Knowledge-Based Systems 282 (2023) 111130

Contents lists available at ScienceDirect

Knowledge-Based Systems

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

Learning to complement: Relation complementation network for few-shot class-incremental learning

Ye Wang [a], Yaxiong Wang [b], Guoshuai Zhao [c], Xueming Qian [a,d,*]

[a] School of Information and Communication Engineering, Xi'an Jiaotong University, Xi'an 710049, China
[b] School of Computer and Information Science, Hefei University of Technology, Hefei, 230000, China
[c] School of Software Engineering, Xi'an Jiaotong University, Xi'an 710049, China
[d] Ministry of Education Key Laboratory for Intelligent Networks and Network Security, Xi'an 710049, China

ARTICLE INFO

Keywords:
Lifelong learning
Class-incremental learning
Few-shot class-incremental learning
Image recognition

ABSTRACT

Real-world industrial scenarios pose a challenging task known as few-shot class-incremental learning (FSCIL), which aims to recognize new classes using a few samples while not forgetting the old classes. Despite the recent advance of FSCIL, most existing methods rely on a single metric for making incremental relation predictions, which is unilateral and lacks stability. In this paper, we remedy this issue from two aspects. Specifically, to make convincing relation predictions, we first propose a relation complementation strategy that aggregates different metric models to investigate the comprehensive relation of classifier weights and test features. Then, to make the proposed strategy well fit the incremental scenarios, we design a pseudo incremental relation complementation learning scheme that constructs the learning tasks by mimicking the data setting in real incremental sessions. Taken together, our proposed method dubbed **R**elation **C**omplementation **N**etwork (RCN) achieves the state-of-the-art performance on *mini*ImageNet, CIFAR100 and CUB200. Our code is available at https://github.com/YeZiLaiXi/KT-RCN.git.

1. Introduction

Deep neural networks (DNNs) have achieved great success in many vision tasks [1,2], but these methods can only process predefined classes. In real-world industrial scenarios, the number of classes that needs to be processed grows continually. The conventional solution is to train the model using the data of old and new classes. Obviously, this solution costs substantial time and effort. In response to this weakness, class-incremental learning (CIL) is proposed [3], aiming to learn new classes fast while maintaining the performance on old classes. Despite the success of current CIL methods [4–6], the key factor for this success is the large number of annotated training samples available for new class learning. However, annotating a large number of training samples still costs time and effort, and the number of training samples in some scenarios, such as identifying rare bird species or part defects, is limited, which often makes these methods fail due to the overfitting problem. In response to such challenging incremental scenarios, few-shot class-incremental learning (FSCIL) [7] is proposed.

FSCIL inherits the characteristics of CIL and few-shot learning (FSL): several learning sessions come in sequence like the common CIL, but the number of training samples for each new class is limited, as FSL assumes. The first (base) session provides sufficient training samples for model learning, but the following (incremental) sessions only possess limited training samples for each class. In each session, the model is trained with only the current session's data but evaluated using the test sets of all encountered classes. The challenges lie in FSCIL are catastrophic forgetting and overfitting problems due to the scarcity of new training samples.

The model decoupling strategy [8–10], which freezes the backbone in incremental sessions, owns a good trait of mitigating catastrophic forgetting and overfitting issues, but it also leads to poor representations for new classes due to the frozen backbone lacking prior information about new classes. With the poor representations, the model is easy to give unreliable relation predictions. To remedy this issue, most existing methods augment the representation by introducing several trainable linear layers [9,10] in the incremental sessions or ensembling different architectures [11]. Despite the advance of these methods, both of them rely on a single metric to make relation predictions which is unilateral and lacks stability due to the limited training samples in the incremental sessions. In contrast to these methods, we propose

* Corresponding author at: School of Information and Communication Engineering, Xi'an Jiaotong University, Xi'an 710049, China.
E-mail addresses: xjtu2wangye@stu.xjtu.edu.cn (Y. Wang), wangyx15@stu.xjtu.edu.cn (Y. Wang), guoshuai.zhao@xjtu.edu.cn (G. Zhao), qianxm@mail.xjtu.edu.cn (X. Qian).

https://doi.org/10.1016/j.knosys.2023.111130
Received 16 July 2023; Received in revised form 24 October 2023; Accepted 30 October 2023
Available online 3 November 2023
0950-7051/

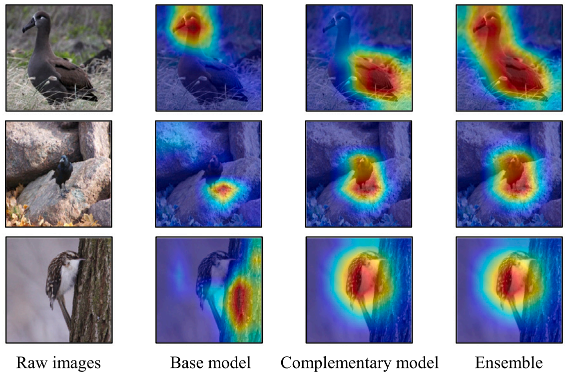


Fig. 1. Our proposed method introduces a complementary model (column 3) to complement the insufficient of the base model (column 2). As a result, satisfactory results are obtained using our proposed method (column 4).

to ensemble different metric models to make convincing relation predictions. Particularly, in this paper, we achieve this by introducing a complementary model with a squared euclidean-distance classifier to couple with the widely used cosine-based metric model (base model). In such a way, as shown in Fig. 1, we can complement the original representation given by the base model by projecting a sample to different feature space. Meanwhile, we can complement the original prediction results by aggregating the prediction results of another metric space, thus obtaining convincing relation prediction results.

To help the complementary model learn to complement the incremental relation measuring results given by the base model, we attempt to optimize the parameters of the complementary model under the incremental setting, such that the model will fit well to the incremental sessions. However, the limitation of using old data and scarce training samples in incremental sessions make such training impossible. Motivated by recent works [8,12], we propose a **P**seudo **I**ncremental **R**elation **C**omplementation (PIRC) learning scheme to construct pseudo incremental tasks with the data sampled from the base session. Concretely, each pseudo incremental task constructed by PIRC consists of three components: the pseudo base classifier weights, the pseudo incremental data, and the synthesized incremental data. PIRC combines the pseudo base classifier weights and the pseudo incremental data to construct the pseudo global incremental task, which ensures the learning objective is consistent with that of FSCIL. Meanwhile, PIRC introduces the synthesized incremental data and combines it with the pseudo incremental data to construct the pseudo local incremental task. The superiority of this operation is that the diversity of pseudo incremental classes can be enhanced by introducing the synthesized data, thus making the pseudo local incremental task more effective. By such a way, we can sufficiently and efficiently utilizes the data provided by the base session to help the complementary model *learn to complement*.

Our main contributions in this paper are summarized as follows:

- **A relation complementation strategy** is proposed, which ensembles different metrics to investigate the comprehensive relation of classifier weights and test features.
- **A pseudo incremental relation complementation learning scheme** is specially designed for FSCIL, in which we construct pseudo incremental tasks globally and locally to help the complementary model learn to complement.
- **Competitive performance**. Quantitative and qualitative experimental results on *mini*ImageNet, CUB200, and CIFAR100 demonstrate the superiority of our proposed method over previous methods.

The remainder of this article is structured as follows. Section 2 provides a brief review of some related works. Section 3 presents preliminary knowledge about FSCIL. Our proposed method is detailed in Section 4. In Section 5, we report the experimental results. The conclusion of this paper is made in Section 6.

2. Related work

2.1. Few-shot learning

Few-shot learning (FSL) aims to develop machine learning algorithms capable of processing unseen classes using limited training samples. Scarce training samples provide limited prior knowledge, making it challenging for the model to recognize unseen classes. To address this issue, the learning paradigm in FSL is often organized as meta-tasks similar to the inference task. Based on this learning paradigm, recent methods can be divided into three categories: metric-based, optimization-based, and hallucination-based methods. The metric-based methods [13,14] leverage different metrics or networks, such as Euclidean or graph neural networks(GNNs), to construct the nearest neighbor classifier to measure the similarity between the prototypes and query samples, where the prototypes are often given by class-wise mean features of the support samples. The optimization-based methods [15,16] design different meta-learners or optimization strategies to learn to adapt to different query sets with the support set, such as the classical and famous optimization-based method, MAML [15], which designs a two-stage optimization strategy that makes the model learn to use the support samples to initialize the model for different tasks. This strategy is conducted by optimizing the model using the support samples in the inner loop and optimizing the model using the query samples in the outer loop when training the meta model. The hallucination-based methods design or utilize various generation models or modules to generate or predict the classifier weights [17] or fake samples [18]. Additionally, some researchers [19,20] adopt the semi-supervised learning paradigm, which combines both the labeled and unlabeled data, to compensate for the insufficient new class data.

