Reservoir Computing Based Attention Network for Few-shot Image Classification

Anonymous submission

Abstract

Few-shot image classification aims to classify unlabeled images from unseen classes using a limited number of labeled images. However, many existing approaches can not learn the features well and suffer from overfitting due to the scarcity of labeled images. Consequently, they face challenges in generalizing well to new tasks, especially in cross-domain scenarios. To tackle this problem, we propose a novel approach called *Reservoir Computing Based Attention Network (RCA)* for few-shot image classification. Firstly, by taking advantage of the high expressivity and natural overfitting avoidance of Reservoir Computing (RC), a novel model RCA is proposed to integrate the extracted features and make them more discriminative. Secondly, to further alleviate the overfitting, we employ a slightly modified two-stage training strategy that improves classification performance in the target domain. Extensive experiments with various backbones are conducted on Cifar10/100, Mini-Imagenet, Tiered-Imagenet, Cifar-FS, FC100, and CUB-200-2011, which indicate that RCA has great performance and generalization ability, outperforming the state-of-the-art methods by $1\% \sim 5\%$.

Introduction

Recently, deep learning methods have achieved great success in the field of computer vision (Wang et al. 2021a; Touvron et al. 2021). However, they are commonly data-hungry, requiring thousands of labeled samples, which are expensive to collect and annotate. When facing limited labeled samples, models usually suffer overfitting, despite using regularization, normalization, and data augmentation (Hui et al. 2019). It brings great challenges to the generalization ability of deep learning methods, leading to the development of few-shot learning (FSL) (Hui et al. 2019; Antoniou, Edwards, and Storkey 2019).

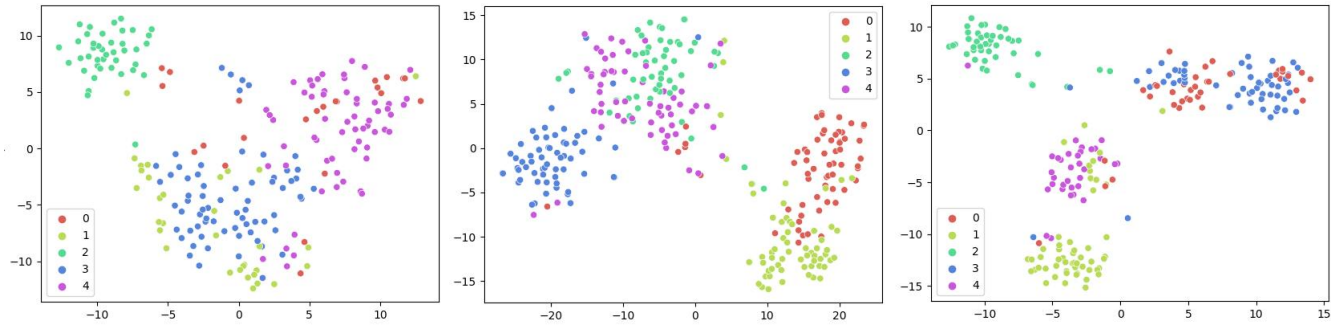
FSL aims to recognize unlabeled samples into unseen classes with very few labeled samples. It is vital for enhancing the performance of FSL to learn a good feature representation and many methods are making efforts on it, such as making the feature distribution with a high inter-class variance and a low intra-class variance (Liang et al. 2021) as shown in Fig 1, or designing methods to increase the discrimination (Bi, Xue, and Zhang 2021), etc.

However, some models and training strategies are still suffering weaknesses as follows and can be further optimized. In the aspect of models, there are many methods, combining the convolutional operator and attention mechanism, to improve the representation ability of models for FSL (Hui et al. 2019). However, these methods often suffer overfitting (Hou et al. 2019), indicating that the discrimination of the extracted features can be improved. In the aspect of training strategy, fine-tuning methods have achieved great success due to their simplicity and prominent performance, but their generalization ability can be further improved (Triantafillou et al. 2021; Li, Liu, and Bilen 2022), especially when facing cross-domain scenes (Chen et al. 2019a).

To tackle these issues, this study proposes a novel model and a slightly modified fine-tuning method. Firstly, to avoid training a large number of parameters while maintaining the representation ability of the data, we suggest the **Reservoir Computing Based Attention Network (RCA)**, depicted in Fig. 2. RCA capitalizes on the high expressive power and inherent overfitting avoidance of Reservoir Computing (RC) by harnessing the complex internal interactions' critical dynamics, devoid of any parameter training within the reservoir (Jaeger 2001; Maass, Natschläger, and Markram 2002). Consequently, RCA can extract and represent input features more effectively in a lower-dimensional space than traditional attention mechanisms employing linear transformations or convolutional operations, as illustrated in Fig. 1. Secondly, to further mitigate overfitting and enhance performance, we partition the original FSL training set into new training and test sets by **identifying appropriate division ratios** in the initial stage, followed by model training on these new sets. Subsequently, the model is fine-tuned in the second stage for few-shot image classification tasks on the original FSL training, validation, and test sets using the N-way K-shot method.

Extensive experiments show that the proposed method can favorably achieve competitive performance. Meanwhile, the cross-domain scene from Mini-Imagenet to CUB-200-2011 achieves consistent improvement, demonstrating the good generalization ability of the proposed RCA. In summary, our contributions are two-fold:

- We propose a novel method, called RCA, which is the pioneer attempt that uses RC on the challenging FSL task. RCA enhances the extracted features and makes the model learn a large inter-class distribution and a compact intra-class distribution of features, which has great generalization ability and performs well on cross-domain scenes.



(a). Conv. (b). Linear. (c). RCA.

Figure 1: An example of the feature distribution enhanced by using different attention mechanisms on Mini-Imagenet with 5-way 1-shot and 45 query images. The backbone is ResNet-12. (a) is generated by the attention mechanism with a convolutional operator. (b) is generated by the attention mechanism with a traditional linear transformation. (c) is generated by the proposed RCA. The proposed RCA makes the feature more discriminative with compact intra-class distributions and high inter-class variances for better classification than (a) and (b).

Meanwhile, we adopt a two-stage training strategy. In the first stage, we divide the training set of the original FSL dataset into a new training set and a new test set by studying the appropriate division ratios of different datasets for sufficient learning source domain knowledge. Then, in the second stage, we fine-tune the model for FSL, especially replacing the linear classifier with the cosine classifier. The training strategy enhances the feature extraction ability to alleviate overfitting.

- We conduct extensive experiments and the performance of our method outperforms a variety of state-of-the-art FSL methods on public few-shot image classification datasets (e.g., Cifar-FS, FC100, Mini-Imagenet, and Tiered-Imagenet) and cross-domain scenes by about $1\% \sim 5\%$. Extensive ablation experiments on different ways of generating attention mechanisms have also verified the effectiveness of our proposed method based on the topology of the reservoir we designed.

