Self-Promoted Prototype Refinement for Few-Shot Class-Incremental Learning

Kai Zhu[1*] Yang Cao[1*] Wei Zhai[1] Jie Cheng[2] Zheng-Jun Zha[1†]
[1] University of Science and Technology of China [2] Huawei Technologies Co. Ltd.
{zkzy@mail., forrest@, wzhai056@mail.}ustc.edu.cn jiecheng2009@google.com zhazj@ustc.edu.cn

Abstract

Few-shot class-incremental learning is to recognize the new classes given few samples and not forget the old classes. It is a challenging task since representation optimization and prototype reorganization can only be achieved under little supervision. To address this problem, we propose a novel incremental prototype learning scheme. Our scheme consists of a random episode selection strategy that adapts the feature representation to various generated incremental episodes to enhance the corresponding extensibility, and a self-promoted prototype refinement mechanism which strengthens the expression ability of the new classes by explicitly considering the dependencies among different classes. Particularly, a dynamic relation projection module is proposed to calculate the relation matrix in a shared embedding space and leverage it as the factor for bootstrapping the update of prototypes. Extensive experiments on three benchmark datasets demonstrate the above-par incremental performance, outperforming state-of-the-art methods by a margin of 13%, 17% *and* 11%, *respectively.*

1. Introduction

Currently, deep convolutional neural networks [23, 11, 35] have made significant breakthroughs in a large number of recognition tasks. When the class is given in advance and the sample is sufficient, we can get a good recognition model by typical supervised learning. In practice, however, we are likely to encounter new classes that were not seen before in continual data stream, and need to add them into the recognition tasks, which forms the problem of class-incremental learning (CIL) [20].

In this case, it is both time consuming and computationally expensive to retrain the model on all the old and new data. And in many cases the old data may not be available, due to data privacy or limited storage. A common solution is to fine-tune old models with new data, but it may arise the problem of catastrophic forgetting. To this end, recent learning-based approaches present to maintain the representation space for the old classes by preserving memories of old classes (e.g., examplar [20]) and introducing various distillation losses, and then reconstruct classifiers (e.g., fully connected layer [31], learnable prototypes [12]) in different ways to correct their preference for new classes.

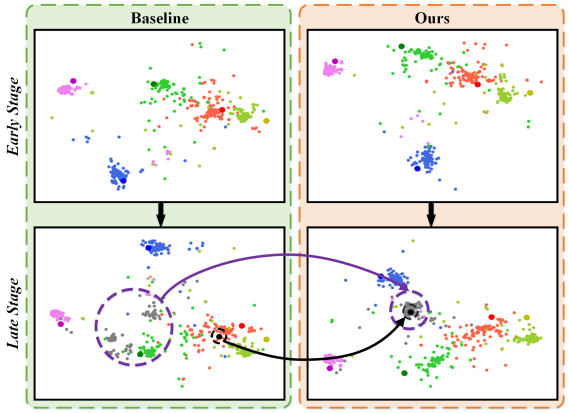

Figure 1. The t-SNE [19] results in different methods at two stages. The initial deep representations obtained by typical incremental learning method [20, 1] and our proposed method are visualized in the upper row (60 classes used, and 5 classes visualized in color). In each colored class, deep-color points are learnable prototypes, and light-color ones show the distribution of real data. The lower row shows the refined representations and prototypes of each class after the increment of the gray class. Compared with previous method, (1) the representations of the incremental classes are more clustered (regions circled in violet dotted lines), (2) and their corresponding prototypes are more discriminative, where the incremental prototype visualized in black is representative and no longer confused with that in red color.

However, existing methods assume that new class samples are available in large quantities, while the incremental classes are usually atypical and the sample size is small in practical applications. For example, in industrial visual inspection tasks, with the continuous progress of production, new classes of defects often appear due to equipment wear and other reasons. These defect samples may not only be

*Co-first Author
†Corresponding Author

essentially different from the old samples, but also small in number. It brings great difficulties to the recognition task, as representation optimization and prototype reorganization brought by new classes are hard to complete under little supervision. This paper focuses on this ability of incrementally learning new classes from few samples, which is called few-shot class-incremental learning (FSCIL [27]).

A natural idea for FSCIL task is to directly apply existing incremental learning methods to solve the problem, but experimental results show that this way results in a dramatic drop in performance. Our analysis suggests that this is mainly due to the following two reasons. First, the initial deep representation space used for CIL is relatively compact, which is conducive to classification of existing classes, but it lacks extensibility for FSCIL. In FSCIL, due to the insufficiency of incremental class samples, there is no enough supervision at each stage to participate in the classification and distillation process. Therefore, it cannot promote the expansion of the representation space as the existing incremental learning methods do. In addition, the small number of new class samples is not sufficient to learn discriminative classifier for new classes while maintaining the performance on old classes. As shown in Fig. 1, the representation space extended by typical incremental learning approach [20, 4] is underrepresented, such that the new classes usually exhibit insufficient aggregation compared to the old ones. Also, due to the insufficiency of new class samples, the prototypes used for classification are prone to be confused with other classes after incremental learning, which greatly deteriorates subsequent tasks.