Although this research field is similar to FSCIL, most FSL methods do not consider the performance on old classes, while FSCIL aims to achieve good performance on both old and new classes.

2.2. Class-incremental learning

The goal of class-incremental learning (CIL) is to continually learn new classes while maintaining the performance on old classes. Because the old data are restricted to use in the incremental phase and the number of training samples for new classes is sufficient, the main problem in CIL is the notorious catastrophic forgetting problem. To solve this problem, the regularization-based methods [3,21,22] distill the knowledge from previous tasks when training the new task to prevent the model from forgetting old, such as Li et al. [3] propose to distill the outputs of classification model while Douillard et al. [21] proposed distilling each layer's features. The rehearsal-based methods [23,24] adopt different strategies, such as reservoir sampling, to restore samples of the previous task and then use them either as inputs or constrain to alleviate catastrophic forgetting when training the new task. To store more old samples with limited memory, Wang et al. [24] propose reducing the image's quality. The isolation-based methods [25,26] introduce extra parameters for each new task, for example, Yan et al. [25] trained a new feature encoder for each new task. With the emergence of foundation model [27,28], rehearsal-free methods design various prompt-based strategies to learn corresponding knowledge for different incremental tasks [29,30].

Despite the advances made by current CIL methods, these methods often assume that many training samples can be used to learn the new classes, which is not suitable for some incremental scenarios where the number of training samples for new classes is limited.

2.3. Few-shot class-incremental learning

FSCIL aims to learn a global classifier in phases, where the number of new training samples is scarce. Because the number of training samples for new classes is scarce, the model often suffers from notorious catastrophic forgetting and overfitting problems in incremental sessions. To mitigate these issues, the knowledge distillation strategy is adopted by most FSCIL methods [31,32]. For example, Dong et al. [31] proposed distilling the relation between different classes to balance stability and plasticity. Cheraghian et al. [32] designed a semantic-guided distillation strategy that distills the semantic information to prevent catastrophic forgetting. As an alternative solution, the model decoupling strategy is broadly adopted [8–10,33]. The model decoupling strategy decouples the learning of representation and classifier. However, because the encoder is frozen in the incremental sessions, the model's plasticity is constrained by the knowledge learned in the base session. To solve this problem, mimicking the incremental setting to construct pseudo incremental tasks becomes an emerging and effective solution. For example, Zhu et al. [12] proposed a pseudo incremental learning learning scheme named random episode selection strategy (RESS) that constructs a series of global pseudo incremental tasks by sampling part of the old data as pseudo new data and the prototypes of other base classes as the pseudo old prototypes. Zhang et al. [8] designed a learning scheme that constructs the local pseudo incremental tasks by sampling and rotating several randomly sampled classes, where the data of each sampled class is limited, similar to the setting in the real incremental sessions. Chi et al. [34] design a learning scheme that constructs sequential-based pseudo incremental tasks as in real incremental settings and design a module to update the model using several support samples. Recently, considering that a single model mainly focuses on one-side knowledge which limits the ability to resist catastrophic forgetting, Ji et al. [11] propose to ensemble different models to capture diverse knowledge to mitigate such limitations, where a CNN architecture is introduced to capture global knowledge and a transformer architecture is introduced to capture local knowledge.

Similar to most previous methods, we adopt the decoupling strategy to design our method. Unlike previous proposed learning schemes, our proposed pseudo incremental relation complementation learning scheme constructs the pseudo incremental tasks globally and locally, which not only coincides with the learning objective of FSCIL but can also improve the model's plasticity. Furthermore, we ensemble different metrics to give convincing incremental relation estimations rather than different architectures.

3. Preliminary knowledge

Before delving into the details of our methodology, we first introduce the problem definition of few-shot class-incremental learning (FSCIL). The goal of FSCIL is to learn a global classifier in phases that classifies all the seen classes, where each learning phase is also called a session in FSCIL. The incremental setting of FSCIL is as follows. Formally, let $\mathcal{D}^0 \rightarrow \mathcal{D}^1 \rightarrow \ldots$ denote the data stream. The classes contained in different sessions satisfy $\mathcal{C}^i \cap \mathcal{C}^j = \emptyset (i \neq j)$. Each $\mathcal{D}^i$ consists of a training set $\mathcal{D}^i_{train}$ and a test set $\mathcal{D}^i_{test}$, where only $\mathcal{D}^0_{train}$ contains many samples, $\mathcal{D}^i_{train}(i > 0)$ contains a few samples. In session i, only $\mathcal{D}^i_{train}$ is available. In contrast, the union of $\{\mathcal{D}^0_{test}, \ldots, \mathcal{D}^i_{test}\}$ is used to evaluate the performance of the model. Under the background of the model decoupling strategy, the essential problem that needs to be solved in FSCIL is an incremental relation measuring problem between test features and classifier weights. However, scarce training samples make such a problem challenging in incremental sessions.

4. Method

In this section, the overall framework is first described in Section 4.1. Then, we describe the conventional training paradigm in Section 4.2. Next, we detail our proposed pseudo incremental relation complementation learning scheme in Section 4.3. Finally, we detail the inference process in Section 4.4.

4.1. Framework overview

Our proposed method consists of a base model with the cosine classifier as in previous FSCIL methods [8,12,35] and a complementary model with the squared Euclidean-based classifier. We first adopt the conventional training paradigm to learn the parameters of the base model. Then, as shown in Fig. 2, our proposed pseudo incremental relation complementation learning scheme constructs pseudo incremental tasks globally and locally to learn the parameters of the complementary model. For the sake of following description, we denote the following feature encoding of the base model and the complementary model as $f_1(x) = \mathcal{N}(x; \theta_1)$ and $f_2(x) = \mathcal{N}(x; \theta_2)$, where θ_1 and θ_2 refer to the parameters of the base model and the complementary model's encoder.

4.2. Conventional training paradigm

Sufficient training samples in the base session enable us to train a satisfactory classification model to classify base classes. However, simply employing a linear layer as the classification layer will result in an imbalance magnitude between the base and future incremental classes [8,36], compromising the model's performance. Therefore, we replace the linear classification layer with the cosine classifier. Concretely, let x denote the image data. We first input the x to the base model and compute the classification score P as follows:

$$P = \mathrm{softmax}(s\Phi_1(f_1(x), W_1)), \tag{1}$$

where s is the scale factor, $\Phi_1(a, b) = \frac{a \cdot b}{\|a\|_2 \|b\|_2}$ is the cosine classifier, $\cdot$ refers to the inner product, and W_1 refers to the classifier weights. After obtaining the classification score P, we optimize the parameters θ_1 and W_1 by

$$\theta_1^*, W_1^* = \underset{\theta_1, W_1}{\arg\min}\, \mathcal{L}_{ce}(P, y), \tag{2}$$

where $\mathcal{L}_{ce}$ denotes the cross-entropy loss function, and y is the ground truth of x.

4.3. Pseudo incremental relation complementation learning

4.3.1. Pseudo incremental task construction

In FSCIL, the scarce training samples in the incremental sessions make it difficult to further train the model, resulting in a common problem, *i.e.* , the representations for incremental classes are weak. Due to the weak representations for incremental classes, the model easily makes improper relation predictions. For this problem, our solution is to use a complementary model with a different metric from the base model and ensemble different metrics to mitigate this problem. To learn the parameters of the complementary model, a straightforward method is to train it in a conventional manner. However, the data forms in the incremental sessions are few-shot based, and the task gap between the base session and the incremental sessions makes such a method suboptimal. To help the complementary model learn to complement the relation prediction results given by the base model, the proposed pseudo incremental relation complementation learning scheme imitates the real incremental setting to construct learning tasks, which is achieved by the proposed pseudo incremental task construction (PITC). PITC constructs many pseudo incremental tasks globally and locally for each episode with the data provided by the base session, where the pseudo global incremental task is used to coincide with the learning