Related Work

Our method is an RC-based attention mechanism for FSL. Therefore, we first introduce the FSL as a whole. Then, we introduce the attention-based FSL, which is related to our method. Finally, we describe the application of RC, especially applied in computer vision tasks.

Few-shot Learning

Most FSL methods can be summarized as follows: **Data augmentation-based methods** (Royle, Dorazio, and Link 2007) usually learn a generator from available images in an instance level. (Boney and Ilin 2018). Enhancing the feature space (Chen et al. 2019b; Ren et al. 2019) is also useful, because the key to few-shot learning is to obtain a feature extractor with a great generalization ability. **Fine-tuning-based methods** (Qiao et al. 2018; Wang et al. 2020) usually train a model based on a large-scale dataset (source domain) to learn representative features, so that classes in the target domain that only contain a few samples can be easily classified based on source classes. However, such methods often suffer overfitting, especially for the cross-domain problem (Chen et al. 2019a; Yu et al. 2020). **Metric-learning-based methods** aim to learn an embedding or metric space and compare sample-to-sample or sample-to-class embedding distances to measure the similarity among samples (Vinyals et al. 2016; Snell, Swersky, and Zemel 2017). **Memory-augmented-based methods** add additional memory modules, such as RNN or LSTM, to remember the characteristic information of the support set and the query set is required to match with the previously obtained knowledge (Santoro et al. 2016; Zhu and Yang 2020). **Optimization-based methods** (Ravi and Larochelle 2017; Rusu et al. 2018) target at learning an optimizer or finding the optimal model parameters, to solve the overfitting due to limited labeled examples (Elsken et al. 2020). **Meta-learning-based methods** aim to train a model over a batch of tasks (episodes) to get the meta-knowledge and fast generalize well to new tasks. Meta-learning-based methods usually are in combination with other methods mentioned above (Ravi and Larochelle 2017; Santoro et al. 2016).

In this work, we use a slightly different fine-tuning method and the meta-learning way to conduct the few-shot image classification tasks, which alleviates overfitting and performs well.

Attention Mechanism in Few-shot Learning

CV attention methods emulate the selective attention mechanism observed in the human visual system, wherein such a focused process facilitates the neural system's capacity to analyze and comprehend intricate scenes with enhanced efficiency.

In FSL, attention mechanisms are usually used to integrate feature information for enhancing the representation. For instance, the multi-attention network precisely captures the representative parts by the attention maps of visual features. MatchingNet (Vinyals et al. 2016) and Cross-Attention (Hou et al. 2019) are proposed to capture the semantic dependency between the support set and the query set

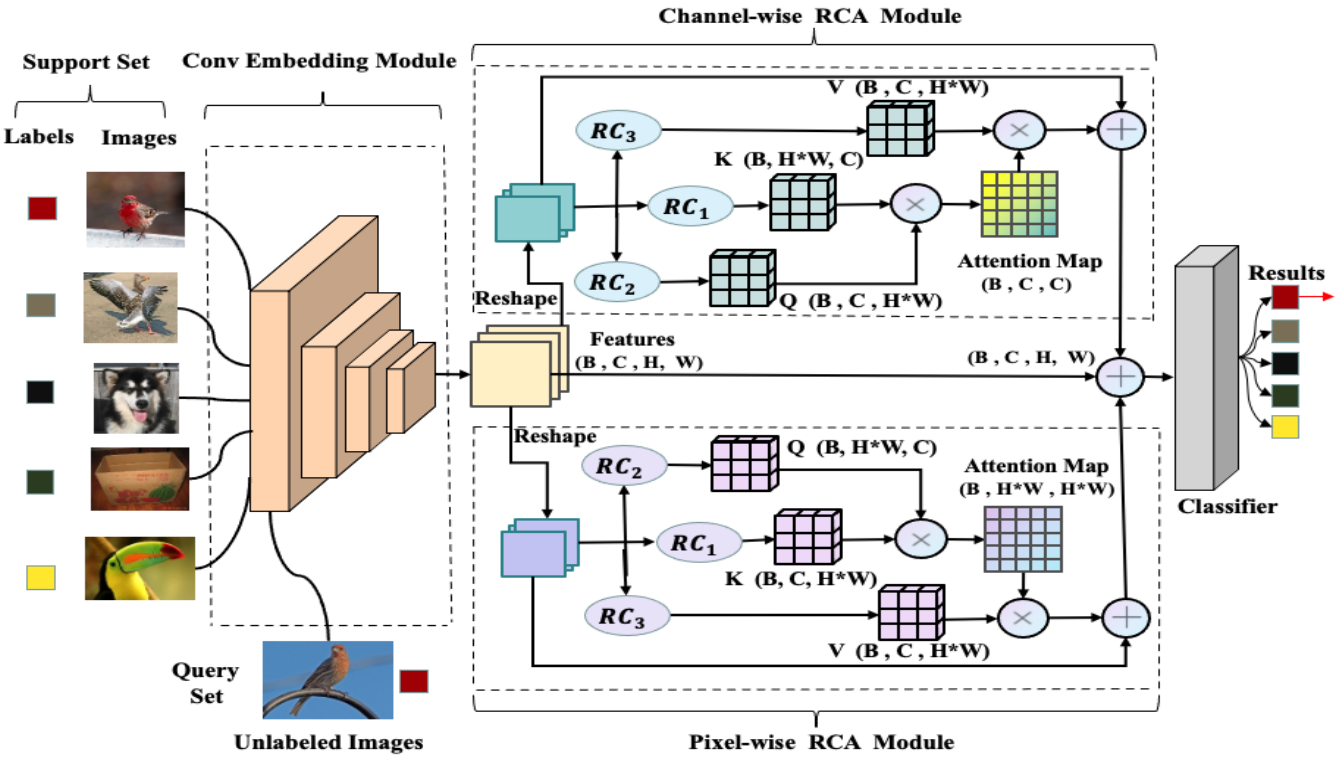


Figure 2: **Reservoir Computing Based Attention Network (RCA)** for a 5-way 1-shot image recognition task. Given a support set consisting of images per class and a query set with unlabeled images, we encode images into a rich feature map using the Cov Embedding Module (ResNet-12/18). Afterward, we use a Channel-wise RCA Module and a Pixel-wise RCA module to enhance the features to improve the performance. The $\otimes$ denotes the matrix multiplication. The $\oplus$ denotes the element-wise sum.

for better classification. However, the generalization ability of such attention mechanisms is usually not optimal, which can be further improved by generating a stronger attention mechanism.