To address this problem, we propose an incremental prototype learning scheme to explicitly learn an extensible feature representation, and thus facilitate subsequent incremental tasks. The scheme is mainly manifested in two aspects. First, we adopt the random episode selection strategy (RESS) to enhance the extensibility of feature representation by forcing features adaptive to various randomly simulated incremental processes. Secondly, we introduce a self-promoted prototype refinement mechanism (SPPR) to update the existing prototypes by utilizing the relation matrix between representations of the new class samples and the old class prototypes. This enhances the expressiveness of the new classes while retaining the relational characteristics among the old classes. Particularly, a novel module called dynamic relation projection is proposed to map the representation of the new class samples and the prototype of the old classes into the same embedding space, and calculate a projection matrix between them by using the distance metric of the two embeddings in the space. We take the matrix as the weight of prototype refinement to guide the dynamic change of the prototype toward maintaining the existing knowledge and enhancing the discriminability of the new class. To demonstrate the superiority of our method, we conducted comparative experiments with existing few-shot class-incremental and typical class-incremental methods on three datasets CIFAR-100, MiniImageNet and CUB200. We achieved the best results against the state-of-the-art methods, leading by 13% , 17%, and 11%, respectively.

Our main contributions are as follows:

1. An incremental prototype learning scheme is proposed for few-shot class-incremental learning, in which a randomly episodic training is accomplished by a self-promoted prototype refinement mechanism, resulting in an extensible feature representation.

2. A novel dynamic relation projection module is proposed, which uses the relational metric between old class prototypes and new class samples to constrain the update of prototypes during training and test.

3. Extensive experiments on benchmark CIFAR-100, MiniImageNet and CUB200 datasets demonstrate the superiority of our proposed method over the state-of-the-art.

2. Related Work

2.1. Incremental Learning

Although deep neural networks have shown excellent performance in many individual tasks, it still remains a substantial challenge to learn different tasks in sequence. Thus incremental learning [15] continues to receive much attention. According to whether the task identity is informed or needs to be inferred, incremental learning can be divided into three categories [28]: task-incremental learning, domain-incremental learning and class-incremental learning (CIL). CIL is the most challenging and the closest to the needs of practical applications. Therefore, a large number of related studies emerge in recent years. iCARL [20] preserves valuable samples of the old classes called exemplars with a limited capacity, and designs a set of strategies for selecting and updating them. It decouples the representation learning and classification by an examplar-rehearsed knowledge distillation and a nearest-mean-of-exemplars rule. Most of the subsequent works follow this framework and makes corresponding improvements.

NCM [12] incorporates cosine normalization, less-forget constraint and inter-class separation, to mitigate the adverse effects of the imbalance between previous and new data. To maintain fairness between old classes and new classes, BiC and WA [31, 36] correct the bias of the last fully connected layer by a linear model. To obtain the optimal exemplars for both the new class and the old class, EEIL [18] proposes a novel and automatic framework mnemonics, where they parameterize exemplars and make them optimizable in an end-to-end manner. SDC [32] proposes a new method called semantic drift compensation to deal with the drift of the data in incremental learning. PODNet and TPCIL [6, 26] replace

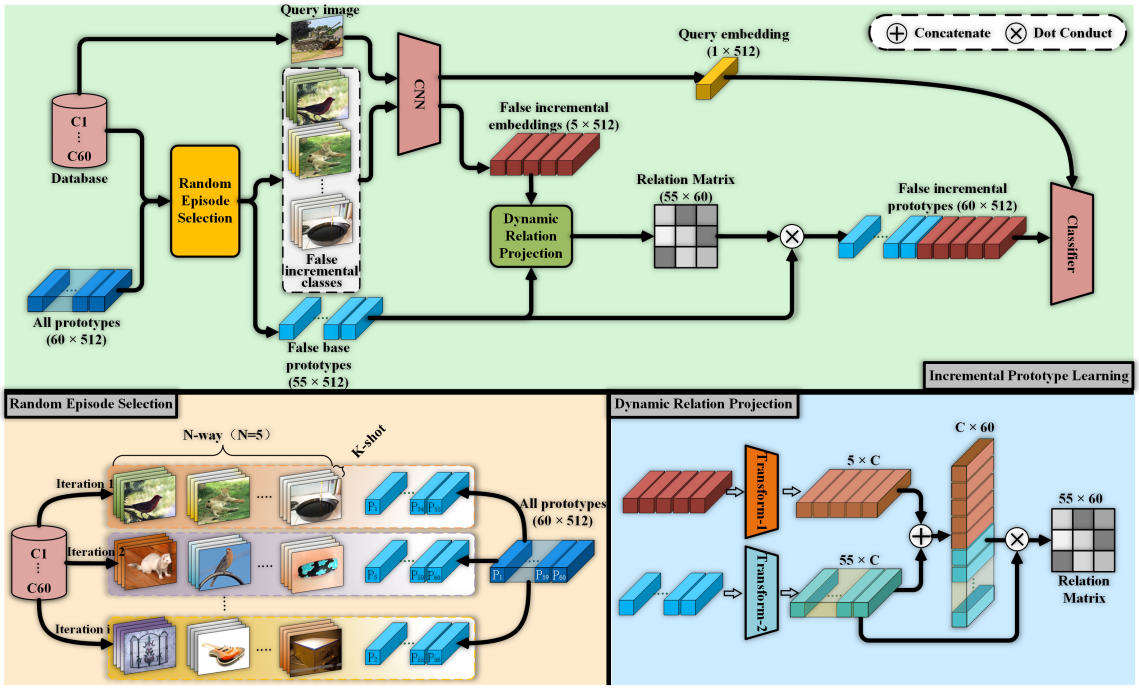

Figure 2. Our incremental prototype learning scheme for few-shot class-incremental learning. (a) Overview of our scheme. (b) Random episode selection strategy. (c) Dynamic relation projection module.

the conventional distillation losses with an efficient spatial-based distillation-loss applied throughout the model and the topology-preserving constraint, respectively.