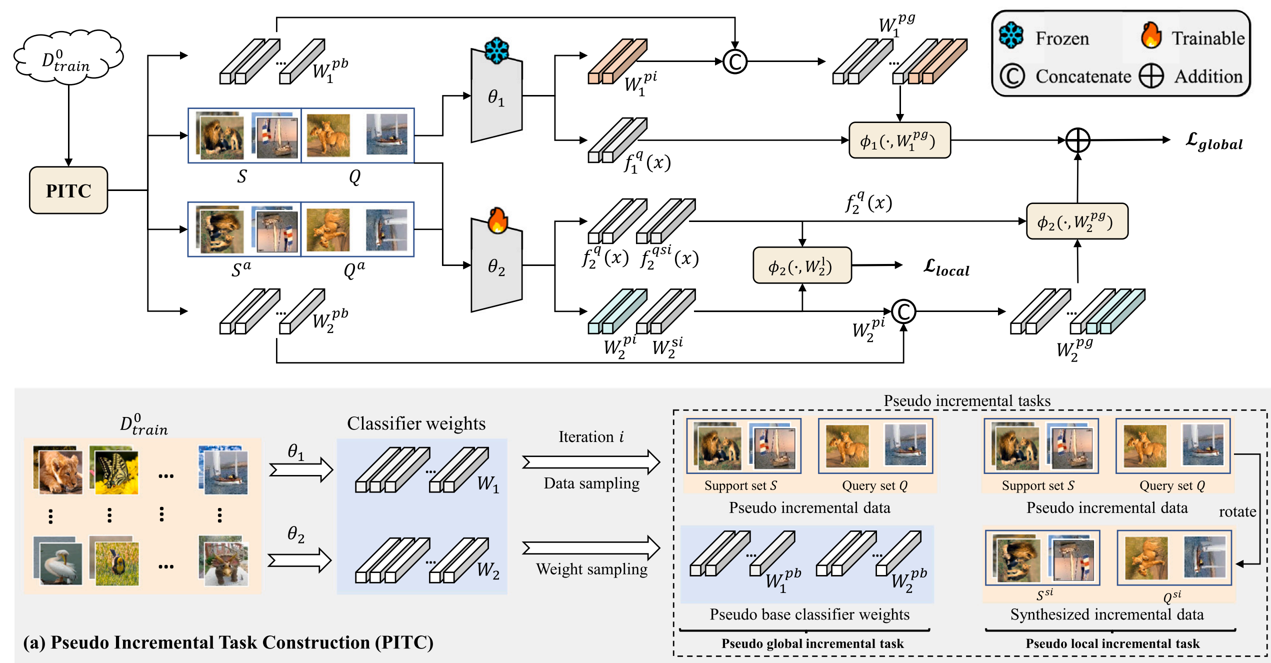


Fig. 2. Overview of our proposed pseudo incremental relation complementation learning scheme. The proposed learning scheme utilizes the (a) pseudo incremental task construction to construct pseudo incremental tasks locally and globally to help the complementary model learn to complement the base model.

objective of FSCIL, and the pseudo local incremental task is used to improve the model's plasticity. To construct these tasks, there are four main steps, weight collection, data sampling, weight sampling, and data synthesis.

- **Weight collection.** To prepare for later weight sampling, with D^0_{train}, PITC first computes the classifier weights W_1 of the base model by

$$W_1 = \mathrm{mean}(f_1(x)) \in \mathbb{R}^{N\times d}, \tag{3}$$

where N denotes the number of base classes, and d refers to the dimension of data embedding. Next, the classifier weights W_2 of the complementary model are computed by

$$W_2 = \mathrm{mean}(f_2(x)) \in \mathbb{R}^{N\times d}, \tag{4}$$

where $f_2(x)$ denotes the data embedding of D^0_{train} encoded by the complementary model. Finally, W_1 and W_2 are stored in the memory.

- **Data sampling.** To mimic the data setting of incremental classes, several classes from C^0 are randomly selected by PITC as the pseudo incremental classes. Then, PITC randomly samples a few data for each selected class to constitute the support set S and the query set Q, where S, and Q will serve as the training set and test set of real incremental classes, respectively.

- **Weight sampling.** To mimic the data setting of base classes in the incremental session, except the classifier weights of pseudo incremental classes, other classifier weights of W_1 and W_2 are selected as the pseudo base classifier weights W_1^{pb} and W_2^{pb} of the base model and complementary model, respectively.

- **Data synthesis.** To improve the model's plasticity, PITC synthesizes incremental data to enhance the diversity of pseudo incremental classes. Concretely, PITC rotates S and Q to synthesize the support set S^{si} and query set Q^{si} of synthesized incremental classes.

Overall, the combination $\left\{S, Q, S^{si}, Q^{si}, W_1^{pb}, W_2^{pb}\right\}$ forms a pseudo incremental task, where the combination $\left\{S, Q, W_1^{pb}, W_2^{pb}\right\}$ forms the pseudo global incremental task, and the combination $\left\{S, Q, S^{si}, Q^{si}\right\}$ forms the pseudo local incremental task.

Algorithm 1 Complementary learning.

Require: The base model $f(;\theta_1)$, the complementary model $f(;\theta_2)$, pseudo global incremental task $\left\{S, Q, S^{si}, Q^{si}, W_1^{pb}, W_2^{pb}\right\}$, pseudo local incremental task $\left\{S, Q, S^{si}, Q^{si}\right\}$.
Ensure: A trained $f(;\theta_2)$.

```
while not done do
    W_1^pi, W_2^pi ← Get the base model's and complementary model's pseudo incremental classifier weights using S, Eq. (3) and Eq. (4), respectively.
    W_1^pg ← Get the base model's pseudo global classifier weights by concatenating W_1^pi and W_1^pb
    W_2^pg ← Get the complementary model's pseudo global classifier weights by concatenating W_2^pi and W_2^pb
    P_global ← Make predictions for Q using W_1^pg, W_2^pg, Eq. (5), (6) and (7)
    L_global ← Compute the global loss by using Eq.(8)
    W_2^si ← Get the synthesized incremental classifier weights using S^si and Eq. (4)
    W_2^l ← Get the local classifier weights by concatenating W_2^pi and W_2^si
    P_local ← Make predictions for {Q, Q^si} using W_2^l, Eq. (6) and (9)
    L_local ← Compute the local loss using P_local and Eq. (10)
    L ← Compute the total loss using Eq.(11)
    Optimize the complementary model with SGD
end while
```

4.3.2. Complementary learning

With the constructed pseudo incremental tasks, the complementary learning aims to optimize the complementary model for relation calibration. The pseudo code of this stage is illustrated in Algorithm 1. With the constructed pseudo global incremental task $\left\{S, Q, W_1^{pb}, W_2^{pb}\right\}$, we first encode the data of S and Q using $f(;\theta_1)$ and $f(;\theta_2)$. The corresponding data embeddings are denoted as $f_1^s(x)$, $f_1^q(x)$, $f_2^s(x)$, and $f_2^q(x)$, respectively. Then, the pseudo incremental classifier weights W_1^{pi} of the base model are computed using Eq. (3) and $f_1^s(x)$. Analogously, the pseudo incremental classifier weights W_2^{pi} of the complementary model are computed using Eq. (4) and $f_2^s(x)$. Next, W_1^{pi} and W_1^{pb} are concatenated as the pseudo global classifier weights W_1^{pg} of the base model, while W_2^{pi} and W_2^{pb} are concatenated as the pseudo global classifier weights W_2^{pg} of the complementary model. Given the pseudo global classifier weights W_1^{pg} and W_2^{pg}, the respective relations can be calculated accordingly:

$$r_1 = \Phi_1(f_1^q(x), W_1^{pg}), \tag{5}$$

$$r_2 = \Phi_2(f_2^q(x), W_2^{pg}), \tag{6}$$

where r_1 and r_2 are the relation estimations given by the base model and complementary model, respectively, and $\Phi_2(a,b) = -\|a-b\|^2/d$ refers to the squared Euclidean distance-based classifier. With r_1 and r_2, the final incremental relation measuring P_{global} is given by integrating the above two predictions:

$$P_{global} = \text{softmax}(s(\frac{r_1}{d} + r_2)), \tag{7}$$

where d is used to eliminate the impact of dimension. Finally, the global loss $\mathcal{L}_{global}$ is computed by the cross entropy (CE) loss:

$$\mathcal{L}_{global} = \mathcal{L}_{CE}(P_{global}, Y_{global}), \tag{8}$$

where Y_{global} refers to the ground truth of the query data contained in the pseudo global incremental task.