In this paper, to extract more representative features, the attention mechanism is generated by a simple and efficient novel brain-like computational mechanism RC (Tanaka et al. 2019), rather than by linear transformations or convolutional operators.

Reservoir Computing

Reservoir Computing (RC), mainly composed of Echo State Networks (ESNs) (Jaeger 2001) and Liquid State Machines (LSMs) (Maass, Natschläger, and Markram 2002), has been successively applied on speech recognition (Skowronski and Harris 2006), time series prediction (Aswolinskiy, Reinhart, and Steil 2018), etc.

Recently, some works begin to study the behavior of such networks on computer vision (Jalalvand et al. 2018; Koprinkova-Hristova 2021). For instance, Shen et al. (Shen et al. 2021a) randomly initialized some of the layers in transformers without updates but obtained impressive performance. However, such studies remain restricted because they simply treat RC as an auxiliary tool but not the core for image tasks. ViR (Wei et al. 2021) model used pure RC for image classification, but not achieved great performance.

Reservoir Computing based Attention Network

The whole network mainly consists of the Conv Embedding Module, the Reservoir Computing Based Attention (RCA) Module, and the Classifier Module. The Conv Embedding Module extracts features of the input image, and then the RCA Module enhances the extracted features, and finally, the outputs of the RCA Module are sent to the Classifier Module for classification.

Conv Embedding Module

The Conv Embedding Module consists of a convolutional network (e.g., ResNet-12/18) as a feature extractor $\mathbf{F}(\cdot|\theta)$ (learnable parameters θ) to embed a feature map $z = \mathbf{F}(x|\theta) \in \mathbb{R}^{(C,H,W)}$ from an input image x, where C is the number of channels, while H and W are the length and the width of the extracted feature map, respectively.

Reservoir Computing Based Attention (RCA)

The RCA Module employs the self-attention mechanism to enhance the extracted features, producing the output **Out=RCA**$(z|\varphi)$, where z is the input feature from the Conv Embedding Module and φ is the learnable parameter. **Out** and z are both $\in \mathbb{R}^{(C,H,W)}$. According to the self-attention

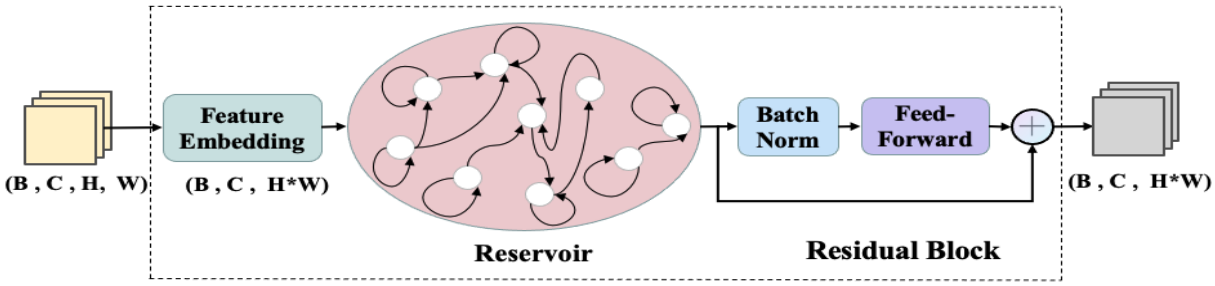


Figure 3: The overview of the **ViR Core**. The extracted features are reshaped by an embedding layer and then sent to the reservoir. The reservoir encodes the input with its high expressivity to make the attention mechanism alleviate the overfitting. Then a residual block including a batch normalization operation and a feed-forward operation operates the output of the reservoir.

mechanism, we change a generic non-local in the deep neural network (Wang et al. 2018) as:

$$\mathbf{Out}_i = \frac{1}{\mathcal{C}(z)} \sum_{\forall j} H(A_i, B_j)\mathbf{R}(z_j) + z_i \tag{1}$$

where A and B are two new feature maps from z generated by RC shown in Fig. 3; i is the position index of the output **Out**, whose response needs to be computed, while j indexes all the possible positions; $H(\cdot,\cdot)$ denotes a pairwise function, measuring the relationship between A_i and all the B_j; $\mathbf{R}(\cdot)$ denotes the transformation of z by RC, and $\mathcal{C}(z)$ is a normalization factor. For the pairwise function $H(\cdot,\cdot)$, the Gaussian function is a natural choice:

$$H(A_i, B_j) = e^{A_i^T B_j} \tag{2}$$

Then, we set:

$$\mathcal{C}(z) = \sum_{\forall j} H(A_i, B_j) \tag{3}$$

leading to:

$$\mathbf{Out}_i = \sum_{\forall j} \frac{e^{A_i^T B_j}}{\sum_{\forall j} e^{A_i^T B_j}} \mathbf{R}(z_j) + z_i \tag{4}$$

where, for a given i, $\frac{e^{A_i^T B_j}}{\sum_{\forall j} e^{A_i^T B_j}}$ becomes the softmax computation along the dimension j. Hence, we get the expression of the self-attention mechanism:

$$\mathbf{Out} = \mathbf{RCA}(z|\varphi) = \text{softmax}(A, B)\mathbf{R}(z) + z \tag{5}$$

Reservoir Computing (RC). As shown in Fig. 3, the RC module consists of a trainable linear feature embedding layer **E**, a reservoir with a nearly full-connected topology (detailed in Appendix C), and a residual block including a batch normalization operation (**BN**) and a feed-forward layer (**FF**). The RC module is used to generate the new feature maps, different from the traditional generation methods like a linear transformation or a convolutional operator. We first reshape the embedded features (the input of the RC module) $z \in \mathbb{R}^{(C,H,W)}$ as $z \in \mathbb{R}^{(C,D)}$. Then we split z along the dimension C and get C inputs $z_t \in \mathbb{R}^{(1,D)}$, where $D = H \times W$ represents the output dimension of the embedding layer **E** and t indicates the t-th input of RC.

The state of the M reservoir units updates according to the following equation:

$$\mathbf{s}_{t+1} = f(\mathbf{z}_{t+1}\mathbf{W}_{in} + \mathbf{s}_t\mathbf{W}_{res}), \quad t = 0, 1, ..., C-1 \tag{6}$$

where $\mathbf{z}_{t+1} = (\mathbf{z}^1_{t+1}, ..., \mathbf{z}^K_{t+1})$ means the activation of K input units for the $(t+1)$-th input; $\mathbf{s}_{t+1}$ means the state of the reservoir units for the $(t+1)$-th input; f is the activation function. All the elements of $\mathbf{s}(0)$ are 0.