Recently, TOPIC [27] presents a challenging but practical FSCIL problem. To address the problem, they propose a neural gas (NG) network to constrain feature space topologies for knowledge representation. We follow their FSCIL task settings. However, different from their work where a topology preservation framework is used to address insufficient samples in incremental training process, we adopt a non-training update mechanism to adapt few incremental samples based on extensible representation and explicit inter-class dependencies.

2.2. Few-shot Learning

Few-shot learning is to learn a model that recognizes the class of new samples given few reference images. Generally, corresponding methods can be divided into three categories. Metric-based methods [24, 25, 2, 3] focus on the similarity metric function over the embeddings. Meta-based methods [7, 22, 37, 10] aims to learn a learning strategy to adjust well to new samples. Augmentation-based methods [14] synthesize more data from the novel classes in different ways to facilitate standard learning.

Most of these methods focus on the fast learning of the novel classes, while neglecting the recognition accuracy on the initial classes. To address this issue, [8] proposes a dynamic few-shot recognition system with an attention based few-shot classification weight generator. [9] introduces Graph Neural Network to capture the co-dependencies between adjacent classes and promote the weight generation of new classes. These works aim to quickly adapt to novel classes from few training data while not forgetting the base classes [21] on which it was trained. Instead, we consider an extensible representation and global dependencies among classes at different sessions to maintain continuous stability during the incremental process.

3. Problem Description

The few-shot class-incremental learning (FSCIL) problem is defined as follows. Here we denote X, Y and Z as the training set, the label set and the test set, respectively. Our task is to train the model from a continuous data stream in a class-incremental form, *i.e.,* training sets $X^1, X^2, \cdots X^n$, where samples of a set X^i are from the label set Y^i, and n represents the incremental session. It should be mentioned that all the incremental classes are disjoint, that is, $Y^i \cap Y^j = \varnothing (i \neq j)$. Except that there are sufficient samples in the first session X^1, only few samples (e.g., 5 samples) are available for each class in the subsequent sessions, which is consistent with the embodiment of FSCIL. To measure the performance of models in FSCIL task, we calculate the classification accuracy on the test set Z^i at each session

i. Different from the training set, the classes of the test set Z^i are from all the seen label sets $Y^1 \bigcup Y^1 \cdots \bigcup Y^i$.

4. Method

We detail the incremental prototype learning scheme and its important components in this section. First of all, we demonstrate the paradigms of standard learning and our proposed incremental prototype learning, respectively. Then two core components random episode selection strategy and the self-promoted prototype refinement mechanism are introduced. Finally, we analyze the optimization flow of the overall pipeline and explain why it works well.

4.1. Standard Learning Paradigm

For the training process of the base classes (*i.e.,* the first session) in incremental task, standard classification pipeline is adopted. In this case, the input of the model is only the query image Q to be predicted. Then a base feature extractor f_e such as VGG [23] or ResNet [11] parameterized by θ_e is utilized to learn the corresponding representation:

$$R_q = f_e(Q; \theta_e). \tag{1}$$

Finally, a certain metric f_m parameterized by θ_m is used to measure the relationship between the representation and the learnable prototypes θ_p for all classes:

$$S = \mathrm{softmax}(f_m(R_q, \theta_p; \theta_m)). \tag{2}$$

f_m can represents a variety of classifiers, including the non-parametric ones (e.g., nearest-mean-of-exemplars classifier [20] and cosine classifier [12]) and the parametric ones (e.g., fully connected classifier [23]). Taking the cosine classifier as an example, the above formula can be written as:

$$S_i = \frac{\exp(\eta({\theta_p^i}^T \cdot R_q)))}{\sum_j \exp(\eta({\theta_p^j}^T \cdot R_q)))}, \tag{3}$$

where i is the calculated class, η is the scale factor, and $\cdot$ represents the operation of inner product. In this case, if η is learnable then θ_m refers to η, otherwise θ_m is empty. Our task is to randomly sample query images from the dataset, train and optimize θ, thus minimizing the loss function L under the supervision of target labels T:

$$\theta_* = arg\min_{\theta} L(S_i, T). \tag{4}$$

Here θ includes above θ_e, θ_p and θ_m. In classification tasks, L usually represents cross-entropy loss function.

4.2. Incremental Prototype Learning

As the representation obtained from standard learning lacks extensibility, we propose an incremental prototype learning scheme. There are two important components in the scheme as follows.

Random Episode Selection. We introduce the random episode selection strategy into the learning process and generate a N-way K-shot [29] incremental episode in each iteration. It enhances the extensibility of the feature representation by forcing gradients to adapt to different simulated incremental processes generated randomly. Unlike the few-shot task where the goal is only to identify the N classes, the simulated incremental process aims at identifying all seen classes with few samples of the N classes. Specifically, in addition to the above query image Q, the input of the model contains a randomly selected N-way K-shot collection C from the base training set X^1. As shown in Fig. 2, in each iteration, N classes are randomly selected from the label space Y^1, and then K samples are selected for the feature extractor. The obtained embeddings are averaged for each class:

$$R_s = \mathrm{mean}(f_e(C; \theta_e)). \tag{5}$$

Finally, these N classes are assumed not to have been seen before this iteration, so their corresponding prototypes will be eliminated. Mathematically,

