W.-Y. Chen, Y.-C. Liu, Z. Kira, Y. Wang, J.-B. Huang, A closer look at few-shot classification, 2019, ArXiv, arXiv:1904.04232.

[30] C. Zhang, Y. Cai, G. Lin, C. Shen, DeepEMD: Few-shot image classification with differentiable earth mover's distance and structured classifiers, 2020 IEEE/ CVF Conf. Comput. Vis. Pattern Recognit. (2020) 12200–12210.

[31] C. Simon, P. Koniusz, R. Nock, M. Harandi, Adaptive subspaces for few-shot learning, in: 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2020, 2020, pp. 4135–4144.

[32] Z. Shen, Z. Liu, J. Qin, M. Savvides, K.-T. Cheng, Partial is better than all: revisiting fine-tuning strategy for few-shot learning, in: Proceedings of the AAAI Conference on Artificial Intelligence, Vol. 35, 2021, pp. 9594–9602.

[33] X. Li, Q. Song, J. Wu, R. Zhu, Z. Ma, J.-H. Xue, Locally-enriched cross-reconstruction for few-shot fine-grained image classification, IEEE Trans. Circuits Syst. Video Technol. (2023).

[34] H.-J. Ye, H. Hu, D.-C. Zhan, F. Sha, Few-shot learning via embedding adaptation with set-to-set functions, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2020, pp. 8808–8817.

[35] A. Afrasiyabi, J.-F. Lalonde, C. Gagné, Mixture-based feature space learning for few-shot image classification, in: Proceedings of the IEEE/CVF International Conference on Computer Vision, 2021, pp. 9041–9051.

[36] J. Xu, H. Le, M. Huang, S. Athar, D. Samaras, Variational feature disentangling for fine-grained few-shot classification, in: Proceedings of the IEEE/CVF International Conference on Computer Vision, 2021, pp. 8812–8821.

[37] H. Tang, C. Yuan, Z. Li, J. Tang, Learning attention-guided pyramidal features for few-shot fine-grained recognition, Pattern Recognit. 130 (2022) 108792.

[38] J.S. Milton, J.C. Arnold, Schaum's Outline of Introduction to Probability & Statistics: Principles & Applications for Engineering & the Computing Sciences, McGraw-Hill Higher Education, 1994.

[39] 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, 2018, ArXiv, arXiv:1803.00676.

[40] R.R. Selvaraju, M. Cogswell, A. Das, R. Vedantam, D. Parikh, D. Batra, Grad-CAM: Visual explanations from deep networks via gradient-based localization, in: Proceedings of the IEEE International Conference on Computer Vision, 2017, pp. 618–626.