$\mathbf{y}_{t+1}$ represents the output of the $(t+1)$-th input processed by the reservoir as follows:

$$\mathbf{y}_{t+1} = [\mathbf{z}_{t+1}; \mathbf{s}_{t+1}; \mathbf{y}_t; \mathbf{z}^2_{t+1}; \mathbf{s}^2_{t+1}; \mathbf{y}^2_t]\mathbf{W}_{out} \tag{7}$$

where ';' is a concatenation operation.

Topology of the Reservoir. Topology is one of the key parameters of a reservoir, which influences its stability and dynamics. Therefore, we design a high-performance reservoir consisting of M units with a nearly full-connected topology. We first generate $M \times M$ matrix $\mathbf{W}_{res}$ with every element set as r_1, and then select elements $\mathbf{W}_{1,M}$ and $\mathbf{W}_{q+1,q}$ set as r_2 with $q = 1, ..., M-1$, and select elements $\mathbf{W}_{k*l+1,(k+1)*l+1}$ and $\mathbf{W}_{(k+1)*l+1,k*l+1}$ set as r_3, with the integer $k = 0, ..., (M-1)/l - 1$ and the jump size $l = 1, ..., M-1$, where the jump means a shortcut connection between two nonadjacent neurons in a reservoir. Finally, 5% elements of $\mathbf{W}_{res}$, with the symmetrical elements are randomly selected and set to 0 indicating no connection between neurons. Hence, most elements in the matrix $\mathbf{W}_{res}$ are set as r_1, with relatively few elements assigned as r_2, r_3 or 0.

For the randomness, there exists a random matrix with the same size as $\mathbf{W}_{res}$, the elements e in which are randomly generated in (0,1]. If $e < 0.5$ (the threshold we set), then the sign of the weight in $\mathbf{W}_{res}$, corresponding to the same position as e, will be $-$, otherwise $+$ (According to cross-validation experiments, we choose 5% elements of $\mathbf{W}_{res}$ and set the threshold as 0.5). More detailed information and other topologies are shown in Appendix C.

As shown in Eq. 7, the essence of the network is to approximate output weights $\mathbf{W}_{out}$ through training samples, to obtain the predictive ability of certain tasks. In our work, we use a learnable linear layer (**LL**) to approximate $\mathbf{W}_{out}$, and then Eq. 7 becomes:

$$\mathbf{y}_{t+1} = [\mathbf{z}_{t+1}; \mathbf{s}_{t+1}; \mathbf{y}_t; \mathbf{z}^2_{t+1}; \mathbf{s}^2_{t+1}; \mathbf{y}^2_t]\mathbf{LL} \tag{8}$$

The residual block is applied to $\mathbf{y}_{t+1}$ and the output of the whole RC module $\mathbf{y}_{RC}$ (not the output of the reservoir) can be shown as:

$$\mathbf{y}_{RC} = \mathbf{FF}(\mathbf{BN}(\mathbf{y}_{t+1})) + \mathbf{y}_{t+1} = \mathbf{R}(\cdot) \tag{9}$$

where we set $\mathbf{R}(\cdot)$ as the reservoir computing operation.

Channel-wise RCA and Pixel-wise RCA. We use the channel-wise RCA and the pixel-wise RCA to emphasize the important channel and position of the embedded feature z similar in (Fu et al. 2019), as shown in Fig. 2. $A = \mathbf{R}_1(z)$ and $B = \mathbf{R}_2(z)$, where $\mathbf{R}_1(\cdot)$ and $\mathbf{R}_2(\cdot)$ represent two different reservoirs, $z \in \mathbb{R}^{(C,H,W)}$ and $A, B \in \mathbb{R}^{(C,H\cdot W)}$. For pixel-wise attention, we perform a matrix multiplication between the transpose of A and B, and get the pixel attention map $\mathbf{attn}_p$:

$$\mathbf{attn}_p = \beta_p(\text{softmax}(A^T B))\mathbf{R}(z) + z, \mathbf{attn}_p \in \mathbb{R}^{(C,H,W)} \tag{10}$$

For channel-wise attention, we perform a matrix multiplication between A and the transpose of B to get the channel attention map $\mathbf{attn}_c$:

$$\mathbf{attn}_c = \beta_c(\text{softmax}(AB^T))\mathbf{R}(z) + z, \mathbf{attn}_c \in \mathbb{R}^{(C,H,W)} \tag{11}$$

Hence, the final output of the RCA is given as:

$$\mathbf{Out} = \mathbf{RCA}(z|\varphi) = \beta_1 \mathbf{attn}_p + \beta_2 \mathbf{attn}_c + z \tag{12}$$

Here, β_1, β_2, β_p and β_c are different learnable scalar factors. Finally, the enhanced features are now fed to the classifier, and we can obtain the final classification results. The Classifier Module is detailed in Appendix E.

Training Strategy

For FSL, a dataset is commonly divided into three disjointed sets: the training set (base classes) $\mathbf{D}_{base}$, the validation set $\mathbf{D}_{valid}$ and the test set (novel classes) $\mathbf{D}_{novel}$. In meta-learning, a model is trained to learn some prior or shared knowledge from a batch of tasks generated from $\mathbf{D}_{base}$ and then tested on tasks from $\mathbf{D}_{novel}$. We use the **N**-way **K**-shot method to generate lots of independent tasks. For each task $\mathcal{T}$, we randomly sample N classes from $\mathbf{D}_{base}$ and randomly sample K images for each class of these N classes as the support set. Meanwhile, we select 15 images for each class of the aforementioned N classes as the query set.

1st stage: pre-training on $\mathbf{D}_{base}$. We divide $\mathbf{D}_{base}$ of FSL datasets into a new training set and a new test set (not the $\mathbf{D}_{novel}$) and train our model in the traditional image classification procedure with a linear classifier. We search the best division ratios as shown in Tab. 1 for learning sufficient knowledge, which outperforms others by about 2%. **2nd stage**. We fine-tune the pre-trained model and replace the linear classifier with the cosine classifier. Then, we train and test our model with lots of meta-t.

Experiments

We conduct experiments on the following widely used few-shot benchmarks: Cifar-FS, FC100, Mini-Imagenet, Tiered-Imagenet, and CUB-200-2011. Core codes are available in the Supplementary Materials.