With the constructed pseudo local incremental task $\{S, Q, S^{si}, Q^{si}\}$, we first encode the data of S^{si} and Q^{si} using $f(;\theta_2)$. The corresponding data embeddings of S^{si} and Q^{si} are denoted as $f_2^{ssi}(x)$ and $f_2^{qsi}(x)$, respectively. Then, the synthesized incremental classifier weights W_2^{si} are computed by Eq. (4) and $f_2^{ssi}(x)$. Next, we concatenate W_2^{si} and W_2^{pi} as the local classifier weights W_2^l to classify $f_2^{qsi}(x)$ and $f_2^q(x)$ using Eq. (6). Let the computed relation be r_{local}, and the local relation estimation P_{local} is predicted by

$$P_{local} = \text{softmax}(sr_{local}). \tag{9}$$

Consequently, the local loss is defined as:

$$\mathcal{L}_{local} = \mathcal{L}_{CE}(P_{local}, Y_{local}), \tag{10}$$

where Y_{local} refers to the ground truth of the query data contained in the pseudo local incremental task.

To coincide with the learning objective of FSCIL and improve the model's new class adaptation ability, we define the total objective as:

$$\mathcal{L} = \lambda_1 \mathcal{L}_{global} + \lambda_2 \mathcal{L}_{local}, \tag{11}$$

where λ_1 and λ_2 are hyperparameters to balance the two losses.

4.4. Incremental relation measurement

In the inference stage, the training set of current incremental sessions is first used to expand the previous classifiers. For example, let $W_1{}^{\circ}$ and $W_2{}^{\circ}$ denote the old classifier weights of the base model and complementary model. In each incremental session, we first obtain the data embedding of the available training set by inputting the training set to the base model and complementary model. Then, we use Eqs. (3) and (4) to compute the classifier weights W_1^n and W_2^n of new classes, respectively. Next, we use the concatenation of $W_1{}^{\circ}$ and W_1^n to expand the old classifier of the base model. The updated classifier weights are denoted as W_1^g. Similarly, the old classifier weights $W_2{}^{\circ}$ of the complementary model are expanded by the concatenation of $W_2{}^{\circ}$ and W_2^n. We denote the updated classifier weights of the complementary model as W_2^g. Given a test sample x from the test set of all encountered classes, the incremental relation estimation P is given by

$$P = \Phi_1(f_1(x), W_1^g) + \Phi_2(f_2(x), W_2^g). \tag{12}$$

5. Experiments

5.1. Datasets

***mini*ImageNet.** The *mini*ImageNet dataset is the subset of the ImageNet [37] dataset. There are 100 classes in this dataset, where each class has 500 training images and 100 test images. Following the incremental setting proposed by [7], we split this dataset into 9 (1+8) sessions, where the base session consists of 60 classes and the following 8 incremental sessions consist of the remaining 40 classes. Each session takes the 5-way-5-shot setting, which implies that there are five classes and each class has five training images.

CIFAR100. As in *mini*ImageNet, the CIFAR100 [38] consists of 100 classes, and each class has 500 training images and 100 test images, but the size of the image is small. We follow the incremental learning setting in *mini*ImageNet to split this dataset, *i.e.* , the base session consists of 60 classes, the following 8 incremental sessions consist of 40 classes, and each incremental session takes the 5-way-5-shot setting.

Caltech-UCSD Birds-200-2011. The CUB200 [39] dataset consists of 200 classes, and each class has approximately 30 training images and 30 test images. Following the incremental setting proposed by [7], we split this dataset into 11 (1+10) sessions, where the base session consists of 100 classes, the following 10 incremental sessions consist of remaining 100 classes, and each session takes a 10-way-5-shot setting.

5.2. Implementation details

We adopt the PyTorch [40] platform to implement our proposed method. Following [12,33], ResNet18 is adopted as the encoder for benchmark datasets.

• In the pretraining stage, on CUB200, the base model is trained for 50 epochs with a batch size of 128 using the SGD optimizer. The initial learning rate, weight decay, and momentum are set to 0.03, 0.0001, and 0.9, respectively. We decay the learning rate by a factor of 0.1 per 10 epochs. On CIFAR100 and *mini*ImageNet, except we set the batch size to 64, the learning rate to 0.1, the weight decay to 0.0005, and decay the learning rate by a factor of 0.1 every 40 epochs, others are the same as the setting on CUB200.

• In the pseudo incremental relation complementation learning stage, we freeze the base model and train the complementary model for 80 epochs. In each epoch, 200 pseudo incremental tasks are randomly constructed. We adopt the SGD optimizer to optimize the model. We set the initial learning rate, weight decay, and momentum to 0.03, 0.0001, and 0.9, respectively. We decay the learning rate by a factor of 0.1 per 20 epochs. The scale factors are set to 16, 16, and 12 for CIFAR100, CUB200, and *mini*ImageNet respectively. Following Zhu et al. [12], we adopt the random resized crop, random horizontal flip, and color jitter techniques to augment the data during training.

5.3. Evaluation protocol

In the inference stage of each session, the test sets until the current session are used to evaluate the performance of the model, and the top-1 accuracy is reported. To evaluate the model's overall performance, we compute the average accuracy $\texttt{Avg.} = \frac{1}{M+1}\sum_{i=0}^{M}\mathcal{A}_i$ across all sessions, where M represents the number of incremental sessions and $\mathcal{A}_i$ represents the top-1 accuracy of the i−th session. Following previous class incremental learning methods [29,41], we also compute the performance gap $\texttt{Diff.} = \mathcal{A}_M - \mathcal{A}_M^{ub}$ between the method and the upper-bound method Joint-CNN, where Joint-CNN represents the method that uses both the training data of old and new classes to train the model in each session and $\mathcal{A}_M^{ub}$ represents the accuracy in the last session of Joint-CNN.

Table 1
Comparison with other methods on *mini*ImageNet.

Method	Sessions									Avg.	Diff.
	0	1	2	3	4	5	6	7	8		
Joint-CNN	81.20	75.62	70.66	65.81	62.20	58.41	55.78	53.16	50.00	63.65	0.00
NCM[a] [42]	61.31	47.80	39.31	31.91	25.68	21.35	18.67	17.24	14.17	30.83	−35.83
iCaRL[a] [23]	61.31	46.32	42.94	37.63	30.49	24.00	20.89	18.80	17.21	33.29	−32.79
EEIL[a] [43]	61.31	46.58	44.00	37.29	33.14	27.12	24.10	21.57	19.58	34.97	−30.42
TOPIC [7]	61.31	50.09	45.17	41.16	37.48	35.52	32.19	29.46	24.42	39.64	−25.58
SPPR [12]	61.45	63.80	59.53	55.53	52.50	49.60	46.69	43.79	41.92	52.76	−8.08
CEC [8]	72.00	66.83	62.97	59.43	56.70	53.73	51.19	49.24	47.63	57.75	−2.37
F2M [33]	72.05	67.47	63.16	59.70	56.71	53.77	51.11	49.21	47.84	57.89	−2.16
MCNet [11]	72.33	67.70	63.50	60.34	57.59	54.70	52.13	50.41	49.08	58.64	−0.92
MetaFSCIL [34]	72.04	67.94	63.77	60.29	57.58	55.16	52.90	50.79	49.19	58.85	−0.81
MFS3 [44]	73.65	68.91	64.60	61.48	58.68	55.55	53.33	51.69	50.26	59.79	+0.26
FACT [45]	72.56	69.63	66.38	62.77	60.60	57.33	54.34	52.16	50.49	60.70	+0.49
C-FSCIL [9]	76.40	71.14	66.46	63.29	60.42	57.46	54.78	53.11	51.41	61.61	+1.41
SoftNet [35]	79.77	75.08	70.59	66.93	64.00	61.00	57.81	55.81	54.68	65.07	+4.68
ALICE [46]	80.60	70.60	67.40	64.50	62.50	60.00	57.80	56.80	55.70	63.99	+5.70
NC-FSCIL [10]	84.02	76.80	72.00	67.83	66.35	64.04	61.46	59.54	58.31	67.82	+8.31
RCN(Ours)	**84.62**	**79.94**	**75.70**	**72.21**	**69.38**	**66.26**	**63.48**	**61.39**	**60.02**	**70.33**	**+10.02**

[a] Represents the results copied from [7].

Table 2
Comparison with other methods on CIFAR100.