$$\theta_p^N = \mathbb{C}_{|Y^1|}^{|Y^1|-N}(\theta_p), \tag{6}$$

where $|Y^1|$ represents the number of classes in label set Y^1, and $\mathbb{C}$ represents the mathematical operation that determines the possible arrangements in a collection of items (*i.e.,* $\mathbb{C}_n^m = \frac{n!}{m!(n-m)!}$). At this point, all the inputs and outputs of the model have been determined. Our goal is still to classify the query image Q given corresponding embeddings and prototypes, which is:

$$S = P(R_q \mid R_s, \theta_p^N). \tag{7}$$

Dynamic Relation Projection. To maintain the dependencies [33, 17, 34] of old classes and enhance the discrimination of the new classes, we propose a self-promoted prototype refinement mechanism f_u. In general, we obtain the refined prototypes θ_p under the guidance of relation matrix [16] between the embeddings of new classes and the prototypes of old classes:

$${\theta_p}' = f_u(R_s, \theta_p^N; \theta_u). \tag{8}$$

Specifically, the embeddings and the old prototypes are first transformed into a shared latent space,

$$T_s = f_{t_1}(Rs; \theta_{t_1}), \tag{9}$$

$$T_p = f_{t_2}(\theta_p^N; \theta_{t_2}), \tag{10}$$

where f_{t_1} and f_{t_2} represents a set of standard convolution block including a 1×1 convolution, a batch normalization layer and a ReLU activation layer. Then we calculate

RESS	SPPR	FT	Sessions									Average
			1	2	3	4	5	6	7	8	9	
		√	64.10	56.49	52.0	46.24	42.36	37.86	36.43	33.99	32.30	44.64
	√		64.10	61.02	56.63	52.88	49.49	46.65	44.06	39.47	37.51	50.20
	√	√	64.10	59.85	55.27	50.99	47.60	44.14	41.64	39.06	36.36	48.20
√		√	64.10	63.51	58.09	53.37	50.28	46.07	43.41	41.16	39.05	51.00
√	√		**64.10**	**66.10**	**61.43**	57.33	**53.72**	50.51	48.24	45.58	42.99	54.44
√	√	√	**64.10**	65.86	61.36	**57.34**	53.69	**50.75**	**48.58**	**45.66**	**43.25**	**54.51**

Table 1. Ablation study on CIFAR-100. All results are the average of multiple tests, and bold fonts represent the best results.

the cosine similarities between the old classes and the new classes in this space as follows:

$$T_{Y^1} = Concat([T_s, T_p]), \tag{11}$$

$$Corr = T_p \cdot T_{Y^1}^T. \tag{12}$$

At this point, we obtain the relation matrix $Corr$ between the old and new classes, and use it as the transition coefficient of prototype refinement,

$$\theta_p{}' = Corr^T \cdot \theta_p^N. \tag{13}$$

Since the refinement mechanism not only explicitly considers the relation between the new and old classes, but is also guided by the random selection process, the prototypes dynamically move toward maintaining existing knowledge and enhancing the discrimination of new classes.

4.3. Optimization

To further understand the role of the above two parts, we analyze their impacts on the optimization process. Compared to standard learning, in addition to the extra parameters of the transform module that need to be optimized, the optimization directions of the feature representations and prototypes have also been significantly changed. Specifically, we integrate the optimization process of the parameter in the feature representation θ_e as follows:

$$\begin{aligned} S_i &= f_m(R_q, \theta_p{}'; \theta_m) \\ &= f_m(f_e(Q;\theta_e), f_u(f_e(C;\theta_e), \theta_p^N; \theta_u); \theta_m). \end{aligned} \tag{14}$$

Under the new learning scheme, we not only learn a representation that is conducive to the classification of existing classes (the former θ_e in the above formula), but also encourage the network to reach an area in the parameter space where update of any class will be beneficial for subsequent incremental task (the latter θ_e). For convenience, we omit the softmax operation in the formula.

For prototypes, it is no longer just supervised by classification labels without any other constraints. Compared to Eq. 2, the learnable prototypes perform joint optimization with the representation of selected collection S under the condition of satisfying mutual relation projection. Such a projection relation, as shown in the later visualization part, is well achieved in the optimization process.

$$\begin{aligned} S_i &= f_m(R_q, \theta_p{}'; \theta_m) \\ &= f_m(f_e(Q;\theta_e), f_u(R_s, \mathbb{C}_{|Y^1|}^{|Y^1|-N}(\theta_p); \theta_u); \theta_m). \end{aligned} \tag{15}$$

5. Experiment

5.1. Dataset and Settings

Dataset. To evaluate the performance of our proposed method, we conduct comprehensive experiments on three datasets CIFAR-100 [13], MiniImageNet [29] and CUB200 [30]. CIFAR-100 contains 60000 images of 32 × 32 size from 100 classes, and each class includes 500 training images and 100 test images. MiniImageNet contains 60000 images of 84 × 84 size from ImageNet-1k [5]. Although it has the same number of classes and samples as CIFAR-100, its content is more complex and is valuable for the study of FSCIL. CUB200 is the most widely used benchmark for fine-grained image classification. The dataset covers 200 species of birds, including 5994 training images and 5794 test images. It provides more sessions and incremental classes to compare the sensitivity of different methods. For all the three datasets, we follow the same settings as [27] including the division of the datasets and incremental training samples. More details can be found in [27].