Datasets

Cifar-FS and FC100. Cifar-FS (Bertinetto et al. 2018) and FC100 (Oreshkin, Rodríguez López, and Lacoste 2018) are divided from Cifar-100, which consists of $60,000$ images in 100 categories. The Cifar-FS is divided into 64, 16, and 20 for training, validation, and test, respectively. The FC100 uses a split similar to Tiered-Imagenet (Ren et al. 2018), where training, validation, and test splits contain 60, 20, and 20 classes. **Mini-Imagenet and Tiered-Imagenet**. Mini-Imagenet (Ravi and Larochelle 2017) is the most popular benchmark selected from ImageNet (Krizhevsky, Sutskever, and Hinton 2012), which is comprised of 64 training classes, 16 validation classes, and 20 test classes of images. Each class has 600 images. Tiered-Imagenet (Ren et al. 2018) is also a subset randomly sampled from ImageNet, which consists of 779165 images in 608 categories. All 608 categories are grouped into 34 broader categories. We use $20/6/8$ broader categories (351 classes/97 classes/160 classes) for training, validation, and test respectively. **CUB-200-2011**. CUB-200-2011 (Wah et al. 2011) consists of 11788 images of 84×84 from 200 classes. We randomly divide it into 100 training classes, 50 validation classes, and 50 test classes. Network architecture, training details and experimental environments are shown in Appendix A.

Results and Analysis

Standard few-shot tasks. During the test stage, we evaluate our method with 100 epochs, where in each epoch the accuracy is the mean accuracy of 1000 randomly sampled tasks. On all the datasets, we achieve better performance than the state-of-the-art methods.

Results on ImageNet derivatives. We report the results in Tab. 2 on Mini-Imagenet and Tiered-Imagenet. On Mini-Imagenet, our method with the ResNet-12 backbone already outperforms the state-of-the-art SSR (Shen et al. 2021b) and outperforms the state-of-the-art LR+DC (Yang, Liu, and Xu 2020) with the ResNet-18 backbone by about 2% on both the 5-way 1-shot and 5-way 5-shot tasks. On Tiered-Imagenet, our method outperforms all previous works by at least $1\% \sim 3\%$ on all tasks.

Results on Cifar derivatives. Similar to experiments on ImageNet derivatives, we evaluate our method with 200 epochs, in each epoch, the accuracy is the mean accuracy of 1000 randomly sampled tasks. Table 3 summarizes the results, which shows that our model with the ResNet-12 backbone outperforms the state-of-the-art methods such as HCTransformer (He et al. 2022), ICI v2 (Wang et al. 2021b), and DPGN (Yang et al. 2020) on Cifar-FS dataset. For the ResNet-18 backbone, our method outperforms the state-of-the-art methods aforementioned by at least 3% on Cifar-FS. On FC100, our method also outperforms the state-of-the-art methods In-Eq (Rizve et al. 2021) and HCTransformer by $1\% \sim 4\%$.

Cross-domain results. To fully demonstrate the remarkable generalization, we train our model with the ResNet-18 backbone on base classes of Mini-Imagenet, while the testing stage is performed on CUB few-shot tasks, with 50 testing classes. Table 4 reports the results. In this cross-domain

Table 1: Division ratios of different datasets (training: test), tasking 5-way 1-shot accuracy with ResNet-18 as an example(%). The former is the ratio, the latter is the accuracy.

	Mini-Imagenet	**Tiered-Imagenet**	**Cifar-FS**	**FC100**
Conventional Setting	0.70 — 71.14 ± 0.31	0.70 — 79.21 ± 0.47	0.70 — 79.25 ± 0.44	0.70 — 49.67 ± 0.62
Ours	0.73 — **75.21 ± 0.42**2	0.78 — **82.31 ± 0.21**	0.67 — **81.93 ± 0.43**	0.82 — **52.30 ± 0.45**

Table 2: **The comparison with the state-of-the-arts** on Mini-Imagenet and Tiered-Imagenet. Average few-shot classification accuracies (%) with 95% confidence intervals with different backbones.

Method	**Backbone**	**Mini-Imagenet**		**Tiered-Imagenet**	
		5-way 1-shot	5-way 5-shot	5-way 1-shot	5-way 5-shot
LLDG (Wang et al. 2021c)	Conv4-64	56.32 ± 0.28	72.64 ± 0.26	58.43 ± 0.38	76.17 ± 0.34
MELR (Fei et al. 2020)	Conv4-64	55.35 ± 0.43	72.27 ± 0.35	56.38 ± 0.48	73.22 ± 0.41
TPN (Wang et al. 2021b)	Conv4-64	53.75	69.43	57.53	72.85
Capsule Net (Wu et al. 2020)	Capsule Net.	66.43 ± 0.26	82.13 ± 0.21	-	-
Meta DeepBDC (Xie et al. 2022)	ResNet-12	67.34 ± 0.43	84.46 ± 0.28	72.34 ± 0.49	87.31 ± 0.32
DMF (Xu et al. 2021)	ResNet-12	67.76 ± 0.46	82.71 ± 0.31	71.89 ± 0.52	85.96 ± 0.35
SSR (Shen et al. 2021b)	ResNet-12	68.10 ± 0.60	76.90 ± 0.40	81.20 ± 0.60	85.70 ± 0.40
Meta-Baseline (Qin et al. 2022)	ResNet-12	64.89 ± 0.23	79.95 ± 0.17	68.62 ± 0.27	83.74 ± 0.18
IEPT (Zhang et al. 2020)	ResNet-12	67.05 ± 0.44	82.90 ± 0.30	72.24 ± 0.50	86.73 ± 0.34
DPGN (Yang et al. 2020)	ResNet-12	67.77 ± 0.32	84.60 ± 0.43	72.45 ± 0.51	87.24 ± 0.39
RCA(ours)	ResNet-12	**68.93 ± 0.44**	**84.84 ± 0.69**	**82.84 ± 0.69**	**90.32 ± 0.13**
MetaQDA (Zhang et al. 2021b)	WRN-28-10	67.83 ± 0.64	84.28 ± 0.69	74.33 ± 0.65	89.56 ± 0.79
LR+DC (Yang, Liu, and Xu 2020)	WRN-28-10	68.57 ± 0.55	82.88 ± 0.42	78.19 ± 0.25	89.90 ± 0.41
AWGIM (Qin et al. 2022)	WRN-28-10	64.16 ± 0.44	80.64 ± 0.32	67.69 ± 0.11	82.82 ± 0.13
MetaQDA (Zhang et al. 2021b)	ResNet-18	65.12 ± 0.66	80.98 ± 0.75	69.97 ± 0.52	85.51 ± 0.58
AFHN (Zhang et al. 2021b)	ResNet-18	62.38 ± 0.72	78.16 ± 0.56	-	-
FEAT (Ye et al. 2020)	ResNet-18	66.78	82.05	70.80 ± 0.23	84.79 ± 0.16
CTM (Li et al. 2019)	ResNet-18	64.12 ± 0.82	80.51 ± 0.13	68.41 ± 0.39	84.28 ± 1.73
S2M2 (Zhang et al. 2021b)	ResNet-18	64.06 ± 0.18	80.58 ± 0.12	-	-
PRRO (Hong et al. 2021)	ResNet-18	-	76.00 ± 0.60	-	78.90 ± 0.70
SimpleShot (Zhang et al. 2021b)	ResNet-18	63.10 ± 0.20	79.92 ± 0.14	69.68 ± 0.22	84.56 ± 0.16
RCA(ours)	ResNet-18	**75.21 ± 0.42**	**88.59 ± 0.59**	**82.31 ± 0.21**	**90.98 ± 0.57**

setting, the RCA outperforms complex meta-learning methods by substantial margins (at least 1%), shows great generalization ability, and alleviates overfitting. One possible explanation is that RC expresses complex features on highly curved manifolds into flattened manifolds in hidden space for learning (Bahri et al. 2020), so the highlight function of the attention mechanism is strengthened.