Method	Sessions									Avg.	Diff.
	0	1	2	3	4	5	6	7	8		
Joint-CNN	80.15	74.57	69.93	65.31	61.00	57.79	54.47	51.59	49.66	62.72	0.00
NCM[a] [42]	64.10	53.05	43.96	36.97	31.61	26.73	21.23	16.78	13.54	34.22	−36.12
iCaRL[a] [23]	64.10	53.28	41.69	34.13	27.93	25.06	20.41	15.48	13.73	32.87	−35.93
EEIL[a] [43]	64.10	53.11	43.71	35.15	28.96	24.98	21.01	17.26	15.85	33.79	−33.81
TOPIC [7]	64.10	55.88	47.07	45.16	40.11	36.38	33.96	31.55	29.37	42.62	−20.29
SPPR [12]	63.97	65.86	61.31	57.60	53.39	50.93	48.27	45.36	43.32	54.45	−6.34
CEC [8]	73.07	68.88	65.26	61.19	58.09	55.57	53.22	51.34	49.14	59.53	−0.52
F2M [33]	71.45	68.10	64.43	60.80	57.76	55.26	53.53	51.57	49.35	59.14	−0.31
MetaFSCIL [34]	74.50	70.10	66.84	62.77	59.48	56.52	54.36	52.56	49.97	60.79	+0.31
C-FSCIL [9]	77.47	72.40	67.47	63.25	59.84	56.95	54.42	52.47	50.47	61.64	+0.81
MCNet [11]	73.30	69.34	65.72	61.70	58.75	56.44	54.59	53.01	50.72	60.40	+1.06
MFS3 [44]	73.42	69.85	66.44	62.81	59.78	56.94	55.04	53.00	51.07	60.93	+1.41
FACT [45]	74.60	72.09	67.56	63.52	61.38	58.36	56.28	54.24	52.10	62.24	+2.44
ALICE [46]	79.00	70.50	67.10	63.40	61.20	59.20	58.10	56.30	54.10	63.21	+4.44
SoftNet [35]	79.88	75.54	71.64	67.47	64.45	61.09	59.07	57.29	55.33	65.75	+5.67
NC-FSCIL [10]	82.52	76.82	73.34	69.68	66.19	62.85	60.96	59.02	56.11	67.50	+6.45
RCN(Ours)	**83.40**	**78.75**	**74.94**	**70.81**	**67.84**	**64.89**	**63.10**	**60.92**	**58.53**	**69.24**	**+8.87**

[a] Represents the results copied from [7].

5.4. Comparison methods

To validate the effectiveness of our proposed method, we compare it with some classical class-incremental learning methods (iCaRL [23], EEIL [43], and NCM [42]) and recent FSCIL methods (TOPIC [7], SPPR [12], CEC [8], F2M [33], C-FSCIL [9], MetaFSCIL [34], FACT [45], ALICE [46], SoftNet [35], MCNet [11], and NC-FSCIL [10]). The descriptions of these methods are presented as follows:

- **iCaRL** stores the data of learned classes by the herding strategy [47] and replays them to mitigate catastrophic forgetting problems.
- **EEIL** further proposes a balanced fine-tuning strategy that selects the same number of old and new training samples to finetune the model after finishing the training in each session.
- **NCM** incorporates cosine normalization, a less-forget constraint, and inter-class separation to learn a unified classifier to balance the bias between old and new data.
- **TOPIC** constrains the topology of the feature space to mitigate the catastrophic forgetting problem.
- **SPPR** utilizes the relations between old and new prototypes to update the global prototypes and proposes a learning scheme that constructs pseudo global incremental tasks to learn functional modules.
- **CEC** propagates context information between old and new classifiers to update the classifier and proposes a learning scheme that constructs pseudo local incremental tasks to learn functional modules.
- **F2M** introduces random noise to the encoder's parameters to find the base training objective's flat local minima and fine-tunes the model within this minima in the incremental sessions.
- **C-FSCIL** freezes the encoder and replays old features with new data to fine-tune the fully-connected layer to adapt the model's outputs to new classes.
- **MetaFSCIL** proposes a learning scheme that optimizes the model by constructing sequential pseudo local incremental tasks.
- **FACT** squeezes the learned classes' feature space to reserve the space for new class learning.
- **ALICE** adopts angular penalty loss and augmentation strategies to improve the generalization ability of the model.
- **SoftNet** freezes the major part and updates the minor part of a subnetwork obtained by a soft mask to mitigate the catastrophic forgetting and overfitting problems.
- **MCNet** enhances the representations for new classes by ensembling information captured by different architectures.
- **NC-FSCIL** preassigned classifier prototypes and fine-tunes a projection layer to drive the output features into their corresponding prototypes.

Table 3
Comparison with other methods on CUB200.

Method	Sessions											Avg.	Diff.
	0	1	2	3	4	5	6	7	8	9	10		
Joint-CNN	78.68	73.49	69.86	66.10	64.74	62.47	60.64	59.32	57.25	57.67	57.50	64.34	0.00
NCM[a] [42]	68.68	57.12	44.21	28.78	26.71	25.66	24.62	21.52	20.12	20.06	19.87	32.49	−37.63
iCaRL[a] [23]	68.68	52.65	48.61	44.16	36.62	29.52	27.83	26.26	24.01	23.89	21.16	36.67	−36.34
EEIL[a] [43]	68.68	53.63	47.91	44.20	36.30	27.46	25.93	24.70	23.95	24.13	22.11	36.27	−35.39
TOPIC [7]	68.68	62.49	54.81	49.99	45.25	41.40	38.35	35.36	32.22	28.31	26.28	43.92	−31.22
SPPR [12]	68.68	61.85	57.43	52.68	50.19	46.88	44.65	43.07	40.17	39.63	37.33	49.34	−20.17
CEC [8]	75.85	71.94	68.50	63.50	62.43	58.27	57.73	55.81	54.83	53.52	52.28	61.33	−5.22
MFS3 [44]	75.63	72.51	69.65	65.29	63.13	60.38	58.99	57.41	55.55	54.95	53.47	62.45	−4.03
MetaFSCIL [34]	75.90	72.41	68.78	64.78	62.96	59.99	58.30	56.85	54.78	53.82	52.64	61.93	−4.86
F2M [33]	77.13	73.92	70.27	66.37	64.34	61.69	60.52	59.38	57.15	56.94	55.89	63.96	−1.61
SoftNet [35]	78.07	74.58	71.37	67.54	65.37	62.60	61.07	59.37	57.53	57.21	56.75	64.68	−0.75
FACT [45]	75.90	73.23	70.84	66.13	65.56	62.15	61.74	59.83	58.41	57.89	56.94	64.42	−0.56
MCNet [11]	77.57	73.96	70.47	65.81	66.16	63.81	62.09	61.82	60.41	60.09	59.08	65.57	+1.58
NC-FSCIL [10]	**80.45**	75.98	72.30	**70.28**	68.17	65.16	64.43	63.25	60.66	60.01	59.44	67.28	+1.94
ALICE [46]	77.40	72.70	70.60	67.20	65.90	63.40	62.90	61.90	60.50	60.60	60.10	65.75	+2.60
RCN(Ours)	79.86	**76.48**	**73.34**	69.72	**68.48**	**65.93**	**64.58**	**63.68**	**62.04**	**61.48**	**60.47**	**67.82**	**+2.97**

[a] Represents the results copied from [7].

Table 4
Ablation studies on *mini*ImageNet. **PIRC** refers to the proposed pseudo incremental relation complementation learning scheme. Compared with single metric, ensembling different metrics achieves better performance, and our proposed PIRC can further boost the performance.

B-model	C-model	PIRC	Sessions								
			0	1	2	3	4	5	6	7	8
√			81.87	76.82	72.29	68.60	65.44	62.45	59.57	57.42	55.98
	√		80.83	74.32	70.20	66.68	63.79	60.92	58.34	56.61	55.18
√	√		82.98	77.60	73.31	69.87	67.03	64.15	61.56	59.83	58.70
	√	√	82.80	77.66	73.59	69.83	67.00	63.48	60.49	58.38	56.68
√	√	√	**84.62**	**79.94**	**75.70**	**72.21**	**69.38**	**66.26**	**63.48**	**61.39**	**60.02**

5.5. Quantitative comparisons

As can be observed from Tables 1, 2, and 3,

- On *mini*ImageNet, CIFAR100, and CUB200, the performance of CIL methods, such as iCaRL, EEIL, and NCM, decreases significantly as the learning process proceeds. The intrinsic reason is that these methods overfit the training data because the training samples in the incremental sessions are scarce.
- On *mini*ImageNet, compared to other methods, our proposed method achieves the best performance in each session. Particularly, compared to the second-best method NC-FSCIL, the average accuracy `Avg.` of our proposed RCN has an improvement of **2.51**%, while the performance gap `Diff.` with the Joint-CNN of our proposed RCN has an improvement of **1.71**%.
- On CIFAR100, compared to other methods, our proposed RCN also achieves the best performance in each session. Particularly, compared to the second-best method NC-FSCIL, the average accuracy `Avg.` of our proposed RCN has an improvement of **1.74**%, while the performance gap `Diff.` with Joint-CNN of our proposed RCN has an improvement of **2.42**%.
- On CUB200, compared to other methods, our proposed RCN achieves the best performance in almost all sessions as on *mini*ImageNet and CIFAR100. Particularly, on the average accuracy `Avg.`, compared to the second-best method NC-FSCIL, our proposed RCN has an improvement of **0.54**%. On the performance gap `Diff.` with the Joint-CNN, our proposed RCN has an improvement of **0.37**% compared to the second-best method ALICE.