Settings. As adopted in [27], we use ResNet-18 as the backbone CNN. We show the classification accuracy at each session and the average accuracy as stated in most CIL work. To make a fair comparison, we achieve the same accuracy in base classes (*i.e.,* session=1) for all the datasets denoted as "$Ours$". We also denote the best result without the constraint of the base classes as "$Ours^*$". To reduce the error caused by random sampling of incremental samples, we select different samples to test 5 times and then take the average for each model. Since the average results are close to those with the same incremental samples as [27], we only

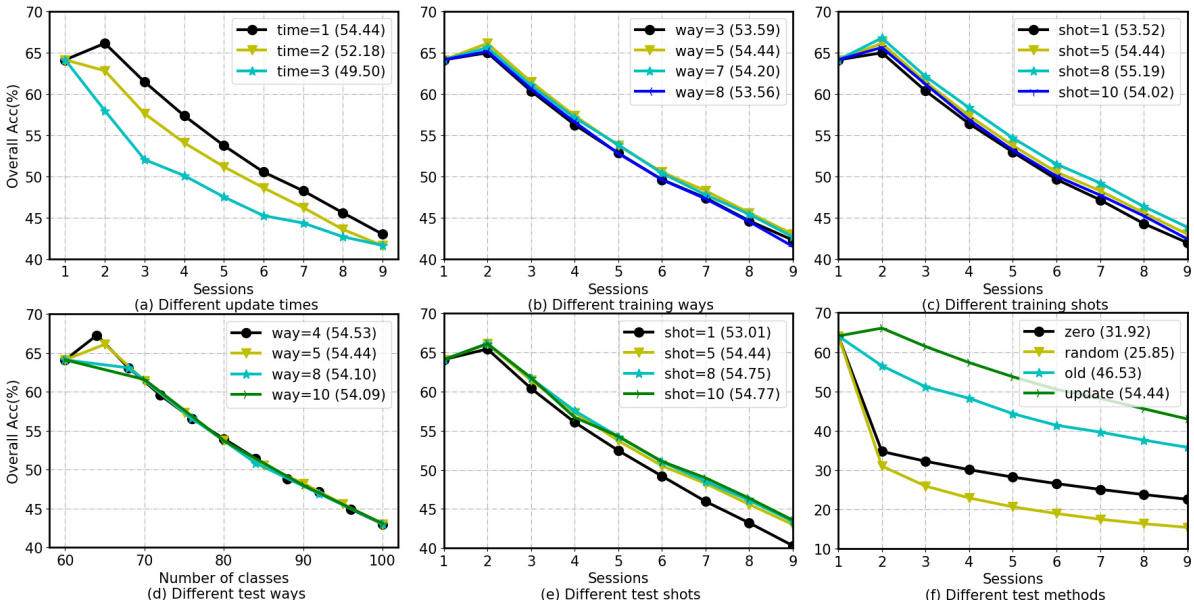

Figure 3. Performance analysis under different conditions on CIFAR-100. (a) The initial extensible representations are obtained through different times of updates. (b) During the random episode selection process, different numbers of ways are chosen in N-way K-shot setting. (c) During the training process at session 1, different numbers of shots are selected. (d) At session n (n$\geq$2), different numbers of incremental classes are used. (e) At session n (n$\geq$2), different numbers of samples in each incremental class are used. (f) At session n (n$\geq$2), different prototype update methods are used.

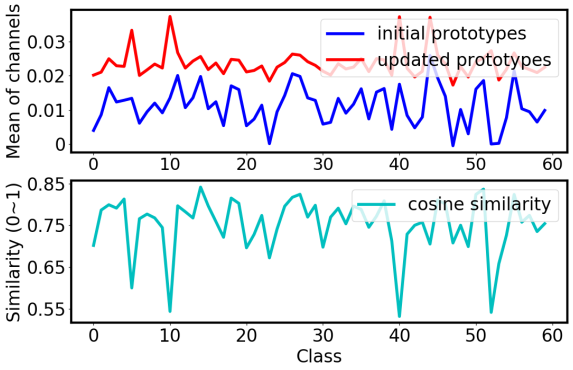

Figure 4. Illustration of the role of the prototype refinement mechanism on CIFAR-100. The distribution of the prototypes before and after update is shown in the upper row. The cosine similarity between the two is shown in the lower row.

provide the former in the text. The latter and the degree of dispersion in multiple tests are provided in the supplementary material. Our model uses the SGD optimizer during the training process. The initial learning rate is set to 0.02 and the attenuation rate is set to 0.0005. The model stops training after 70 epochs, and batch size is set to 128.

5.2. Ablation Study

To prove the effectiveness of our proposed method, we conduct several ablation experiments on CIFAR-100. The performance of our scheme is mainly attributed to two prominent components: the random episode selection strategy (RESS) and the self-promoted prototype refinement mechanism (SPPR). To clarify the function of these two parts, we replace the extensible representation and the SPPR with the representation obtained by standard learning and the fine-tuning update method (FT), respectively.

As can be seen from Table. 1 (the last three rows), the extensible representation brings a huge improvement in overall performance, about 5 percentage points on average. It demonstrates that extensible representation is far more useful than standard representation in FSCIL task. Without extensible representation, SPPR alone cannot play its role, and its performance drops by 4.24%. Due to the fact that SPPR enhances the expression ability of the feature representation, it brings an over 3 percent increment compared to fine-tuning. At the same time, we add the fine-tuning process to the overall scheme for observation. It can be seen that this step does not boost performance, which demonstrates the effectiveness of SPPR in updating the prototypes.