Comparisons with Other Attention Methods

Comparisons with other few-shot learning baselines which exploited attention mechanisms are conducted. As we can see from Appendix F, our performance is better than others given the same backbone by $1\% \sim 4\%$.

Ablation Studies

Effects of different methods to generate attention mechanisms. We conduct ablation experiments on different ways to generate attention mechanisms and evaluate them on the Mini-Imagenet dataset with the ResNet-18 backbone. As shown in Tab. 5, firstly, models with an attention mechanism (rows 3-5 in Tab. 5) outperform the model without using an attention method (the second row in Tab. 5). Secondly, our method outperforms the traditional linear transformation method and the convolutional operator method by about 4%. Furthermore, we conduct experiments on Cifar10 and Cifar100 detailed in Appendix B, and get the same conclusion.

Effects of using the same reservoir. Results of using the same reservoir to generate the RCA are shown in Appendix D, which indicates that using different reservoirs performs better than using the same reservoirs in the RCA since different reservoirs make the network more dynamic.

Effects of different topologies of RC. The topology affects the dynamics (the expressivity ability) of the reservoir. Hence, we conduct experiments on different topologies to find the one suitable for few-shot tasks on Mini-Imagenet and Cifar-FS. Appendix G shows that our nearly full-connected topology outperforms the others by about 2%.

Table 3: **The comparison with the state-of-the-arts** on Cifar-FS and FC100. Average few-shot classification accuracies (%) with 95% confidence intervals with different backbones.

Method	Backbone	Cifar-FS		FC100	
		5-way 1-shot	5-way 5-shot	5-way 1-shot	5-way 5-shot
HCTransformers (He et al. 2022)	ViT-S	78.89 ± 0.18	87.73 ± 0.11	48.27 ± 0.15	61.49 ± 0.15
SIB (Hu et al. 2020)	WRN-28-10	80.00 ± 0.60	85.30 ± 0.40	-	-
MetaQAD(Zhang et al. 2021b)	WRN-28-10	75.83 ± 0.88	88.79 ± 0.75	-	-
ConstellationNets (He et al. 2022)	ResNet-12	75.40 ± 0.20	86.80 ± 0.20	43.80 ± 0.20	59.70 ± 0.20
In-Eq (Rizve et al. 2021)	ResNet-12	77.87 ± 0.85	89.74 ± 0.57	47.76 ± 0.77	65.30 ± 0.76
ICI v2 (Wang et al. 2021b)	ResNet-12	80.74 ± 0.61	87.16 ± 0.36	-	-
BML (He et al. 2022)	ResNet-12	73.45 ± 0.47	88.04 ± 0.33	-	-
Meta-NVG (Zhang et al. 2021a)	ResNet-12	74.63 ± 0.91	86.45 ± 0.59	46.40 ± 0.81	61.33 ± 0.71
SSR (Shen et al. 2021b)	ResNet-12	72.00 ± 0.60	78.50 ± 0.40	-	-
TPMN (He et al. 2022)	ResNet-12	75.50 ± 0.90	87.20 ± 0.60	46.93 ± 0.71	63.26 ± 0.74
MABAS (Rizve et al. 2021)	ResNet-12	73.51 ± 0.92	85.65 ± 0.65	42.31 ± 0.75	58.16 ± 0.78
RFS-distill (Rizve et al. 2021)	ResNet-12	73.90 ± 0.80	86.90 ± 0.50	44.60 ± 0.70	60.90 ± 0.60
DPGN (Yang et al. 2020)	ResNet-12	77.90 ± 0.50	90.20 ± 0.40	-	-
MetaOptNet (Wang et al. 2021b)	ResNet-12	72.60 ± 0.70	84.30 ± 0.50	41.10 ± 0.60	55.50 ± 0.60
Centroid (Afrasiyabi, Lalonde, and Gagné 2020)	ResNet-18	-	-	45.83 ± 0.48	59.74 ± 0.56
RCA(Ours)	ResNet-12	**81.40 ± 0.37**	**91.28 ± 0.23**	**51.06 ± 0.12**	**65.56 ± 0.61**
RCA(Ours)	ResNet-18	**81.93 ± 0.43**	**91.84 ± 0.37**	**53.30 ± 0.45**	**68.23 ± 0.57**

Table 4: Cross-domain few-shot classification results from Mimi-Imagenet → CUB-200-2011.

Model	Backbone	5-way 1-shot	5-way 5-shot
MAML (Zhang et al. 2021b)	Conv4-64	34.01 ± 1.25	-
BASELINE++ (Zhang et al. 2021b)	Conv4-64	39.19 ± 0.12	-
LFWT (Zhang et al. 2021b)	ResNet-10	47.47 ± 0.75	66.98 ± 0.68
LRP (CAN) (Zhang et al. 2021b)	ResNet-12	46.23 ± 0.42	66.58 ± 0.39
RCA(Ours)	ResNet-12	**48.44 ± 0.57**	**70.71 ± 0.09**
S2M2 (Zhang et al. 2021b)	WRN-28-10	48.24 ± 0.84	70.44 ± 0.75
OVE PG GP (Snell and Zemel 2020)	ResNet-18	39.66 ± 0.18	55.71 ± 0.31
SimpleShot (Zhang et al. 2021b)	ResNet-18	46.68 ± 0.49	65.56 ± 0.70
MetaQDA (Zhang et al. 2021b)	ResNet-18	48.88 ± 0.64	68.59 ± 0.59
RCA(Ours)	ResNet-18	**49.63 ± 0.48**	**71.33 ± 0.27**

Table 5: Models using attention methods or not and comparison different ways of generating attention mechanism on Mini-Imagenet with the ResNet-18 backbone.