In summary, the quantitative comparisons demonstrate the superiority of our proposed RCN compared to other methods.

5.6. Ablation study

To validate the effectiveness of each component in RCN, we conduct several ablation studies on *mini*ImageNet and CUB200. As we can see from Table 4, Row 1/2 refers to using only the base model (B-model) / the complementary model (C-model) trained by the conventional training paradigm to make incremental predictions, and the accuracy in the last session is 55.98%/55.18%. Row 3 refers to combining the B-model and C-model to make incremental predictions. The accuracy in the last session is 58.70% which surpasses that given by using the B-model or the C-model alone by a margin of **2.72**% and **3.52**%, respectively. The results demonstrate that ensembling different metrics is an effective strategy for FSCIL. Row 4 refers to using our proposed pseudo incremental relation complementation (PIRC) learning scheme to train the C-model. We can see that our proposed PIRC boosts the C-model (Row 2) performance in the last session from 55.18% to 56.68%. Similarly, Row 5 refers to using PIRC to train the C-model and combining the B-model and C-model to make incremental predictions, we can see that our proposed PIRC boosts the performance of the full model (Row 3) from 58.70% to 60.02%. Furthermore, as we can

Fig. 3. Visualization with GradCAM on CUB200, where (a) raw images of new classes, (b) results without using our proposed pseudo incremental relation complementation learning scheme, (c) results with using our proposed pseudo incremental relation complementation learning scheme.

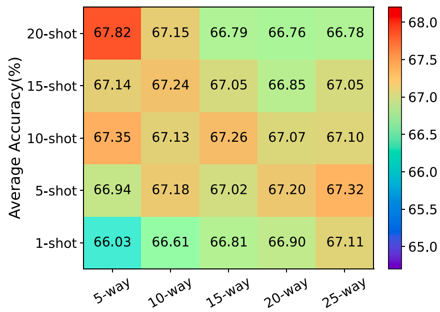


Fig. 4. The performance under various pseudo-incremental settings on CUB200 in the pseudo incremental relation complementation learning stage. The method we propose prefers a small way and a large shot.

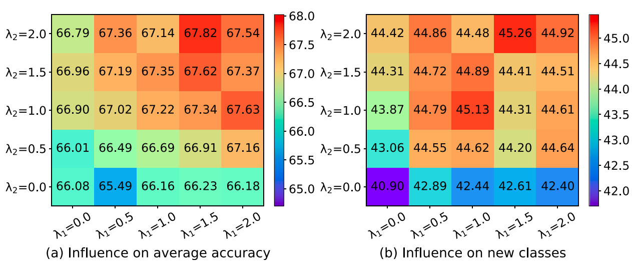


Fig. 5. The influence of losses on CUB200, where we change λ_1 and λ_2 among different values and report the (a) average accuracy and (b) the accuracy on new classes.

see from Fig. 3, using our proposed PIRC can help the model capture more effective regions of the target for new classes than not using our proposed PIRC. The quantitative and visualization results shown above demonstrate that our proposed PIRC is effective.

5.7. Discussion

5.7.1. The influence of sampling setting

In the pseudo incremental relation complementation learning stage, we adopt the N-way-K-shot setting to sample the data of pseudo incremental classes from the base session. To study the influence of the N-way-K-shot setting on average accuracy in the pseudo incremental relation complementation learning stage, we change the number of ways among $\{5, 10, 15, 20, 25\}$ and the number of shots among $\{1, 5, 10, 15, 20\}$. The results given by different combinations are reported in Fig. 4. Regardless of whether the number of ways or shots is fixed, our proposed method can achieve a satisfactory result as long as suitable shots or ways are set. Particularly, setting the number of ways to 5 and the number of shots to 20 achieves the highest average accuracy on CUB200. The primary reason we guess is that the data of constructed pseudo tasks is sampled from the base session. Setting a small way can reduce overfitting, and a large shot can provide sufficient prior information.

5.7.2. The influence of $\mathcal{L}_{global}$ and $\mathcal{L}_{local}$

To study the influence of $\mathcal{L}_{global}$ and $\mathcal{L}_{local}$, we change λ_1 and λ_2 among $\{0, 0.5, 1, 1.5, 2.0\}$ and report the average accuracy and the accuracy on new classes. As shown in Fig. 5(a), using only $\mathcal{L}_{global}/\mathcal{L}_{local}$, the maximum average accuracy is 66.23%/66.96%. When we combine $\mathcal{L}_{global}$ and $\mathcal{L}_{local}$ to optimize the model, the maximum average accuracy is 67.82%. The results demonstrate that $\mathcal{L}_{global}$ and $\mathcal{L}_{local}$ are effective. As we can see from Fig. 5(b), using only $\mathcal{L}_{global}$ improves the model's performance on new classes from 40.90% to 42.89%. The results indicate that the model's plasticity can be improved by making the pseudo incremental task coincide with the learning objective of FSCIL. Furthermore, combining $\mathcal{L}_{global}$ and $\mathcal{L}_{local}$ to optimize the model, the model's performance on new classes given by using only $\mathcal{L}_{global}$ is improved from 42.89% to 45.26%. The results indicate that $\mathcal{L}_{local}$ can further improve the model's plasticity. In summary, the results described above indicate that $\mathcal{L}_{global}$ and $\mathcal{L}_{local}$ are effective and can be used to improve the model's plasticity. Particularly, setting λ_1 to 1.5 and λ_2 to 2.0 is an optimal configuration.

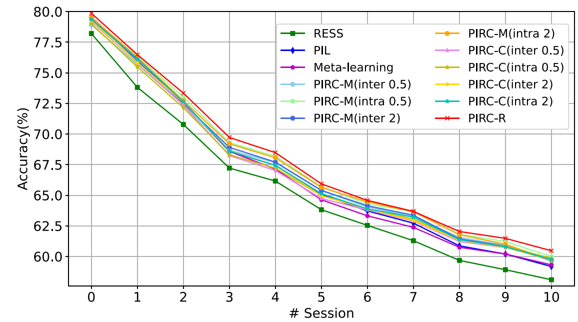


Fig. 6. Comparison with different pseudo incremental learning schemes on CUB200, where RESS [12], PIL [8], and meta-learning [34] are the previously proposed pseudo incremental learning schemes, PIRC refers to our proposed pseudo incremental relation complementation learning scheme, M, C, R, inter and intra refer to MixUp [48], CutMix [49], Rotate [8], class-wise operation and sample-wise operation, respectively. Values shown in the bracket are coefficients of Beta distribution $B(\alpha, \alpha)$. Detailed descriptions of RESS, PIL and meta-learning can be found in Section 2.3.

5.7.3. Further analysis of pseudo incremental relation complementation learning

To further study the influence of pseudo incremental relation complementation learning (PIRC), we adopt different strategies to synthesize the data and compare the corresponding performance with the previously proposed pseudo incremental learning schemes RESS [12], PIL [8], and meta-learning [34]. As we can see from Fig. 6, compared to the performance given by RESS, PIL, or meta-learning, our proposed PIRC can achieve the highest accuracy on each session on CUB200 under different conditions. The results demonstrate that optimizing the model from the global perspective to coincide with the learning objective of FSCIL and the local perspective to improve the model's plasticity is a more effective than global task-focused RESS, local task-focused PIL, and sequential local task-focused meta-learning. Furthermore, we can see that using rotation [8] can help our proposed method achieve better performance than the benefit brought by using MixUp [48] or CutMix [49]. The results show that rotate [8] is more effective than MixUp [48] and CutMix [49].

5.7.4. The influence of ensemble strategies

To study the influence of ensemble strategies, we use different metric-level and output-level ensemble strategies and report the corresponding results in Table 5 and Fig. 7. From the metric level, as we can see from Table 5, using the KL divergence shows the strongest catastrophic forgetting resistance ability compared to other metrics. Using the squared Euclidean distance shows the best plasticity compared to other metrics and achieves the highest average accuracy. From the output level, as shown in Fig. 7, we can observe that simply adding the relation logits of the base model and complementary model without any further operation obtains the best performance.