5.3. Analysis

The impact of the number of updates. To explore the impact of the number of updates on extensible representation during training, we design the following experiments on CIFAR-100. We obtain different extensible representation by repeating the random selection process n times and

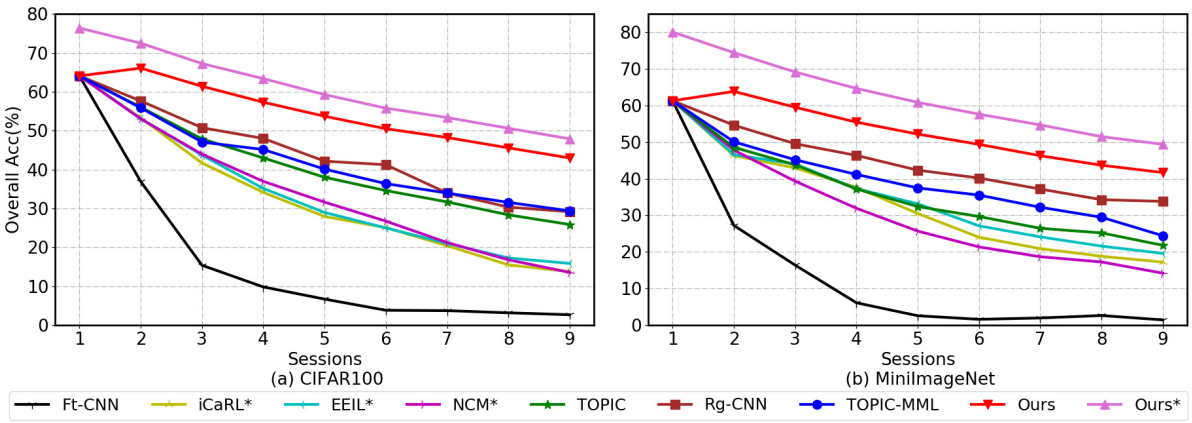

Figure 5. Comparison of our classification results with other methods on CIFAR-100 and MiniImageNet. The abscissa represents the incremental sessions and the ordinate represents the classification accuracy on the test set containing all the seen classes.

compare the test results. Take n equal to 2 as an example. That is to say, the final 60 classes of prototypes will be obtained through two incremental updates from 50 classes of false base prototypes (5 classes each time). We try to further strengthen the extensible of the representation by increasing the iteration numbers of prototype updates. However, the results show that with the number of iterations increases, the corresponding accuracy drops as in Fig. 3 (a). We argue that the difficulty of training may also increase sharply through multiple iterations, so the obtained representation does not benefit from it.

The sensitivity of the number of ways and shots during training. To verify the impact of selection classes and sample numbers during training, we train multiple models for comparison. As can be seen in Fig. 3 (b) and (c), the training processes of different ways and different shots achieve similar result. The performance of 1-shot learning is slightly worse than other conditions, and the performance also drops slightly when the number of ways exceeds 5. It suggests that our learning strategy is not sensitive to the change of the number of episodic classes and samples. For convenience, we set the fixed incremental classes (N=5, K=5) for all the training process in this paper.

The sensitivity of the number of ways and shots during test. To explore whether the obtained representation is adapted to different test settings, we show the test results with different numbers of incremental classes and samples. As can be seen in Fig. 3 (d), the feature representation obtained by 5-way training can achieve almost the same curve in the case of different test sessions, which further demonstrates the robustness of our proposed method. In Fig. 3 (e), when the shot of the test images is reduced to 1, the final result has a significant drop. However, when the shot of images increase to 5 and 10, it makes nearly no difference. We think it is because we obtain the class embeddings directly by averaging all the shots, which will not benefit from the increase of shots. Finally, we tested different update methods, namely setting new prototypes to zero, random numbers and keeping old prototypes unchanged. It can be seen in Fig. 3 (f) that our method achieves the best results.

5.4. Visualization

To verify the role of the self-promoted prototype refinement mechanism in this task, we show the following visualization results. As shown in Fig. 4, the initial prototypes of 60 base classes from CIFAR-100 dataset are averaged in the feature dimension and visualized in blue. Then 25 samples from 5 random classes are chosen and utilized to update the prototypes as Section. 4.2 states. It can be seen that the updated prototypes in red are close to the initial ones in both value and trend. And their high cosine similarity demonstrates that almost every prototype has the same distribution in the feature dimensions before and after the update.

5.5. Comparison with SOTA

To better assess the overall performance of our scheme, we compare it to the state-of-the-art methods of FSCIL (TOPIC and TOPIC-MML [27]) and some classical methods of CIL (iCARL* [20], EEIL* [4] and NCM* [12]). The following asterisk represents the result of applying corresponding CIL methods to the FSCIL task. In addition, we set the fine-tuning method (Ft-CNN) as the baseline, and adopt some regularization techniques (*i.e.,* weight regularization, data augmentation and distillation) on this basis (Rg-CNN) for comparison.

CIFAR-100 and MiniImageNet. It can be seen in Fig. 5 that under the average accuracy metric, our method surpasses the SOTA method over 13 percentage on CIFAR-100 dataset, and yields 17 percentage improvement on MiniImageNet dataset. At the same time, our incremental classifi-