Linear	Convolution	RC	5-way 1-shot	5-way 5-shot
-	-	-	61.45 ± 0.12	77.56 ± 0.43
√	-	-	68.91 ± 0.37	84.51 ± 0.22
-	√	-	64.58 ± 0.41	83.78 ± 0.29
-	-	√	**75.21 ± 0.42**	**88.59 ± 0.59**

Visualization Analysis

To qualitatively evaluate whether the RCA enhances the features from the backbone or forces the backbone to extract more representative features, we visualize the distributions of the features from the RCA, the linear transformation, and the convolutional operator by t-SNE (Van der Maaten and Hinton 2008). As shown in Fig. 1, our method makes the backbone to extract features with low intra-class distribution and high inter-class variance.

Conclusions

In this work, we introduce the Reservoir Computing Based Attention Network (RCA), employing an optimized fine-tuning strategy for few-shot image classification. Leveraging the intrinsic critical dynamics of reservoir computing without reservoir training, we devised a specific topology for FSL that exhibits robust generalization and mitigates overfitting. Experimental results demonstrate that RCA surpasses state-of-the-art methods on several public datasets. Future work may extend RCA to diverse networks and deep-learning applications.

References

Afrasiyabi, A.; Lalonde, J.-F.; and Gagné, C. 2020. Associative alignment for few-shot image classification. In *ECCV*, 18–35.

Antoniou, A.; Edwards, H.; and Storkey, A. 2019. How to train your MAML. In *ICLR*.

Aswolinskiy, W.; Reinhart, R. F.; and Steil, J. 2018. Time series classification in reservoir-and model-space. *NPL*, 48(2): 789–809.

Bahri, Y.; Kadmon, J.; Pennington, J.; Schoenholz, S. S.; Sohl-Dickstein, J.; and Ganguli, S. 2020. Statistical mechanics of deep learning. *Annu. Rev. Condens. Matter Phys.*, 11(1): 501–528.

Bertinetto, L.; Henriques, J. F.; Torr, P.; and Vedaldi, A. 2018. Meta-learning with differentiable closed-form solvers. In *ICLR*.

Bi, Y.; Xue, B.; and Zhang, M. 2021. Dual-tree genetic programming for few-shot image classification. *IEEE T. Evolut. Comput.*, 26(3): 555–569.

Boney, R.; and Ilin, A. 2018. Semi-supervised few-shot learning with MAML. In *ICLR*.

Chen, W.-Y.; Liu, Y.-C.; Kira, Z.; Wang, Y.-C. F.; and Huang, J.-B. 2019a. A closer look at few-shot classification. In *ICLR*.

Chen, Z.; Fu, Y.; Zhang, Y.; Jiang, Y.-G.; Xue, X.; and Sigal, L. 2019b. Multi-level semantic feature augmentation for one-shot learning. *IEEE TIP*, 28(9): 4594–4605.

Elsken, T.; Staffler, B.; Metzen, J. H.; and Hutter, F. 2020. Meta-learning of neural architectures for few-shot learning. In *CVPR*, 12365–12375.

Fei, N.; Lu, Z.; Xiang, T.; and Huang, S. 2020. Melr: Meta-learning via modeling episode-level relationships for few-shot learning. In *ICLR*.

Fu, J.; Liu, J.; Tian, H.; Li, Y.; Bao, Y.; Fang, Z.; and Lu, H. 2019. Dual attention network for scene segmentation. In *CVPR*, 3146–3154.

He, Y.; Liang, W.; Zhao, D.; Zhou, H.-Y.; Ge, W.; Yu, Y.; and Zhang, W. 2022. Attribute surrogates learning and spectral tokens pooling in transformers for few-shot learning. In *CVPR*, 9119–9129.

Hong, J.; Fang, P.; Li, W.; Zhang, T.; Simon, C.; Harandi, M.; and Petersson, L. 2021. Reinforced attention for few-shot learning and beyond. In *CVPR*, 913–923.

Hou, R.; Chang, H.; Ma, B.; Shan, S.; and Chen, X. 2019. Cross attention network for few-shot classification. *NeurIPS*, 32.

Hu, S. X.; Moreno, P. G.; Xiao, Y.; Shen, X.; Obozinski, G.; Lawrence, N. D.; and Damianou, A. C. 2020. Empirical bayes transductive meta-learning with synthetic gradients. In *ICLR*.

Hui, B.; Zhu, P.; Hu, Q.; and Wang, Q. 2019. Self-attention relation network for few-shot learning. In *ICME Workshops*, 198–203.

Jaeger, H. 2001. The "echo state" approach to analysing and training recurrent neural networks-with an erratum note. *Bonn, Germany: German National Research Center for Information Technology GMD Technical Report*, 148(34): 13.

Jalalvand, A.; Demuynck, K.; De Neve, W.; and Martens, J.-P. 2018. On the application of reservoir computing networks for noisy image recognition. *Neurocomputing*, 277: 237–248.

Koprinkova-Hristova, P. D. 2021. Reservoir computing approach for gray images segmentation. arXiv:2107.11077.

Krizhevsky, A.; Sutskever, I.; and Hinton, G. E. 2012. Imagenet classification with deep convolutional neural networks. *NeurIPS*, 25.

Li, H.; Eigen, D.; Dodge, S.; Zeiler, M.; and Wang, X. 2019. Finding task-relevant features for few-shot learning by category traversal. In *CVPR*, 1–10.

Li, W.-H.; Liu, X.; and Bilen, H. 2022. Cross-domain few-shot learning with task-specific adapters. In *CVPR*, 7161–7170.

Liang, H.; Zhang, Q.; Dai, P.; and Lu, J. 2021. Boosting the generalization capability in cross-domain few-shot learning via noise-enhanced supervised autoencoder. In *ICCV*, 9424–9434.

Maass, W.; Natschläger, T.; and Markram, H. 2002. Real-time computing without stable states: A new framework for neural computation based on perturbations. *Neural Comput.*, 14(11): 2531–2560.

Oreshkin, B.; Rodríguez López, P.; and Lacoste, A. 2018. Tadam: Task dependent adaptive metric for improved few-shot learning. *NeurIPS*, 31.

Qiao, S.; Liu, C.; Shen, W.; and Yuille, A. L. 2018. Few-shot image recognition by predicting parameters from activations. In *CVPR*, 7229–7238.

Qin, Z.; Wang, H.; Mawuli, C. B.; Han, W.; Zhang, R.; Yang, Q.; and Shao, J. 2022. Multi-instance attention network for few-shot learning. *Inf. Sci.*

Ravi, S.; and Larochelle, H. 2017. Optimization as a model for few-shot learning. In *ICLR*.