5.7.5. The sensitivity of the number of incremental shots

To explore the influence of the number of shots in the inference stage, we change the number of shots among $\{1, 5, 10, 20, 50\}$. As can be seen from Fig. 8(a), increasing the number of shots from 1 to 5, the performance on incremental sessions is improved by a large margin. However, such improvement gets small when we continually increase the number of shots. The main reasons are that increasing the number

Table 5
The influence of metric-level ensemble strategies, where `Base acc.` refers to the accuracy on the base classes, `New acc.` refers to the accuracy on the new classes, Cos., KL div., and S-Euc. represent the metric models using cosine, KL divergence, and squared Euclidean distance as the classifier, respectively.

Type	Base acc.	New acc.	Avg.
B-Model	75.14	40.90	66.08
+Cos.	75.28	41.03	66.07
+KL div.	**77.13**	39.79	66.73
+S-Euc.	75.63	**45.26**	**67.82**

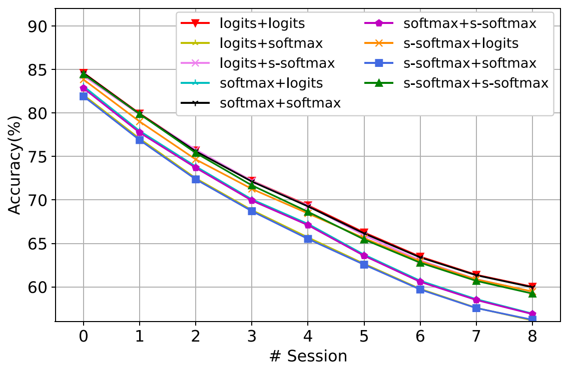
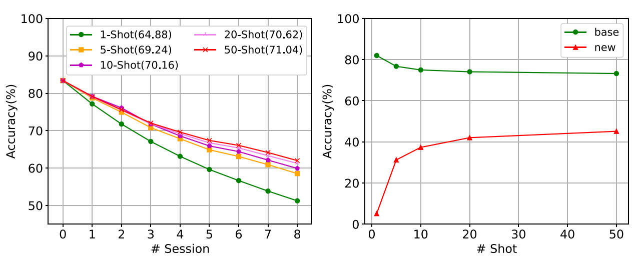


Fig. 7. The influence of output-level ensemble strategies, where logits refer to the estimation results given by the classifier directly, s-softmax refers to multiplying the logits with a scale factor and then applying the softmax function.



(a) Influence of incremental shots   (b) Influence on base and new classes

Fig. 8. The performance given by different incremental shots on CIFAR100 in the inference stage, where (a) the performance in each session, (b) the accuracy on base and new classes, values shown in the bracket of the legend is the average accuracy across all sessions.

of shots from 1 to 5 slightly drops the model's performance on old classes but improves the model's performance on new classes by a large margin as shown in Fig. 8(b), and increasing the number of shots from 5 to a larger value seems to have a slight influence on old and new classes.

5.7.6. The sensitivity of the number of incremental steps

To explore the influence of the number of steps in incremental sessions, we change the number of steps among $\{1, 2, 4, 5, 8, 20, 40\}$. As can be seen from Fig. 9, increasing the number of steps drops the performance, but the decreasing trend is gradually flattening out as the number of steps increases. The results demonstrate that our proposed method is not sensitive to the incremental steps as in conventional class-incremental learning.

5.7.7. The sensitivity of incremental orders

To explore the influence of incremental order, we use the general order given by [7] as the baseline and randomly shuffle the order five times. As we can see from Fig. 10, different incremental orders mainly influence the performance on middle incremental sessions, and

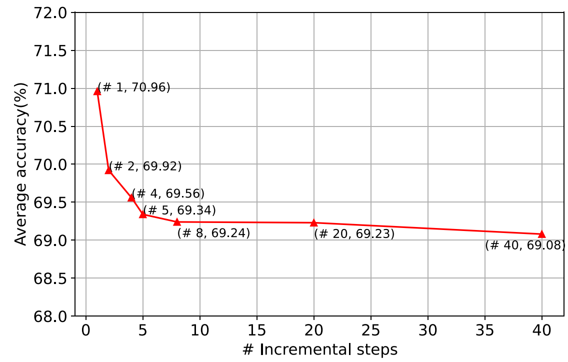


Fig. 9. The influence of incremental steps, where (# a, b) refers to the number of incremental steps is a and the average accuracy is b.

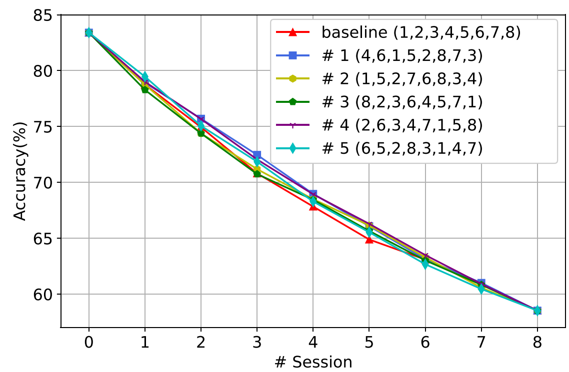


Fig. 10. The influence of incremental order on CIFAR100, where values shown in the bracket of the legend are relative incremental orders.

the influence is slight for our proposed method. The results show that our proposed method has a strong resistance ability for the incremental order.

6. Conclusion

In this paper, we focus on solving the challenging few-shot class-incremental learning (FSCIL) task. Firstly, a relation complementation strategy that integrates different metric models is proposed to perform incremental relation measuring. Secondly, an effective pseudo incremental relation complementation learning scheme is proposed to help the complementary model learn to complement the relation prediction results given by the base model. The proposed learning scheme mimics the real incremental setting and constructs the pseudo incremental tasks globally and locally, where the pseudo global incremental task is used to coincide with the learning objective of FSCIL, and the pseudo local incremental task is used to improve the model's plasticity. Experiments are conducted on three FSCIL datasets, and the quantitative and qualitative results demonstrate the superiority of our proposed method compared to previous methods.

Limitations and future work: Although the proposed pseudo incremental relation complementation learning scheme is effective and general as demonstrated in the main paper, the large size of the model makes it difficult to deploy the model to some memory-constrained devices. Moreover, the performance of our proposed method is limited by the model's representation ability for new classes. In future work, based on this work, we will consider designing a model with a small size and has strong representation ability for new classes, even if the number of training samples for the new classes is limited.

CRediT authorship contribution statement

Ye Wang: Writing – original draft, Software, Methodology, Formal analysis, Data curation. **Yaxiong Wang:** Supervision, Methodology, Formal analysis, Conceptualization. **Guoshuai Zhao:** Writing – review & editing. **Xueming Qian:** Supervision, Resources.

Declaration of competing interest

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

Data availability

I have shared the link to my code.

Acknowledgments

This work was supported by in part by the NSFC, China under Grant 62272380 and 62103317, the Science and Technology Program of Xi'an, China under Grant 21RGZN0017, Natural Science Foundation in Shanxi Province of China under grant 2021JQ-289, and China Postdoctoral Science Foundation, China under Grant 2021M700533.