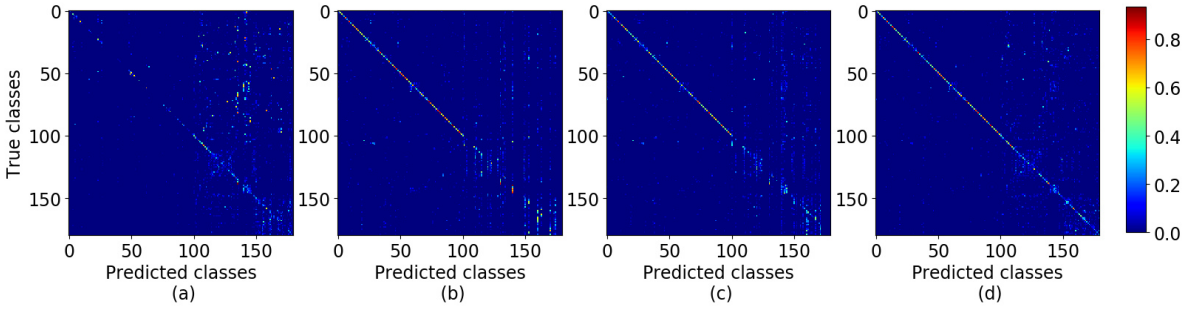

Figure 6. Confusion matrix of four different variations on CUB200. (a) Ft-CNN. (b) Fixed representation with distill loss function. (c) Extensible representation with distill loss function. (d) Our method.

Method	Sessions										
	1	2	3	4	5	6	7	8	9	10	11
Ft-CNN	68.68	44.81	32.26	25.83	25.62	25.22	20.84	16.77	18.82	18.25	17.18
iCaRL*	68.68	52.65	48.61	44.16	36.62	29.52	27.83	26.26	24.01	23.89	21.16
EEIL*	68.68	53.63	47.91	44.20	36.30	27.46	25.93	24.70	23.95	24.13	22.11
NCM*	68.68	57.12	44.21	28.78	26.71	25.66	24.62	21.52	20.12	20.06	19.87
TOPIC	68.68	61.01	55.35	50.01	42.42	39.07	35.47	32.87	30.04	25.91	24.85
TOPIC-MML	68.68	**62.49**	54.81	49.99	45.25	41.40	38.35	35.36	32.22	28.31	26.28
Ours	**68.68**	61.85	**57.43**	**52.68**	**50.19**	**46.88**	**44.65**	**43.07**	**40.17**	**39.63**	**37.33**

Table 2. Comparison of our classification resullts with other methods on CUB200.

cation results is higher than other methods at all sessions, and the attenuation is also slower. To make a fair comparison, we provide the results under the same accuracy of base classes, which are far below the best accuracy of ResNet-18 on these two datasets. This is why the results of the second session are even higher than the first session. The increment in the accuracy of the old classes benefits from the reverse effect of the relation projection. To manifest the real function of our method, we show the accuracy curve of the best result without the constraint of the base classes in violet line. It can be seen that in this case we get the best extensible representation and classification result.

To provide further insight into the behaviors of different methods, we compare their confusion matrix. As shown in Fig. 6, fine-tuning tends to classify all the samples into the incremental classes due to overfitting. Typical incremental methods (*i.e.,* fixed or extensible representation followed by different distill loss functions) often make mistakes when distinguishing newly incremental classes because of the lack of discriminative prototypes. The confusion matrix of our method suggests the superiority of both the representation and prototypes in all classes.

CUB200. As shown in Table 2, we achieve over 11 percentage improvement compared to the SOTA method. Since the number of incremental classes is twice than each of the above datasets, the forgetting rate at each session is much higher. It can be seen that the difficulty increases as the number of incremental classes and sessions increases. Different from that on above datasets, the initial classification accuracy (68.68%) is close to the best result ("$Ours^*$"), so we only report one result ("$Ours$") on this dataset.

6. Conclusion

In this paper, a novel incremental prototype learning scheme is proposed for FSCIL task. A random episode selection strategy is firstly proposed to enhance the extensibility and optimization capability of feature representation, and then all the prototypes are reorganized with a self-promoted prototype refinement mechanism. Consequently, our method incorporates incremental classes with few samples into recognition. Experimental results show that our model is superior in both performance and adaptability with respect to SOTA methods.

Acknowledgments. This work was supported by the National Key R&D Program of China under Grand 2020AAA0105702, National Natural Science Foundation of China (NSFC) under Grants 61872327 and U19B2038, the Fundamental Research Funds for the Central Universities under Grant WK2380000001 as well as Huawei Technologies Co., Ltd.

References

[1] Eden Belouadah and A. Popescu. Il2m: Class incremental learning with dual memory. *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 583–592, 2019. 1

[2] Junjie Cai, Zheng-Jun Zha, Meng Wang, Shiliang Zhang, and Qi Tian. An attribute-assisted reranking model for web image search. *IEEE transactions on image processing*, 24(1):261–272, 2014. 3

[3] Junjie Cai, Zheng-Jun Zha, Wengang Zhou, and Qi Tian. Attribute-assisted reranking for web image retrieval. In *Proceedings of the 20th ACM international conference on Multimedia*, pages 873–876, 2012. 3

[4] Francisco M Castro, Manuel J Marín-Jiménez, Nicolás Guil, Cordelia Schmid, and Karteek Alahari. End-to-end incremental learning. In *Proceedings of the European conference on computer vision (ECCV)*, pages 233–248, 2018. 2, 7

[5] Jia Deng, W. Dong, R. Socher, L. Li, K. Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *CVPR 2009*, 2009. 5

[6] Arthur Douillard, Matthieu Cord, Charles Ollion, Thomas Robert, and Eduardo Valle. Podnet: Pooled outputs distillation for small-tasks incremental learning. 2020. 2

[7] Chelsea Finn, Pieter Abbeel, and Sergey Levine. Model-agnostic meta-learning for fast adaptation of deep networks. *arXiv preprint arXiv:1703.03400*, 2017. 3

[8] Spyros Gidaris and Nikos Komodakis. Dynamic few-shot visual learning without forgetting. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4367–4375, 2018. 3