Ren, M.; Liao, R.; Fetaya, E.; and Zemel, R. 2019. Incremental few-shot learning with attention attractor networks. *NeurIPS*, 32.

Ren, M.; Triantafillou, E.; Ravi, S.; Snell, J.; Swersky, K.; Tenenbaum, J. B.; Larochelle, H.; and Zemel, R. S. 2018. Meta-learning for semi-supervised few-shot classification. In *ICLR*.

Rizve, M. N.; Khan, S.; Khan, F. S.; and Shah, M. 2021. Exploring complementary strengths of invariant and equivariant representations for few-shot learning. In *CVPR*, 10836–10846.

Royle, J. A.; Dorazio, R. M.; and Link, W. A. 2007. Analysis of multinomial models with unknown index using data augmentation. *J. Comput. Graph. Stat.*, 16(1): 67–85.

Rusu, A. A.; Rao, D.; Sygnowski, J.; Vinyals, O.; Pascanu, R.; Osindero, S.; and Hadsell, R. 2018. Meta-learning with latent embedding optimization. In *ICLR*.

Santoro, A.; Bartunov, S.; Botvinick, M.; Wierstra, D.; and Lillicrap, T. 2016. Meta-learning with memory-augmented neural networks. In *ICML*, 1842–1850.

Shen, S.; Baevski, A.; Morcos, A.; Keutzer, K.; Auli, M.; and Kiela, D. 2021a. Reservoir transformers. In *IJCNLP*, 4294–4309.

Shen, X.; Xiao, Y.; Hu, S. X.; Sbai, O.; and Aubry, M. 2021b. Re-ranking for image retrieval and transductive few-shot classification. *NeurIPS*, 34: 25932–25943.

Skowronski, M. D.; and Harris, J. G. 2006. Minimum mean squared error time series classification using an echo state network prediction model. In *ISCAS*, 4–12.

Snell, J.; Swersky, K.; and Zemel, R. 2017. Prototypical networks for few-shot learning. *NeurIPS*, 30.

Snell, J.; and Zemel, R. 2020. Bayesian few-shot classification with one-vs-each Pólya-Gamma augmented gaussian processes. In *ICLR*.

Tanaka, G.; Yamane, T.; Héroux, J. B.; Nakane, R.; Kanazawa, N.; Takeda, S.; Numata, H.; Nakano, D.; and Hirose, A. 2019. Recent advances in physical reservoir computing: A review. *Neural Networks*, 115: 100–123.

Touvron, H.; Cord, M.; Douze, M.; Massa, F.; Sablayrolles, A.; and Jégou, H. 2021. Training data-efficient image transformers & distillation through attention. In *ICML*, 10347–10357.

Triantafillou, E.; Larochelle, H.; Zemel, R.; and Dumoulin, V. 2021. Learning a universal template for few-shot dataset generalization. In *ICML*, 10424–10433. PMLR.

Van der Maaten, L.; and Hinton, G. 2008. Visualizing data using t-SNE. *JMLR*, 9(11).

Vinyals, O.; Blundell, C.; Lillicrap, T.; Wierstra, D.; et al. 2016. Matching networks for one shot learning. *NeurIPS*, 29.

Wah, C.; Branson, S.; Welinder, P.; Perona, P.; and Belongie, S. 2011. The caltech-ucsd birds-200-2011 dataset.

Wang, X.; Girshick, R.; Gupta, A.; and He, K. 2018. Non-local neural networks. In *CVPR*, 7794–7803.

Wang, Y.; Xu, C.; Liu, C.; Zhang, L.; and Fu, Y. 2020. Instance credibility inference for few-shot learning. In *CVPR*, 12836–12845.

Wang, Y.; Xu, Z.; Wang, X.; Shen, C.; Cheng, B.; Shen, H.; and Xia, H. 2021a. End-to-end video instance segmentation with transformers. In *CVPR*, 8741–8750.

Wang, Y.; Zhang, L.; Yao, Y.; and Fu, Y. 2021b. How to trust unlabeled data instance credibility inference for few-shot learning. *IEEE TPAMI*.

Wang, Z.; Miao, Z.; Zhen, X.; and Qiu, Q. 2021c. Learning to learn dense gaussian processes for few-shot learning. *NeurIPS*, 34: 13230–13241.

Wei, X.; Wang, B.; Chen, M.; Yuan, J.; Lan, H.; Shi, J.; Tang, X.; Jin, B.; Chen, G.; and Yang, D. 2021. ViR: the vision reservoir. arXiv:2112.13545.

Wu, F.; Smith, J. S.; Lu, W.; Pang, C.; and Zhang, B. 2020. Attentive prototype few-shot learning with capsule network-based embedding. In *ECCV*, 237–253.

Xie, J.; Long, F.; Lv, J.; Wang, Q.; and Li, P. 2022. Joint distribution matters: Deep brownian distance covariance for few-shot classification. In *CVPR*, 7972–7981.

Xu, C.; Fu, Y.; Liu, C.; Wang, C.; Li, J.; Huang, F.; Zhang, L.; and Xue, X. 2021. Learning dynamic alignment via meta-filter for few-shot learning. In *CVPR*, 5182–5191.

Yang, L.; Li, L.; Zhang, Z.; Zhou, X.; Zhou, E.; and Liu, Y. 2020. Dpgn: Distribution propagation graph network for few-shot learning. In *CVPR*, 13390–13399.

Yang, S.; Liu, L.; and Xu, M. 2020. Free lunch for few-shot learning: Distribution calibration. In *ICLR*.

Ye, H.-J.; Hu, H.; Zhan, D.-C.; and Sha, F. 2020. Few-shot learning via embedding adaptation with set-to-set functions. In *CVPR*, 8808–8817.

Yu, Z.; Chen, L.; Cheng, Z.; and Luo, J. 2020. Transmatch: A transfer-learning scheme for semi-supervised few-shot learning. In *CVPR*, 12856–12864.

Zhang, C.; Ding, H.; Lin, G.; Li, R.; Wang, C.; and Shen, C. 2021a. Meta navigator: Search for a good adaptation policy for few-shot learning. In *ICCV*, 9435–9444.

Zhang, M.; Zhang, J.; Lu, Z.; Xiang, T.; Ding, M.; and Huang, S. 2020. IEPT: Instance-level and episode-level pretext tasks for few-shot learning. In *ICLR*.

Zhang, X.; Meng, D.; Gouk, H.; and Hospedales, T. M. 2021b. Shallow bayesian meta learning for real-world few-shot recognition. In *ICCV*, 651–660.

Zhu, L.; and Yang, Y. 2020. Label independent memory for semi-supervised few-shot video classification. *TPAMI*, 44(1): 273–285.