References

[1] C. Ye, X. Li, S. Lai, Y. Wang, X. Qian, Scale adaption-guided human face detection, Knowl.-Based Syst. 253 (2022) 109499.
[2] Y. Xue, S. Liu, Y. Li, P. Wang, X. Qian, A new weakly supervised strategy for surgical tool detection, Knowl.-Based Syst. 239 (2022) 107860.
[3] Z. Li, D. Hoiem, Learning without forgetting, in: ECCV, 2016, pp. 614–629.
[4] Q. Wang, J. Liu, Z. Ji, Y. Pang, Z. Zhang, Hierarchical correlations replay for continual learning, Knowl.-Based Syst. 250 (2022) 109052.
[5] Z. Ji, J. Liu, Q. Wang, Z. Zhang, Coordinating experience replay: A harmonious experience retention approach for continual learning, Knowl.-Based Syst. 234 (2021) 107589.
[6] F. Mao, W. Weng, M. Pratama, E.Y.K. Yee, Continual learning via inter-task synaptic mapping, Knowl.-Based Syst. 222 (2021) 106947.
[7] X. Tao, X. Hong, X. Chang, S. Dong, X. Wei, Y. Gong, Few-shot class-incremental learning, in: CVPR, 2020.
[8] C. Zhang, N. Song, G. Lin, Y. Zheng, P. Pan, Y. Xu, Few-shot incremental learning with continually evolved classifiers, in: CVPR, 2021, pp. 12455–12464.
[9] M. Hersche, G. Karunaratne, G. Cherubini, L. Benini, A. Sebastian, A. Rahimi, Constrained few-shot class-incremental learning, in: CVPR, 2022, pp. 9057–9067.
[10] Y. Yang, H. Yuan, X. Li, Z. Lin, P. Torr, D. Tao, Neural collapse inspired feature-classifier alignment for few-shot class-incremental learning, in: ICLR, 2023.
[11] Z. Ji, Z. Hou, X. Liu, Y. Pang, X. Li, Memorizing complementation network for few-shot class-incremental learning, IEEE Trans. Image Process. 32 (2023) 937–948.
[12] K. Zhu, Y. Cao, W. Zhai, J. Cheng, Z.-J. Zha, Self-promoted prototype refinement for few-shot class-incremental learning, in: CVPR, 2021, pp. 6801–6810.
[13] Y. Liu, W. Zhang, C. Xiang, T. Zheng, D. Cai, X. He, Learning to affiliate: Mutual centralized learning for few-shot classification, in: CVPR, 2022, pp. 14411–14420.
[14] J. Zhou, Q. Lv, C.Y.-C. Chen, Dynamic concept-aware network for few-shot learning, Knowl.-Based Syst. 258 (2022) 110045.
[15] C. Finn, P. Abbeel, S. Levine, Model-agnostic meta-learning for fast adaptation of deep networks, in: ICML, Vol. 70, 2017, pp. 1126–1135.
[16] Y. Zheng, X. Zhang, Z. Tian, W. Zeng, S. Du, Detach and unite: A simple meta-transfer for few-shot learning, Knowl.-Based Syst. 277 (2023) 110798.
[17] Y. Guo, N.-M. Cheung, Attentive weights generation for few shot learning via information maximization, in: CVPR, 2020.
[18] J. Xu, H. Le, Generating representative samples for few-shot classification, in: CVPR, 2022, pp. 9003–9013.
[19] J. Lin, H. Shao, Z. Min, J. Luo, Y. Xiao, S. Yan, J. Zhou, Cross-domain fault diagnosis of bearing using improved semi-supervised meta-learning towards interference of out-of-distribution samples, Knowl.-Based Syst. 252 (2022) 109493.
[20] Z. Lao, D. He, Z. Jin, C. Liu, H. Shang, Y. He, Few-shot fault diagnosis of turnout switch machine based on semi-supervised weighted prototypical network, Knowl.-Based Syst. 274 (2023) 110634.
[21] A. Douillard, M. Cord, C. Ollion, T. Robert, E. Valle, Podnet: Pooled outputs distillation for small-tasks incremental learning, in: ECCV, 2020, pp. 86–102.
[22] K. Wei, C. Deng, X. Yang, D. Tao, Incremental zero-shot learning, IEEE Trans. Cybern. 52 (12) (2022) 13788–13799.
[23] S.-A. Rebuffi, A. Kolesnikov, G. Sperl, C.H. Lampert, iCaRL: Incremental classifier and representation learning, in: CVPR, 2017.
[24] L. Wang, X. Zhang, K. Yang, L. Yu, C. Li, L. Hong, S. Zhang, Z. Li, Y. Zhong, J. Zhu, Memory replay with data compression for continual learning, in: ICLR, 2022.
[25] S. Yan, J. Xie, X. He, DER: Dynamically expandable representation for class incremental learning, in: CVPR, 2021, pp. 3014–3023.
[26] Y. Liu, B. Schiele, Q. Sun, Adaptive aggregation networks for class-incremental learning, in: CVPR, 2021, pp. 2544–2553.
[27] A. Radford, J.W. Kim, C. Hallacy, A. Ramesh, G. Goh, S. Agarwal, G. Sastry, A. Askell, P. Mishkin, J. Clark, et al., Learning transferable visual models from natural language supervision, in: ICML, PMLR, 2021, pp. 8748–8763.
[28] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, J. Uszkoreit, N. Houlsby, An image is worth 16x16 words: Transformers for image recognition at scale, in: ICLR, 2021.
[29] Z. Wang, Z. Zhang, C.-Y. Lee, H. Zhang, R. Sun, X. Ren, G. Su, V. Perot, J. Dy, T. Pfister, Learning to prompt for continual learning, in: CVPR, 2022, pp. 139–149.
[30] Y. Wang, Z. Huang, X. Hong, S-prompts learning with pre-trained transformers: An occam's razor for domain incremental learning, in: A.H. Oh, A. Agarwal, D. Belgrave, K. Cho (Eds.), Advances in Neural Information Processing Systems, 2022.
[31] S. Dong, X. Hong, X. Tao, X. Chang, X. Wei, Y. Gong, Few-shot class-incremental learning via relation knowledge distillation, AAAI 35 (2) (2021) 1255–1263.
[32] A. Cheraghian, S. Rahman, P. Fang, S.K. Roy, L. Petersson, M. Harandi, Semantic-aware knowledge distillation for few-shot class-incremental learning, in: CVPR, 2021, pp. 2534–2543.
[33] G. Shi, J. Chen, W. Zhang, L.-M. Zhan, X.-M. Wu, Overcoming catastrophic forgetting in incremental few-shot learning by finding flat minima, NeurIPS 34 (2021) 6747–6761.
[34] Z. Chi, L. Gu, H. Liu, Y. Wang, Y. Yu, J. Tang, MetaFSCIL: A meta-learning approach for few-shot class incremental learning, in: CVPR, 2022, pp. 14166–14175.
[35] H. Kang, J. Yoon, S.R.H. Madjid, S.J. Hwang, C.D. Yoo, On the soft-subnetwork for few-shot class incremental learning, in: ICLR, 2023.
[36] S. Hou, X. Pan, C.C. Loy, Z. Wang, D. Lin, Learning a unified classifier incrementally via rebalancing, in: CVPR, 2019, pp. 831–839.
[37] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M.a. Bernstein, ImageNet large scale visual recognition challenge, IJCV 115 (3) (2015) 211–252.
[38] A. Krizhevsky, G. Hinton, Learning multiple layers of features from tiny images, in: Handbook of Systemic Autoimmune Diseases, Vol. 1, No. 4, 2009.
[39] C. Wah, S. Branson, P. Welinder, P. Perona, S. Belongie, The Caltech-UCSD Birds-200–2011 Dataset, California Institute of Technology, 2011.
[40] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, et al., Pytorch: An imperative style, high-performance deep learning library, NeurIPS 32 (2019).
[41] Z. Wang, Z. Zhang, S. Ebrahimi, R. Sun, H. Zhang, C.-Y. Lee, X. Ren, G. Su, V. Perot, J. Dy, et al., DualPrompt: Complementary prompting for rehearsal-free continual learning, in: ECCV, 2022.
[42] S. Hou, X. Pan, C.C. Loy, Z. Wang, D. Lin, Learning a unified classifier incrementally via rebalancing, in: CVPR, 2019.
[43] F.M. Castro, M.J. Marin-Jimenez, N. Guil, C. Schmid, K. Alahari, End-to-end incremental learning, in: ECCV, 2018.
[44] X. Xu, S. Niu, Z. Wang, W. Guo, L. Jing, H. Yang, Multi-feature space similarity supplement for few-shot class incremental learning, Knowl.-Based Syst. 265 (2023) 110394.
[45] D.-W. Zhou, F.-Y. Wang, H.-J. Ye, L. Ma, S. Pu, D.-C. Zhan, Forward compatible few-shot class-incremental learning, in: CVPR, 2022, pp. 9046–9056.
[46] C. Peng, K. Zhao, T. Wang, M. Li, B.C. Lovell, Few-shot class-incremental learning from an open-set perspective, in: ECCV, 2022, pp. 382–397.
[47] M. Welling, Herding dynamical weights to learn, in: ICML, 2009, pp. 1121–1128.
[48] H. Zhang, M. Cisse, Y.N. Dauphin, D. Lopez-Paz, mixup: Beyond empirical risk minimization, 2017, arXiv preprint arXiv:1710.09412.
[49] S. Yun, D. Han, S.J. Oh, S. Chun, J. Choe, Y. Yoo, Cutmix: Regularization strategy to train strong classifiers with localizable features, in: ICCV, 2019, pp. 6023–6032.