[9] Spyros Gidaris and Nikos Komodakis. Generating classification weights with gnn denoising autoencoders for few-shot learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 21–30, 2019. 3

[10] Haoyu He, Jing Zhang, Bhavani Thuraisingham, and Dacheng Tao. Progressive one-shot human parsing. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 2021. 3

[11] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016. 1, 4

[12] Saihui Hou, Xinyu Pan, Chen Change Loy, Zilei Wang, and Dahua Lin. Learning a unified classifier incrementally via rebalancing. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 831–839, 2019. 1, 2, 4, 7

[13] A. Krizhevsky. Learning multiple layers of features from tiny images. 2009. 5

[14] K. Li, Y. Zhang, Kunpeng Li, and Y. Fu. Adversarial feature hallucination networks for few-shot learning. *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 13467–13476, 2020. 3

[15] Zhizhong Li and Derek Hoiem. Learning without forgetting. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 40:2935–2947, 2018. 2

[16] Zhetao Li, Wei Wei, Tianzhu Zhang, Meng Wang, Sujuan Hou, and Xin Peng. Online multi-expert learning for visual tracking. *IEEE Transactions on Image Processing*, 29:934–946, 2019. 4

[17] Zhetao Li, Jie Zhang, Kaihua Zhang, and Zhiyong Li. Visual tracking with weighted adaptive local sparse appearance model via spatio-temporal context learning. *IEEE Transactions on Image Processing*, 27(9):4478–4489, 2018. 4

[18] Yaoyao Liu, Yuting Su, An-An Liu, Bernt Schiele, and Qianru Sun. Mnemonics training: Multi-class incremental learning without forgetting. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12245–12254, 2020. 2

[19] Laurens van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. *Journal of machine learning research*, 9(Nov):2579–2605, 2008. 1

[20] Sylvestre-Alvise Rebuffi, Alexander Kolesnikov, Georg Sperl, and Christoph H Lampert. icarl: Incremental classifier and representation learning. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, pages 2001–2010, 2017. 1, 2, 4, 7

[21] Mengye Ren, Renjie Liao, Ethan Fetaya, and R. Zemel. Incremental few-shot learning with attention attractor networks. In *NeurIPS*, 2019. 3

[22] Andrei A Rusu, Dushyant Rao, Jakub Sygnowski, Oriol Vinyals, Razvan Pascanu, Simon Osindero, and Raia Hadsell. Meta-learning with latent embedding optimization. *arXiv preprint arXiv:1807.05960*, 2018. 3

[23] K. Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *CoRR*, abs/1409.1556, 2015. 1, 4

[24] Jake Snell, Kevin Swersky, and Richard Zemel. Prototypical networks for few-shot learning. In *Advances in neural information processing systems*, pages 4077–4087, 2017. 3

[25] Flood Sung, Yongxin Yang, Li Zhang, Tao Xiang, Philip HS Torr, and Timothy M Hospedales. Learning to compare: Relation network for few-shot learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1199–1208, 2018. 3

[26] Xiaoyu Tao, Xinyuan Chang, Xiaopeng Hong, Xing Wei, and Yihong Gong. Topology-preserving class-incremental learning. ECCV, 2020. 2

[27] Xiaoyu Tao, Xiaopeng Hong, Xinyuan Chang, Songlin Dong, Xing Wei, and Yihong Gong. Few-shot class-incremental learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12183–12192, 2020. 2, 3, 5, 7

[28] Gido M. van de Ven and A. Tolias. Three scenarios for continual learning. *ArXiv*, abs/1904.07734, 2019. 2

[29] Oriol Vinyals, Charles Blundell, Timothy Lillicrap, Daan Wierstra, et al. Matching networks for one shot learning. In *Advances in neural information processing systems*, pages 3630–3638, 2016. 4, 5

[30] C. Wah, S. Branson, P. Welinder, P. Perona, and Serge J. Belongie. The caltech-ucsd birds-200-2011 dataset. 2011. 5

[31] Yue Wu, Yinpeng Chen, Lijuan Wang, Yuancheng Ye, Zicheng Liu, Yandong Guo, and Yun Fu. Large scale incremental learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 374–382, 2019. 1, 2

[32] Lu Yu, Bartlomiej Twardowski, Xialei Liu, Luis Herranz, Kai Wang, Yongmei Cheng, Shangling Jui, and Joost van de Weijer. Semantic drift compensation for class-incremental learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6982–6991, 2020. 2

[33] Wei Zhai, Yang Cao, Zheng-Jun Zha, HaiYong Xie, and Feng Wu. Deep structure-revealed network for texture recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11010–11019, 2020. 4

[34] Hanwang Zhang, Zheng-Jun Zha, Shuicheng Yan, Jingwen Bian, and Tat-Seng Chua. Attribute feedback. In *Proceedings of the 20th ACM international conference on Multimedia*, pages 79–88, 2012. 4

[35] Jing Zhang and Dacheng Tao. Empowering things with intelligence: A survey of the progress, challenges, and opportunities in artificial intelligence of things. *IEEE Internet of Things Journal*, 2020. 1

[36] Bowen Zhao, Xi Xiao, Guojun Gan, Bin Zhang, and Shu-Tao Xia. Maintaining discrimination and fairness in class incremental learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13208–13217, 2020. 2

[37] Kai Zhu, Wei Zhai, Z. Zha, and Y. Cao. Self-supervised tuning for few-shot segmentation. In *IJCAI*, 2020. 3